Topic Modeling with Structured Priors for Text-Driven Science

by

Michael J. Paul

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

July, 2015

Abstract

Many scientific disciplines are being revolutionized by the explosion of public data on the web and social media, particularly in health and social sciences. For instance, by analyzing social media messages, we can instantly measure public opinion, understand population behaviors, and monitor events such as disease outbreaks and natural disasters. Taking advantage of these data sources requires tools that can make sense of massive amounts of unstructured and unlabeled text. Topic models, statistical models that posit low-dimensional representations of data, can uncover interesting latent structure in large text datasets and are popular tools for automatically identifying prominent themes in text. For example, prominent themes of discussion in social media might include politics and health.

To be useful in scientific analyses, topic models must learn interpretable patterns that accurately correspond to real-world concepts of interest. This thesis will introduce topic models that can encode additional structures such as factorizations, hierarchies, and correlations of topics, and can incorporate supervision and domain knowledge. For example, topics about elections and Congressional legislation are related to each other (as part of a

ABSTRACT

broader topic of "politics"), and certain political topics have partisan associations. These types of relations between topics can be modeled by formulating the Bayesian priors over parameters as functions of underlying "components," which can be constrained in various ways to induce different structures.

This approach is first introduced through a topic model called *factorial LDA*, which models a factorized structure in which topics are conceptually arranged in multiple dimensions. Factorial LDA can be used to model multiple types of information, for example topic and political ideology. We then introduce a family of structured-prior topic models called SPRITE, which creates a unifying representation that generalizes factorial LDA as well as other existing topic models, and creates a powerful framework for building new models. This thesis will also show how these topic models can be used in various scientific applications, such as extracting medical information from forums, measuring healthcare quality from patient reviews, and monitoring public opinion in social media.

Committee:

Mark Dredze

Jason Eisner

Eric Horvitz

Hanna Wallach

Acknowledgments

This thesis would not have been possible without the support of so many people. Perhaps the people who had the most impact on my PhD are my four committee members.

I thank Mark Dredze for having faith in me as a junior grad student. It was a class project in my first semester—on topic modeling for public health—that turned out to set the research agenda for much of my PhD. Since then, Mark has advised me in many different ways, helping me formulate and accomplish goals. Mark generously involved me in many different projects in the past five years, and I'm grateful for the experiences I've had.

I owe much of my current knowledge and skills to Jason Eisner, who has always had so much to share and has pushed me to expand my boundaries. I remember getting private lessons on Jason's whiteboard on many topics, from semirings to Fourier transforms. Jason's passion for both research and education is inspiring: my habits of staying up late perfecting my slides and writing are likely attributable to Jason.

Eric Horvitz has been an amazing mentor both during and outside of my time at Microsoft Research, where I had the pleasure of interning for two summers. It has been a privilege to work with someone with so many ideas and so much wisdom to share, and Eric

always struck the perfect balance of proposing his ideas while letting me explore my own. I look forward to working together again in the future.

I was lucky to be able to talk to Hanna Wallach frequently over the years, and we have had many great conversations and brainstorming sessions, giving me feedback on my ideas while I was still figuring them out. We had important discussions about what eventually became SPRITE, a major component of this thesis. Hanna has a remarkable talent of combining unconditional enthusiasm with a critical eye.

Beyond my committee, I've benefitted from a number of other relationships with colleagues over the years, including but not limited to: Jordan Boyd-Graber, who enthusiastically shared a lot of ideas and wisdom with me during my PhD while he was at UMD, and who I am excited to work with in my next job at Colorado; Byron Wallace, who I've been lucky to collaborate with on multiple projects, and who has been happy to give advice on both research and life; Jacob Eisenstein, with whom I've had many helpful discussions, and who gave me advice and feedback while I was still figuring out what is now called factorial LDA; Noémie Elhadad, who has been a great collaborator in recent months and has had a lot of wisdom and advice to give; and David Broniatowski, who has become an important colleague from whom I've learned a lot.

I was fortunate to do my PhD at CLSP and the HLTCOE (the two NLP centers at JHU) and benefitted from the support and advice of many faculty over the years, especially David Yarowsky, Ben Van Durme, Chris Callison-Burch, and Suchi Saria. Perhaps more than faculty, I owe a lot to my labmates whose helpful discussions have helped me along

the way: Nick Andrews, Olivia Buzek, Frank Ferraro, Matt Gormley, Rebecca Knowles, Scott Novotney, Violet Peng, Delip Rao, Jason Smith, Adam Teichert, Tim Vieira, Svitlana Volkova, Travis Wolfe, Xuchen Yao, and certainly others. I want to thank Adrian Benton in particular for doing a lot of work recently experimenting with SPRITE and improving the code. I have also enjoyed talking to students at nearby UMD, especially Yuening Hu and Viet-An Nguyen.

It was exciting to be around so many great researchers during my two summers at Microsoft Research, where I had a lot of support from Ryen White and Janice Tsai. Before Microsoft, I spent a summer at Twitter, where I had the chance to try out topic modeling in the real world. I met a lot of great people there, especially my mentor Alek Kolcz as well as Jimmy Lin, whose sabbatical at Twitter overlapped with my time there.

I owe a lot of my success to my undergraduate mentors at the University of Illinois, who helped me on my path to grad school. Most especially, my undergrad thesis advisor Roxana Girju was instrumental in getting me excited about research and helping me get an early start in publishing. I also had great support from Dan Roth, whose lab I worked in for a year as a research programmer (among many other interactions); and Cheng Zhai, who was a great teacher and was always happy to make time to discuss ideas with me.

I have also had opportunities to interact with many colleagues outside of NLP and computer science, which have broadened my perspective and influenced some of the applied research in this thesis. In particular, there are a number of colleagues in health and medicine with whom I've had the fortunate opportunity to collaborate, including: Meg Chisolm, Matt

ACKNOWLEDGMENTS

Johnson, and Ryan Vandrey from the Department of Psychiatry and Behavioral Sciences at JHU; Urmimala Sarkar from the Department of Medicine at UCSF; Andrea Dugas from the Department of Emergency Medicine at JHU; and John Brownstein and Mauricio Santillana from the Harvard Medical School and Center for Biomedical Informatics.

I am grateful for the financial support I received from Microsoft Research, the National Science Foundation, and the Dean of the Whiting School of Engineering at Johns Hopkins. These fellowships provided a huge amount of academic freedom during my time here.

Finally, I am grateful for the support of my friends and loved ones over the years, especially my parents who brought me into the world of science early on.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Massive quantities of public text data are readily available across the Web in a wide variety of domains and formats, including news media articles, scientific literature, reviews of products and services, weblogs and forum discussions, and social media status updates. Analysis of such text allows researchers to detect and monitor current events (Allan et al., 1998; Sakaki et al., 2010), understand consumer preferences and public opinion (Pang and Lee, 2004; O'Connor et al., 2010), and much more.

The availability of text vastly overwhelms the availability of human readers and their time. Managing large volumes of unstructured text requires tools that can categorize words and documents in ways that provide a high-level overview that humans can work with and understand. One such tool is called a *topic model*. The primary contribution of this thesis is the introduction of novel types of topic models and demonstrating how they can be used for various tasks.

Topic models (in particular, *probabilistic* topic models (Blei et al., 2003b)) cluster words together into "topics" and automatically label tokens in a corpus with those topics. (A "token" is an instance of a word in a document.) For example, in a collection of news articles, one might expect to find articles about topics such as politics and finance. Figure 1.1 shows examples of what a topic model learns from news articles: each topic is a list of related words. Decomposing the content of documents into a small set of topics allows humans to understand the high-level content of a corpus that may be too large to read. Probabilistic topic models allow one to understand the thematic composition of a corpus and to identify which documents are about which topics, with potentially multiple topics per document. This makes topic models well-suited for visualizing and exploring text corpora (Eisenstein et al., 2012; Chaney and Blei, 2012; Snyder et al., 2013).

Topic information extracted from topic models can also be analyzed for specific tasks and research problems. To give some examples, topic models have been used for:

- studying trends in scientific research topics over time (Hall et al., 2008; Paul and Girju, 2009b) and organizing scientific research grants (Talley et al., 2011);
- understanding historic themes in large collections of literature (Jockers and Mimno, 2013; Blei, 2013), an important part of the emerging discipline of *digital humanities*;
- analyzing the ideology of politicians (Gerrish and Blei, 2012; Nguyen et al., 2015a);
- identifying geographic variation in language (Eisenstein et al., 2010) and inferring demographics from writing (Rao et al., 2011);

Topic 3	**Topic 23**	**Topic 24**	**Topic 35**	**Topic 50**
FINANCE	ENVIRONMENT	ELECTIONS	EDUCATION	MEDICINE
company	environmental	election	school	health
million	plant	state	students	aids
billion	water	campaign	university	medical
corp	energy	vote	student	hospital
share	waste	republican	college	disease
stock	state	democratic	education	drug
business	nuclear	percent	schools	patients
new	pollution	voters	board	doctors
offer	epa	candidates	teachers	dr
companies	department	votes	high	research
chairman	plants	won	training	treatment
president	quake	elected	public	blood
companys	gas	political	teacher	heart
firm	air	presidential	williams	care
percent	species	race	class	virus

Figure 1.1: Examples of topics (out of 50 total topics) from a topic model applied to 2,248 news articles, visualized as the 15 most probable words from each topic's unigram distribution. The smallcaps labels for each topic are manually assigned by the author.

- tracking health issues in social media (Paul and Dredze, 2011; Prier et al., 2011; Resnik et al., 2015) and summarizing medical records (Cohen et al., 2014).

The title of this thesis refers to *text-driven science*, meaning the use of large text corpora as data sources for answering scientific questions. This thesis will show how structured topic models can be be applied to a variety of problems in a health and social science, including extracting information about drugs, learning about patient perceptions of healthcare quality, and measuring public opinion.

1.1 Probabilistic Modeling of Text

How do topic models work? Before we can understand probabilistic topic modeling, let us first consider probabilistic modeling in a general.

A probabilistic model posits that data is generated according to some underlying probabilistic process. Suppose the data consists of text. The simplest probabilistic model that could generate text is to randomly pick words according to some probability distribution. If we were modeling English, words like "the" and "a" would have high probability, words like "cat" and "green" would have less high probability, and words like "marvelry" and "thrombus" would have low probability. This model is called a *unigram* model. One could randomly generate documents in this manner. Such documents would loosely resemble characteristics of English, but would be nonsensical.

A slightly more complex model would randomly pick words based on a category that a document belongs to. For example, documents about sports would be likely to include words like "baseball" and "catch", while documents about animals are more likely to include words like "cat" and "dog". This can be achieved by having different distributions over words for each document category.

This is called a *mixture* model because the corpus is modeled as a mix of different category-specific unigram models. As a data-generating process, the way this model imagines that each document is generated is by first randomly choosing a category and then randomly choosing words from that category's word distribution. This is the model underlying naive Bayes classification in machine learning, and is similar to a topic model since

each "category" represents something like a topic.

The term "topic model" refers to a more specific, more complex model. Like the category mixture model, probabilistic topic models have a unigram model for each "topic", but instead of choosing one topic in the entire document, a topic is chosen for each word token in the document, which then influences the choice of word. This is called an *admixture* model, meaning that each document has its own mixture of topics, though the topics themselves are shared across all documents. We will describe the data-generating process of topic models formally in Chapter 2.

Clearly, these simplistic models are not complete or accurate representations of how language works. For one thing, standard topic models do not even distinguish between different orderings of words, treating documents as weighted sets of words (that is, a *bag-of-words* representation). However, such models are useful tools for representing particular aspects of language that we can reason about. Probabilistic topic models can answer the question, "If this is how language worked, what would the topics associated with these documents look like?" By answering this question, we can obtain the meaningful clusters of words shown in Figure 1.1, where each "cluster" is given by the 15 most probable words in the topic's word distribution.

1.2 Thesis Overview

We will begin by providing important background on probabilistic topic modeling in Chapter 2. This chapter provides background on common probability distributions, then defines common topic models—including latent Dirichlet allocation (LDA) (Blei et al., 2003b), which forms the basis of the topic models introduced later—as well as more complex topic models that capture various structures, such as topic hierarchies. We will describe learning and inference in topic models, focusing on Gibbs sampling as the inference method used in this thesis, and we will discuss how to evaluate topic models.

The subsequent chapters introduce new topic models (Chapters 3 and 5) and show how to use these topic models for applications in health and social science (Chapters 4 and 6).

Chapter 3 introduces factorial LDA (FLDA), a novel topic model that organizes word distributions into different dimensions. For example, one could use FLDA to jointly model topic and sentiment—two dimensions—in product reviews. This model makes use of a novel *structured prior* to induce the multi-dimensional structure. This chapter is based on the published work of Paul and Dredze (2012a).

Chapter 4 applies FLDA to two applications in health science: summarizing information about drugs and measuring healthcare quality from reviews. A contribution of this chapter is to extend the structured priors to incorporate domain knowledge for the particular applications, as well as additional evaluation of the utility of the model for real tasks. The drug application is based on the papers of Paul and Dredze (2013) and Paul and Dredze (2012b), while the healthcare application is based on the papers of Paul et al. (2013) and

Wallace et al. (2014). The healthcare regression experiments in this chapter were conducted by Byron Wallace and Thomas Trikalinos.

Chapter 5 introduces SPRITE, a structured-prior topic modeling framework that generalizes FLDA. We show how a variety of topic structures can be modeled with SPRITE, and we show how a variety of previous structured topic models can be derived as special cases of this model. This chapter is based on the published article of Paul and Dredze (2015).

Chapter 6 applies SPRITE to the task of modeling perspective in text. This includes modeling opinions and ideology in reviews and political text, as well as modeling public opinion in social media on a variety of issues. A technical contribution is to show specific instantiations of SPRITE that capture multiple structures, including factorizations and hierarchies. This chapter includes additional experiments published in Paul and Dredze (2015), as well as experiments from a paper in preparation by Benton et al. (2015). Some experiments were conducted by Adrian Benton.

Chapter 7 then summarizes the contributions of each chapter and suggests future research directions.

1.2.1 Notation

We use smallcaps to refer to hypothetical or human-assigned names of topics, e.g., SPORTS or POLITICS. For example, we might refer to "the SPORTS topic" as a hypothetical example while explaining the intuition of a topic model, even though the model itself does not assign a name to any topic, as in Figure 1.1.

When using lowercase letters as indices of vectors, we use the uppercase of the same letter to denote the dimensionality of the vector. For example, if $\boldsymbol{x}$ is an I-dimensional vector, each ith component is denoted x_i.

We sometimes use bold letters to make it clear that a variable is a vector, but not all vectors will be bolded if it is obvious. For example, in a topic model, each mth document has a vector of parameters θ_m, while we use bold $\boldsymbol{\theta}$ to denote the set of all parameters, $\{\theta_m\}_{m=1}^{M}$, to make this distinction clear.

The hyperparameters of Dirichlet priors use the same variable name as the parameters but with a tilde; for example, multinomial parameters θ have a Dirichlet($\tilde{\theta}$) prior.

Chapter 2

Background: Probabilistic Topic Modeling

This chapter introduces the important concepts behind probabilistic topic modeling, including model definitions, learning, and evaluation, before introducing new topic models in Chapters 3 and 5.

The first section provides some background on important probability distributions used by probabilistic topic models.

Section 2.2 introduces the simplest probabilistic topic model, while Section 2.3 describes Bayesian topic modeling and introduces the most popular topic model, latent Dirichlet allocation (Blei et al., 2003b). Subsection 2.3.3 introduces the concept of a structured prior, which is an important concept in this thesis.

Section 2.4 describes extensions to topic models that include richer topic structures.

This section focuses on structures and models that will be used later in this thesis in Chapters 3 and 5.

Section 2.5 discusses the inference and optimization of topic models, with a focus on Gibbs sampling. Section 2.6 then discusses how to evaluate and select topic models.

2.1 Preliminaries: Multinomials, Dirichlets, and Dirichlet-multinomials

As explained in the introduction, topic modeling is an instance of probabilistic modeling, in which a dataset such as a text corpus is modeled as the output of an underlying probabilistic process. This section provides background on the basic probability distributions used by the topic models described in this thesis.

2.1.1 The Multinomial Distribution

The *multinomial* distribution describes a probability distribution over histograms of a fixed size. Let $\boldsymbol{n} \in \mathbb{N}_{\geq 0}^{I}$ denote an I-dimensional vector of counts, such that n_i is the integer count of the ith component. For example, in a unigram language model, n_v would denote the count of the vth word in the corpus. Let N be the total count, $N = \sum_{i=1}^{I} n_i$. In a language model, N would be the number of tokens in a corpus.

Among the set of possible histograms of size N (of which there are "$N - I - 1$ choose

$I-1$" combinations), the probability of a particular histogram $\boldsymbol{n}$ is given by the multinomial probability mass function with parameters θ:

$$P(\boldsymbol{n}|\theta) = \frac{\Gamma((\sum_{i=1}^{I} n_i) + 1)}{\prod_{i=1}^{I} \Gamma(n_i + 1)} \prod_{i=1}^{I} \theta_i^{n_i} \tag{2.1}$$

where Γ is the gamma function, which for integers is defined as $\Gamma(x) = (x-1)!$. This expression on the left is a normalization term that is constant with respect to θ.

The parameters $\theta \in \mathbb{R}^I_{\geq 0}$ have the constraint that $\sum_{i=1}^{I} \theta_i = 1$ and can be interpreted as an explicit probability distribution.

A special case of the multinomial distribution is when $N = 1$. In this case, this distribution is also called the *categorical* distribution, and the probability of the random variable taking the ith value is simply θ_i, that is $P(n_i = 1|\theta) = \theta_i$. When $N = 1$, the vector $\boldsymbol{n}$ is called an *indicator* vector.

2.1.2 The Dirichlet Distribution

The *Dirichlet* distribution can be thought of as a "distribution over distributions", as it is a distribution over vectors in an $(I-1)$-dimensional simplex. These vectors are most commonly used to parameterize multinomial distributions.[1] That is, while a multinomial describes the probability of a histogram $\boldsymbol{n}$ conditioned on parameters θ, a Dirichlet distribution can describe the likelihood of the parameters θ conditioned on hyperparameters

[1] However, in Section 5.1.1 we will see an example of using a Dirichlet distribution to generate a vector that is not used to parameterize a probability distribution.

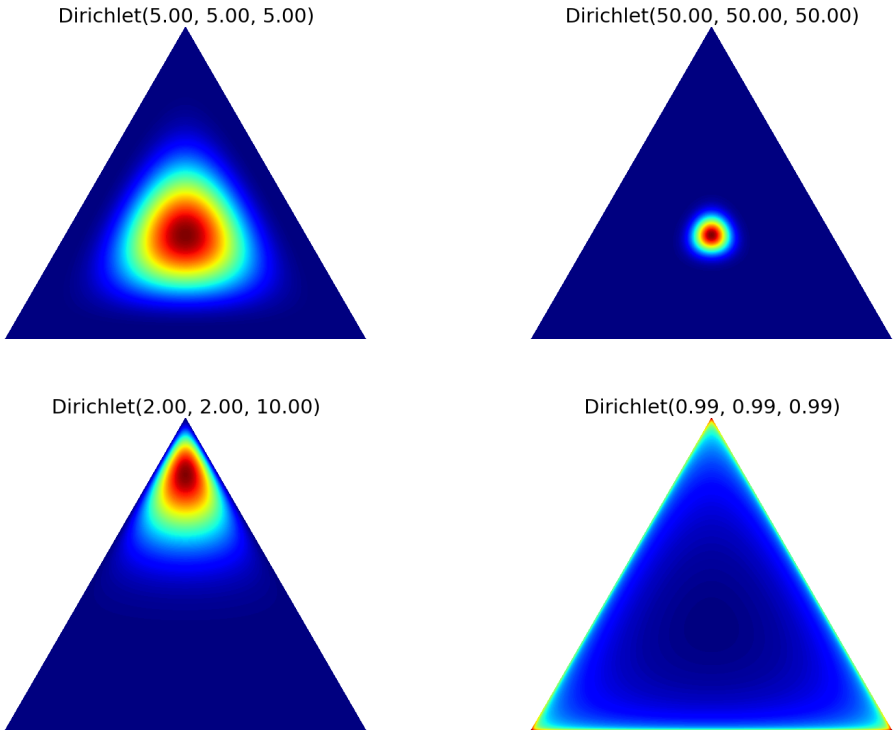


Figure 2.1: The Dirichlet probability density function under different parameterizations, represented on a 2-simplex. The key observations are that the density becomes increasingly concentrated around the mean as the hyperparameters increase, and the density concentrates around the boundaries of the simplex when the hyperparameters are less than 1.0.

$\tilde{\theta} \in \mathbb{R}^I_{>0}$. This notation is consistent with the convention used in this thesis, in which a distribution over parameters θ has hyperparameters denoted with a tilde, $\tilde{\theta}$. The Dirichlet has the following probability density function:

$$P(\theta|\tilde{\theta}) = \frac{\Gamma(\sum_{i=1}^{I} \tilde{\theta}_i)}{\prod_{i=1}^{I} \Gamma(\tilde{\theta}_i)} \prod_{i=1}^{I} \theta_i^{\tilde{\theta}_i - 1} \tag{2.2}$$

The mean of a Dirichlet($\tilde{\theta}$) distribution is proportional to $\tilde{\theta}$, so $\mathbb{E}_{\text{Dirichlet}(\tilde{\theta})}[\theta_i] \propto \tilde{\theta}_i$. This means that $\tilde{\theta}$ describes the "average" distribution under this distribution over distributions.

$\sum_{i=1}^{I} \tilde{\theta}_i$ is the *precision*, also called the concentration, and decreases monotonically with the inverse variance of the distribution.

When the $\tilde{\theta}_i$ values are less than 1, the density function is concentrated around the boundaries of the $(I-1)$-simplex, which means that higher likelihood is given to θ vectors where most of the mass is on a small number of components, and most of the components have value near zero. This can be observed in Figure 2.1, which provides plots of the Dirichlet density function for different hyperparameter values.

If all components of $\tilde{\theta}$ have the same value—that is, $\tilde{\theta}_i = \tilde{\theta}_{i'}$ for all i, i'—then this is called a *symmetric* Dirichlet distribution; otherwise it is called asymmetric. The mean of symmetric Dirichlet distribution is the center of a simplex, while the precision controls how tightly concentrated the density is around the mean.

2.1.3 Combining Multinomials and Dirichlets

In Bayesian modeling, explained in Section 2.3, $P(\theta|\tilde{\theta})$ is called the *prior* distribution and $P(\theta|\boldsymbol{n}, \tilde{\theta}) = \frac{P(\theta, \boldsymbol{n}|\tilde{\theta})}{P(\boldsymbol{n}|\tilde{\theta})}$ is called the *posterior* distribution. The prior can be interpreted as expressing prior beliefs about the possible values of the parameters before the data $\boldsymbol{n}$ is considered, while the posterior reflects both the prior and the data.

The distribution $P(\boldsymbol{n}|\tilde{\theta})$ is called the *marginal* distribution of the counts $\boldsymbol{n}$ conditioned on the Dirichlet($\tilde{\theta}$) prior, because this can be obtained by marginalizing θ out of the joint distribution of $\boldsymbol{n}$ and θ: $\int_\theta P(\boldsymbol{n}|\theta)P(\theta|\tilde{\theta})\, \mathrm{d}\theta$. This marginal distribution is called the *Dirichlet-multinomial* distribution, or sometimes the Dirichlet compound multinomial

(DCM) distribution or the multivariate Pólya distribution. It is defined as:

$$P(\boldsymbol{n}|\tilde{\theta}) = \int_{\theta} P(\boldsymbol{n}|\theta)P(\theta|\tilde{\theta})\,\mathrm{d}\theta = \frac{\Gamma(\sum_{i=1}^{I}\tilde{\theta}_i)}{\Gamma(\sum_{i=1}^{I} n_i + \tilde{\theta}_i)}\prod_{i=1}^{I}\frac{\Gamma(n_i+\tilde{\theta}_i)}{\Gamma(\tilde{\theta}_i)} \tag{2.3}$$

The joint distribution $P(\theta_i, n_i|\tilde{\theta}) = P(n_i|\theta_i)P(\theta_i|\tilde{\theta})$ is proportional to $\theta_i^{n_i}\theta_i^{\tilde{\theta}_i-1} = \theta_i^{n_i+\tilde{\theta}_i-1}$. For this reason, the $\tilde{\theta}$ parameters are sometimes referred to as "pseudocounts" because they augment the observed counts $\boldsymbol{n}$ when the Dirichlet distribution is multiplied by the multinomial.

From these joint and marginal distributions, one can solve for the posterior distribution:

$$\begin{aligned} P(\theta|\boldsymbol{n},\tilde{\theta}) = \frac{P(\theta,\boldsymbol{n}|\tilde{\theta})}{P(\boldsymbol{n}|\tilde{\theta})} &= \frac{\Gamma(\sum_{i=1}^{I} n_i + \tilde{\theta}_i)}{\prod_{i=1}^{I}\Gamma(n_i+\tilde{\theta}_i)}\prod_{i=1}^{I}\theta_i^{n_i+\tilde{\theta}_i-1} \\ &= \text{Dirichlet}(\boldsymbol{n}+\tilde{\theta}) \end{aligned} \tag{2.4}$$

It turns out that the posterior distribution over the parameters is also a Dirichlet distribution, whose parameters are the sum of both the counts $\boldsymbol{n}$ and the "pseudocounts" $\tilde{\theta}$. This property is used in Section 2.5.1.1. Because the posterior of the multinomial parameters is the same type of distribution as the prior, the Dirichlet distribution is called a *conjugate* prior to the multinomial distribution (Raiffa and Schlaifer, 1961).

2.2 Basic Topic Modeling

In a probabilistic topic model, it is assumed that documents in a text corpus can be explained as a combination of underlying "topics". There are T topics, and each topic is defined as a probability distribution over all V words in the vocabulary. The parameters for the distribution of the tth topic are denoted ϕ_t. Each ϕ_t is thus a vector of length V.

Additionally, each document in the corpus is associated with a distribution over the set of T topics, denoted θ_m for the mth document. Each θ_m is a vector of length T.

Each word token in the vocabulary is associated with two random variables: the word value (which is observed for real documents) and its topic assignment (which is unobserved). The histogram of topic assignments in each document is distributed according to a multinomial distribution with parameters θ_m, and conditioned on the topic assignments, the histogram of word counts for tokens assigned to each topic is distributed according to a multinomial distribution with parameters ϕ_t.

For the nth token within the mth document, the joint distribution over the observed word value and unobserved topic assignment is factorized as, $P(w_{mn} = v|z_{mn} = t, \phi_t)$ $P(z_{mn} = t|\theta_m)$, the product of the topic's word distribution and the document's topic distribution:

$$P(w_{mn} = v|z_{mn} = t, \phi_t) = \phi_{tv} \tag{2.5}$$

$$P(z_{mn} = t|\theta_m) = \theta_{mt} \tag{2.6}$$

This is an admixture model, where each document is modeled as its own mixture (θ_m), but the mixture components ($\boldsymbol{\phi}$) are shared across all documents.

The marginal probability of an observed word token is thus:

$$\begin{aligned} P(w_{mn} = v|\boldsymbol{\theta}, \boldsymbol{\phi}) &= \sum_{t=1}^{T} P(w_{mn} = v|z_{mn} = t, \phi_t) P(z_{mn} = t|\theta_m) \\ &= \sum_{t=1}^{T} \phi_{tv} \theta_{mt} \end{aligned} \tag{2.7}$$

Conditioned on the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, the words are conditionally independent. Therefore, the joint likelihood of all words in the corpus $\boldsymbol{w}$ is the product of each word token's marginal probability:

$$P(\boldsymbol{w}|\boldsymbol{\theta}, \boldsymbol{\phi}) = \prod_{m=1}^{M} \prod_{n=1}^{N_m} P(w_{mn}|\boldsymbol{\theta}, \boldsymbol{\phi}) \tag{2.8}$$

where M is the number of documents and N_m is the number of tokens in the mth document.

An alternative way of defining this model is with its "generative story", meaning the description of how the model assumes the variables are randomly generated. (The generative story could be interpreted as pseudocode for implementing an algorithm that randomly generates data according to the model.) The generative story for this basic topic model is:

1. For each document m:
   (a) For each token n in document m:
      i. Sample topic value $z_{mn} \sim \theta_m$

ii. Sample word value $w_{mn} \sim \phi_{z_{mn}}$

Figure 2.2(a) shows the graphical model diagram, explained in the caption.

Most famously, this model was introduced as *probabilistic latent semantic analysis* (PLSA) (Hofmann, 1999), where it was presented as a probabilistic version of latent semantic analysis (Deerwester et al., 1990), which factorizes a matrix of document-term counts into matrices that can be interpreted as associations between topics in documents and words in topics.

A less general variant of the same model was independently proposed by McCallum in the same year (1999). McCallum's model was motivated by the problem of multi-label classification and was used to attribute each document's tokens to one of the document labels. As such, each "topic" corresponded to a label in the data, and each document's distribution over "topics" was restricted to the document's set of labels.[2]

2.3 Bayesian Topic Modeling

In Bayesian modeling, the parameters of the distributions are also treated as random variables. The parameters are therefore modeled as distributions which are called *prior* distributions, as described in Section 2.1.3. In the case of probabilistic topic models, the parameters are $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, and Bayesian topic models define prior distributions (typically referred to as simply "priors") over these parameters, $P(\theta|\tilde{\theta})$ and $P(\phi|\tilde{\phi})$.

[2]A Bayesian extension of McCallum's label attribution model was independently proposed as Labeled LDA (Ramage et al., 2009).

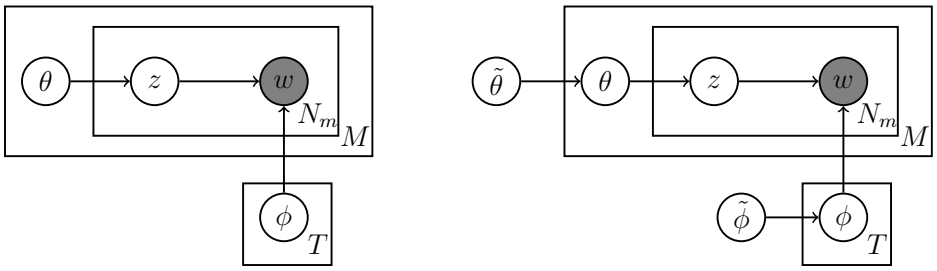
The parameters of the prior distributions are called *hyperparameters*. In this thesis, we denote the hyperparameters for a parameter's prior using the same variable name but with a tilde above it. That is, the prior over θ is parameterized by $\tilde{\theta}$ and the prior over ϕ is parameterized by $\tilde{\phi}$.

As shown in Section 2.5, modeling the parameters this way allows us to reason about the distribution over the parameters conditioned on the data. This is called the *posterior* distribution. Bayesian inference allows us to reason about the likelihood of different parameterizations of a model.

The posterior distribution depends on both the observed data and the prior. The prior can therefore be used to bias the estimate of the parameters.

One reason to bias the parameters is to favor simpler models, which can prevent overfitting and learn parameters that better generalize to new data (Mitchell, 1980). This type of bias is called *regularization*. A common type of regularization when modeling text is *smoothing*, in which "pseudocounts" are added to observed counts when estimating the parameters. It can be shown that pseudocount smoothing can be derived as an estimate of the parameters when the prior is the Dirichlet distribution (Section 2.1.2), and in fact this type of smoothing is sometimes called "Dirichlet prior smoothing" (Zhai and Lafferty, 2004). The Dirichlet distribution is a common prior for text models including topic models, as will be discussed in the next subsection.

Another reason to bias the parameters is to guide the parameters toward values informed by prior knowledge and domain expertise of humans. For example, priors can be used to



(a) PLSA (b) LDA

Figure 2.2: The graphical model plate diagrams for (a) probabilistic latent semantic analysis (PLSA) and (b) latent Dirichlet allocation (LDA). These diagrams show the factorization of the joint distribution over the variables in the models. Nodes represent random variables, where the gray shaded variables are observed. Variables are conditionally independent of each other given the values of the parent nodes. Boxes ("plates") indicate that there are multiple random variables, where the lower right indicates the number (for example, there are T different ϕ vectors).

encode preferences that certain words should or should not co-occur together in topics (Andrzejewski et al., 2009; Hu et al., 2011). In Chapter 4 of this thesis, we will show methods for incorporating domain expertise into topic model priors.

We will now describe a few types of priors used in topic modeling.

2.3.1 Latent Dirichlet Allocation

A natural prior for the multinomial parameters of a topic model is the Dirichlet distribution. When the topic model from the previous subsection uses Dirichlet priors for the parameters, it is called *latent Dirichlet allocation* (LDA) (Blei et al., 2003b). This model is illustrated in Figure 2.2(b). LDA is by far the most widely used probabilistic topic model and popularized the concept of topic modeling.

The generative story for LDA thus includes the generation of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$:

1. For each topic t:

   (a) Sample $\phi_t \sim \text{Dirichlet}(\tilde{\phi})$

2. For each document m:

   (a) Sample $\theta_m \sim \text{Dirichlet}(\tilde{\theta})$

   (b) For each token n in document m:

      i. Sample topic value $z_{mn} \sim \theta_m$

      ii. Sample word value $w_{mn} \sim \phi_{z_{mn}}$

The joint likelihood of all random variables in LDA is then:

$$P(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\phi}|\tilde{\theta}, \tilde{\phi}) = \prod_{t=1}^{T} P(\phi_t|\tilde{\phi}) \prod_{m=1}^{M} P(\theta_m|\tilde{\theta}) \prod_{n=1}^{N_m} P(z_{mn}|\theta_m) P(w_{mn}|z_{mn}, \phi_t) \quad (2.9)$$

where $P(\theta_m|\tilde{\theta})$ is given by Dirichlet$(\tilde{\theta})$ and $P(\phi_t|\tilde{\phi})$ is given by Dirichlet$(\tilde{\phi})$.

The inference and optimization algorithms used in this thesis will make use of the joint distribution of the data variables $\boldsymbol{w}$ and $\boldsymbol{z}$, marginalizing over the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, so we will define this distribution here.

The marginal probability of the topic counts in each document is given by a Dirichlet-multinomial distribution, defined in Section 2.1.3. The joint probability of all topic counts

$\boldsymbol{z}$ across the corpus is the product of each document's probability:

$$\begin{aligned} P(\boldsymbol{z}|\tilde{\theta}) &= \prod_{m=1}^{M} \int_{\theta_m} P(\theta_m|\tilde{\theta}) P(\boldsymbol{z}_m|\theta_m) \, \mathrm{d}\theta_m \\ &= \prod_{m=1}^{M} \frac{\Gamma(\sum_{t=1}^{T} \tilde{\theta}_t)}{\Gamma(\sum_{t=1}^{T} n_t^m + \tilde{\theta}_t)} \prod_{t=1}^{T} \frac{\Gamma(n_t^m + \tilde{\theta}_t)}{\Gamma(\tilde{\theta}_t)} \end{aligned} \tag{2.10}$$

where $\boldsymbol{z}_m$ denotes the subset of $\boldsymbol{z}$ for tokens within the mth document, and n_t^m is the number of tokens in the mth document assigned to the tth topic.

Similarly, a Dirichlet-multinomial distribution describes the marginal probability of the words counts among tokens assigned to each topic. The joint probability of all word counts $\boldsymbol{w}$ across the corpus is the product of each topic's probability:

$$\begin{aligned} P(\boldsymbol{w}|\boldsymbol{z}, \tilde{\phi}) &= \prod_{t=1}^{T} \int_{\phi_t} P(\phi_t|\tilde{\phi}) P(\boldsymbol{w}_t|\phi_t) \, \mathrm{d}\phi_t \\ &= \prod_{t=1}^{T} \frac{\Gamma(\sum_{v=1}^{V} \tilde{\phi}_v)}{\Gamma(\sum_{v=1}^{V} n_v^t + \tilde{\phi}_v)} \prod_{v=1}^{V} \frac{\Gamma(n_v^t + \tilde{\phi}_v)}{\Gamma(\tilde{\phi}_v)} \end{aligned} \tag{2.11}$$

where $\boldsymbol{w}_t$ denotes the subset of $\boldsymbol{w}$ such that the tokens' z variables have value t, and n_v^t is the number of tokens assigned to the tth topic with the vth vocabulary value.

The joint distribution $P(\boldsymbol{z}, \boldsymbol{w}|\tilde{\theta}, \tilde{\phi})$ is then the product of $P(\boldsymbol{z}|\tilde{\theta})$ and $P(\boldsymbol{w}|\boldsymbol{z}, \tilde{\phi})$.

2.3.2 Logistic Normal Priors

Other distributions have also been used as priors in topic modeling. We will not cover all alternative priors here, but we will describe the *logistic normal* distribution, which is

used by topic models described in Sections 2.3.3.2 and 2.4.1.

The logistic normal prior reparameterizes the topic distributions θ_m using a multivariate logistic function, and the parameters of the logistic function are normally distributed. That is, each $\theta_{mt} \propto \exp(\eta_{mt})$ and $\boldsymbol{\eta}_m \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Since $\boldsymbol{\eta} \in \mathbb{R}$, there is flexibility in the choice of prior over $\boldsymbol{\eta}$. The advantage of the normal distribution is that it can model covariance between different elements of $\boldsymbol{\eta}$, and so the logistic normal prior was originally used in topic modeling to learn correlations between topics (Blei and Lafferty, 2007).

2.3.3 Structured Priors

In this thesis, we define a *structured prior* as one in which the hyperparameters of a prior density function are functions of additional parameters. A major contribution of this thesis is to introduce new types of structured priors for enriching topic models. This subsection will review two existing topic models with structured priors.

2.3.3.1 Dirichlet-multinomial regression

One type of structured prior is to rewrite the hyperparameters of a Dirichlet distribution as functions of additional parameters. This approach will used later in this thesis.

This idea was first applied to topic modeling in the *Dirichlet-multinomial regression* (DMR) topic model by Mimno and McCallum (2008). We will refer to this topic model simply as DMR, although the term Dirichlet-multinomial regression applies to a broader concept (Guimaraes and Lindrooth, 2005). In Mimno and McCallum's version of DMR,

the Dirichlet prior over the mth document's topic distribution is defined such that $\tilde{\theta}_{mt} = \exp(\sum_i \alpha_{mi}\lambda_{it})$, where α_m is a vector of feature values associated with the mth document (observed as input), and λ_i is a vector of regression coefficients for the ith feature. The $\boldsymbol{\lambda}$ vectors are used to increase or decrease the prior for different topics given a document's feature values. For example, α_m might encode a document's timestamp or author, and given these values, the prior for certain topics will be altered depending the values of $\boldsymbol{\lambda}$. The log-linear formulation, meaning a linear function that is exponentiated, constrains $\tilde{\theta}$ to positive values, as is necessary for the Dirichlet distribution.

Notice that this prior is specific to the mth document, rather than using a single Dirichlet distribution that is shared across all documents, as in LDA. The $\boldsymbol{\lambda}$ parameters are shared globally.

2.3.3.2 Sparse additive generative models

Sparse additive generative models (SAGE) (Eisenstein et al., 2011) reparameterize the word distributions $\boldsymbol{\phi}$ in terms of a multivariate logistic function, similar to the logistic normal prior described in Section 2.3.2, but applied to the word distributions rather than the topic distributions.

The SAGE parameterization includes more structure than a standard logistic normal prior and allows the logistic parameters to be functions of multiple variables. A standard

topic parameterization under SAGE is:

$$\phi_{tv} \propto \exp(\eta_{0v} + \eta_{tv}) \tag{2.12}$$

where η_{tv} is a weight of the vth word in the tth topic, while η_{0v} is a globally shared weight of the vth word for all topics. SAGE can therefore learn topic-specific weights as deviations from an overall "background" distribution, which offers additional interpretability over a standard topic model.

Eisenstein et al. (2011) also showed that this formulation of the prior can be extended to additional variables beyond η_0 and η_t. For example, a third term could be added if one wanted to jointly model word associations with topics and an additional property of documents such as timestamp or author. This is the idea of multi-dimensional topic modeling, described in Section 2.4.2.

Finally, another difference between the SAGE prior and the logistic normal prior is that rather than using a normal distribution as the prior for the $\boldsymbol{\eta}$ parameters, SAGE uses a Laplace distribution, which induces sparse parameters, as discussed in Section 2.4.3.

2.3.3.3 Structural topic models

The *structural topic model* (STM) (Roberts et al., 2013) combines the ideas of SAGE and DMR, described in the previous two subsections. Word distributions in STM are modeled the same way as SAGE (Section 2.3.3.2), The topic distributions are similar to DMR

(Section 2.3.3.1) in that each document has its own prior, which is a linear function of document attributes.

The difference between the document priors in STM and DMR is that STM priors use the logistic normal distribution (Section 2.3.2) rather than the Dirichlet distribution. Concretely, for each document, each prior mean μ_{mt} is a function of I document attributes: $\mu_{mt} = \sum_i \alpha_{mi}\lambda_{it}$, where $\boldsymbol{\lambda}$ are regression coefficients. The means are then used in logistic normal priors: $\theta_{mt} \propto \exp(\eta_{mt})$ where $\boldsymbol{\eta}_m \sim \mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma})$

2.4 Structure in Topics

There are many ways in which topic models can be augmented with additional *structure* (Wallach, 2008). For example, topic models might incorporate linguistic structure (Wallach, 2006; Boyd-Graber and Blei, 2008). In this thesis, we will focus on various organizational structures of how topics are related to each other. This section summarizes some of these structures and the models that encode these structures, focusing on structures and models that we will revisit in Chapters 3 and 5.

2.4.1 Topic Correlations and Hierarchies

An important characteristic that can be modeled is the property that certain topics are more or less likely to occur together in documents. For example, a topic about BASEBALL is more likely to occur in a document with a BASKETBALL topic than a HEALTHCARE

topic.

As mentioned in Section 2.3.2, the logistic normal distribution can be used to encode covariance between topics. A topic model with this prior is called the *correlated topic model* (Blei and Lafferty, 2007). Other topic models have also been proposed that include additional structure to learn associations between topics, which we now summarize.

2.4.1.1 Pachinko allocation

The *Pachinko allocation* model (PAM) (Li and McCallum, 2006) includes multiple levels of latent topic variables, with dependencies between the levels.

The standard version of PAM includes two levels of latent topic variables, called "supertopics" and "subtopics". There are typically more subtopics than supertopics, and subtopics correspond to finer-grained concepts. Similar to LDA, each document has a distribution over supertopics, denoted here as $\theta_m^{(1)}$, with a Dirichlet prior $\tilde{\theta}^{(1)}$. Additionally, each document has multiple distributions over subtopics; one such distribution for each supertopic, denoted $\tilde{\theta}_{mt}^{(2)}$. The idea is that the distribution over subtopics changes with the choice of supertopic, and certain supertopics are associated with certain subsets of subtopics. Each token is associated with a latent supertopic and subtopic variable, denoted $z^{(1)}$ and $z^{(2)}$, and the observed word depends on the token's subtopic value.

The generative story for documents under PAM can be described as:

1. For each document m:

   (a) Sample supertopic distribution $\theta_m^{(1)} \sim \text{Dirichlet}(\tilde{\theta}^{(1)})$

(b) For each supertopic t:

i. Sample subtopic distribution $\theta^{(2)}_{mt} \sim \text{Dirichlet}(\tilde{\theta}^{(2)}_{t})$

(c) For each token n in document m:

i. Sample supertopic value $z^{(1)}_{mn} \sim \theta^{(1)}_{m}$

ii. Sample subtopic value $z^{(2)}_{mn} \sim \theta^{(2)}_{mz^{(1)}_{mn}}$

iii. Sample word value $w_{mn} \sim \phi_{z^{(2)}_{mn}}$

A variant of PAM by Mimno et al. (2007) allows both supertopics and subtopics to be associated with word distributions. In general, PAM can contain more levels than just the two (supertopics and subtopics) described here, but this is the most commonly used form of PAM.

These models organize the topics in a structure similar to a hierarchy, with fine-grained subtopics depending on coarse-grained supertopics, although this is not strictly a "hierarchy" in the sense of the term used in Chapter 5 because subtopics depend on multiple supertopics, rather than a single parent. The models can learn that certain topics are likely to co-occur in documents because the selection of a supertopic will increase or decrease the prior likelihood for certain subtopics.

2.4.1.2 Shared components topic models

PAM and the correlated topic model learn associations between topics based on their co-occurrences within documents. An alternative way of learning associations between

topics is to model the property that certain topics have words in common. This is the idea of *shared components topic models* (SCTM) (Gormley et al., 2010).

Under SCTM, each tth topic's word distribution ϕ_t is defined as a product of some number of other word distributions, called "components", and then renormalized to form a new distribution. To be consistent with the notation used in Chapter 5, we will denote each cth component as ω_c, which under SCTM is a distribution over the vocabulary. Each topic is associated with a binary vector β_t of length C, the number of components, which is used to choose which subset of components are multiplied together to form the topic's word distribution.

A topic under SCTM is then defined as:

$$\phi_{tv} \propto \prod_{c=1}^{C} \omega_{cv}^{\beta_{tc}} \tag{2.13}$$

As noted by Gormley et al. (2010), this formulation is closely related to SAGE's formulation of word distributions in Eq. 2.12, where a product of component weights is parameterized as a product of exponentiated log-weights. The SAGE definition in Eq. 2.12 would correspond to SCTM with the $\boldsymbol{\beta}$ matrix defined so that the first column is always 1 and the remaining columns have 1 down the diagonal.

SCTM is a type of *product-of-experts* model (Hinton, 2002), in which different distributions are multiplied together and renormalized to form a joint distribution.

2.4.2 Multi-Dimensional Topics

Multi-dimensional topic models associate each word token with multiple latent variables, rather than just a single "topic" variable, and each token's observed word value is dependent on the values of all of the token's latent variables. This is desirable in corpora that might be modeled with more than one latent factor to explain the choice of words. For example, in a corpus of restaurant reviews, one could imagine a set of latent aspects such as FOOD QUALITY and SERVICE, as well as latent sentiment values of POSITIVE and NEGATIVE. A natural model would associate each word token with a pair of (aspect, sentiment) values, such as (SERVICE, POSITIVE). Indeed, there are many factors that may contribute to a document's word choice: topic, sentiment, author perspective, and others.

Chapter 3 will introduce *factorial LDA* (FLDA), a flexible multi-dimensional model, as a main contribution of this thesis. This subsection will describe other multi-dimensional models.

2.4.2.1 The topic aspect model

The *topic aspect model* (TAM) (Paul and Girju, 2010b) is an LDA-like model under which each word token is associated with *two* latent variables, generically referred to as "topic" and "aspect", and each (topic, aspect) pair indexes into a distribution over words. Each document has a distribution over topics as well as a distribution over aspects. In the generative story, each word is generated by first sampling a topic value from the document's topic distribution and an aspect value from the document's aspect distribution, and then

sampling the word according to the (topic, aspect) pair's word distribution.

Paul and Girju (2010a) used TAM to infer and summarize differing political perspectives, such that the "aspects" of the model corresponded to perspectives. They showed that a "two-dimensional" model like TAM was better able to infer political perspectives from text than a "one-dimensional" model like LDA, even with no supervision.

TAM was extended by this author for the application of mining health information from social media. This extension was called the Ailment Topic Aspect Model (ATAM), under which each health topic also combined with an "aspect" indicating whether a word token was describing a symptom, a medical treatment, or a general word (Paul and Dredze, 2011, 2014).

2.4.2.2 Multi-view topic models

"Cross-collection" or "multi-view" topic models assume a corpus is partitioned into distinct "collections" or "views". The first cross-collection topic model was based on PLSA, introduced by Zhai et al. (2004). Bayesian extensions to this model were independently introduced as *cross-collection LDA* (Paul and Girju, 2009a) and *multi-view LDA* (Ahmed and Xing, 2010).

These models all associate each topic with multiple word distributions, one for each collection or view, as well as a collection-independent word distribution. The collection or view thus acts as a second dimension, similar to TAM, where the word variables are conditioned on both the topic and the collection or view. Unlike TAM, this second dimension

is a variable associated with the document, rather than with each token, and the value is observed as input rather than being latent.

2.4.3 Sparsity in Topic Models

Topic models have a large number of parameters. For improved interpretability, it can be desirable to learn *sparsity* in the parameters, meaning that some parameter values will be zero or close to zero. This subsection describes a few different ways in which topic models can be sparse.

2.4.3.1 Sparsity in the topic distributions

One way in which a topic model can be sparse is to have sparsity in each document's topic distribution, such that only a small number of topics have a non-negligible probability within each document.

Certain hyperparameter settings of the Dirichlet prior over topics can be used to encourage sparsity. As explained in Section 2.1.2, if the Dirichlet precision is less than 1, the density concentrates around the boundaries of the simplex, giving high prior likelihood to sparse topic distributions.

However, even if the prior prefers sparse distributions, it is still possible for the posterior to have high density around dense topic distributions. One technique to address this is to alter the learning objective to favor sparse distributions in the posterior. Balasubramanyan and Cohen (2013) did this by adding a regularizer to the objective that preferred the topic

distributions to have low entropy, thereby inducing sparsity.

Focused topic models (Williamson et al., 2010) take a more explicit approach by directly modeling the property that only a subset of topics have nonzero probability in each document. The topic distributions are augmented with a binary vector that acts as a "mask", zeroing out some of the topics from the document's topic distribution. The document-specific binary vectors are additional random variables in the generative model.

2.4.3.2 Sparsity in the set of topics

The idea of zeroing out a subset of topics can also be applied to the entire corpus, rather than having each document select its own subset of topics. In this scenario, we imagine that parameters for a large number of topics are generated, but then only a smaller subset of these topics can actually be sampled from when generating documents. This type of sparsity is used in factorial LDA (FLDA, introduced in Chapter 3), which is a multi-dimensional model that parameterizes word distributions for every combination of components in the different dimensions. Sparsity is used to select only a subset of the Cartesian product to be sampled from.

2.4.3.3 Sparsity in the word distributions

Similar to sparsity in each document's topic distribution, one might model sparsity within each topic's word distribution. This could allow us to learn that each topic is associated with only a subset of the vocabulary.

Sparse topic models (Wang and Blei, 2009) are similar to focused topic models, where a subset of the vocabulary is explicitly modeled as having zero probability in each topic's word distribution.

Sparsity in word distributions is modeled in a different way in SAGE (Eisenstein et al., 2011) (described in Section 2.3.3.2). In SAGE, word distributions are functions of topic-specific weight parameters that adjust an overall distribution over the vocabulary. The weights have Laplace priors, which serve as sparsity-inducing ℓ_1 regularizers, so that only a subset of the vocabulary will have nonzero weight in each topic's parameters.

2.5 Learning Topic Models

This section will describe learning in latent Dirichlet allocation. The variables in LDA include the observed data $\boldsymbol{w}$, the unobserved data $\boldsymbol{z}$, the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, and the hyperparameters $\tilde{\theta}$ and $\tilde{\phi}$. The values of the latent variables $\boldsymbol{z}$, $\boldsymbol{\theta}$, and $\boldsymbol{\phi}$ must be learned. In a Bayesian setting, this is formulated as a problem of probabilistic inference, in which we compute the distribution over these latent variables conditioned on the observed data $\boldsymbol{w}$ and the priors. This distribution is the posterior:

$$P(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}|\boldsymbol{w}, \tilde{\theta}, \tilde{\phi}) = \frac{P(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}, \boldsymbol{w}|\tilde{\theta}, \tilde{\phi})}{P(\boldsymbol{w}|\tilde{\theta}, \tilde{\phi})} \tag{2.14}$$

One might wish to learn the values of the latent variables that maximize this posterior distribution, i.e., $\arg\max_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{z}} P(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}|\boldsymbol{w}, \tilde{\theta}, \tilde{\phi})$. These maximizing values are called the

maximum a posteriori (MAP) estimate of the model. Alternatively, one might wish to know this entire distribution, not just the maximizing point. This is called *posterior inference*. Inferring the actual posterior is necessary if one wants to find the mean rather than the mode of the posterior, for example.

MAP estimation involves optimization of the joint likelihood function, $P(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{z}, \boldsymbol{w}|\tilde{\theta}, \tilde{\phi})$, or alternatively the distribution $P(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{w}|\tilde{\theta}, \tilde{\phi})$ with the topic assignments $\boldsymbol{z}$ marginalized out, if one is only interested learning the parameters. In either case, the denominator $P(\boldsymbol{w}|\tilde{\theta}, \tilde{\phi})$ need not be computed because it is constant with respect to the variables being optimized. Any continuous optimization algorithm can be used to solve for the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, such as gradient ascent. The posterior density function of LDA is not concave, so standard convex optimization algorithms are only guaranteed to find a local, not global, maximum. A very popular algorithm for MAP estimation is the Expectation Maximization (EM) algorithm (Dempster et al., 1977), in which values of the unobserved data $\boldsymbol{z}$ are estimated conditioned on the current estimates of the parameters, and then the parameters are updated to maximize their posterior conditioned on the currently-inferred values of $\boldsymbol{z}$.

The numerator in Eq. 2.14 is tractable to evaluate for a given set of variable assignments. Computing the actual posterior distribution, however, requires computing the denominator, which integrates over all possible assignments of $\boldsymbol{z}$, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. Computing this is intractable, so the posterior can only be approximated.[3] The two most commonly used methods for approximate posterior inference are variational methods (Jordan et al., 1999)

[3]While we showed in Section 2.1.3 that there is an analytical solution to the posterior of a single multinomial with a Dirichlet prior (Eq. 2.4), there is no such solution when multiple multinomial parameters, θ and ϕ, are coupled (Blei et al., 2003b).

and Markov Chain Monte Carlo (MCMC) methods (Gilks et al., 1995).

Variational methods of inference involve finding an approximation of the posterior within an analytically tractable family of distributions. The idea is to choose a distribution within the simplified family that is closest to the true posterior.

MCMC methods involve generating samples that are provably distributed according to the posterior. Unlike variational inference, MCMC has asymptotic guarantees. The work presented in this thesis uses an MCMC algorithm called Gibbs sampling, so this approach is now described in more detail.

2.5.1 Posterior Inference with Gibbs Sampling

Monte Carlo methods approximate distributions by taking random samples that are distributed according to the target distribution, even if that distribution can't be analytically computed. Markov chain Monte Carlo (MCMC) methods draw samples by simulating a random walk through a Markov chain, where a *state* in the Markov chain corresponds to a configuration of values of the random variables in the model.

If a random walk satisfies certain conditions,[4] the distribution over states will become *stationary*, meaning the distribution stays constant at each time step. If a Markov chain

[4] The conditions are that: (1) the Markov chain is *irreducible*, meaning it must be possible for the random walk to reach any state starting from any other state; (2) all states in the Markov chain are *recurrent*, meaning it is possible for the random walk to return to any state after leaving; and (3) all states in the Markov chain are *aperiodic*, meaning the random walk can return to any state after any number of time steps. (For example, a random walk on a bipartite graph is not aperiodic, because it requires an even number of steps to return to a state.) The easiest way to satisfy all three conditions is to define the transition matrix such that all states have non-zero probability of transitioning to each other (Page et al., 1999).

has a stationary distribution at all, it has a unique stationary distribution, meaning that the distribution over states will converge to the same distribution regardless of what state the random walk started in.

For posterior inference, MCMC methods construct a Markov chain such that the state space corresponds to the possible variable configurations over which we want to learn a posterior, and the chain's stationary distribution *is* the posterior distribution over the variable configurations. By simulating a random walk through such a Markov chain, it is possible to generate random samples which in the limit are guaranteed to be distributed according to the posterior.

The ability to provably sample from the true posterior is a major advantage of MCMC methods. This property justifies sampling as a reasonable approach to take, though in high-dimensional models like LDA, only a small number of samples can be generated relative to the massive size of the state space.

Simplicity is another advantage of sampling relative to other methods of posterior inference, as many MCMC algorithms are relatively easy to derive. *Gibbs sampling* (Geman and Geman, 1984) is a particularly popular MCMC algorithm for topic modeling, first used for LDA by Griffiths and Steyvers (2004). The derivation of a Gibbs sampler follows from the model definition and is often straightforward to construct.

A Gibbs sampler iteratively samples a new value for each random variable[5] from a

[5]The algorithm should randomly pick which variable to sample at each time step. In practice (including in the implementations used in this thesis), however, it is common to step through the variables in a fixed order. If this is done, the Markov chain will not have a stationary distribution because the aperiodicity criterion described in Footnote (4) will be violated. If the random variables are iterated in a fixed order, then the random walk will have a period of N, where N is the number of variables.

distribution conditioned on the currently sampled values of all remaining variables in the model. (Typically only one variable is sampled at a time, though a *blocked* Gibbs sampler jointly samples new values for multiple variables at once.) It can be shown that the samples will be distributed according to the posterior.

For a given model, the conditional distributions from which new values are sampled must be derived. The sampling distribution for LDA is now given.

2.5.1.1 The LDA collapsed Gibbs sampler

In LDA, we wish to infer the posterior distribution over the hidden variables $\boldsymbol{z}$, $\boldsymbol{\theta}$, and $\boldsymbol{\phi}$. This subsection describes how to construct a Gibbs sampler to sample from this posterior.

For LDA, it is typical to use a *collapsed* sampler. In a collapsed Gibbs sampler, rather than sampling values of the parameters, we only sample the latent topic assignments $\boldsymbol{z}$ according to their marginal distribution where the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ have been integrated over. This is possible in LDA because the Dirichlet prior is conjugate to the multinomial distribution, as explained in Section 2.1.3, so there is an analytical solution to the marginalization of the parameters.

The collapsed Gibbs sampler will iteratively sample a topic assignment z_{mn} for each token w_{mn} in the data from a distribution conditioned on all other variables in the model, including the observed data ($\boldsymbol{w}$) and the priors ($\tilde{\theta}$ and $\tilde{\phi}$). For each token w_{mn}, we sample a value (that is, a topic assignment) for the random variable z_{mn} according to the conditional distribution, $P(z_{mn} = t|\boldsymbol{z} - z_{mn}, \boldsymbol{w}, \tilde{\theta}, \tilde{\phi})$. The notation $\boldsymbol{z} - z_{mn}$ refers to the set of all

topic assignment variables excluding z_{mn}. It can be shown that this conditional distribution is proportional to:

$$P(z_{mn} = t|\boldsymbol{z} - z_{mn}, \boldsymbol{w}, \tilde{\theta}, \tilde{\phi}) \propto \left(n_t^m + \tilde{\theta}_t\right) \left(\frac{n_{w_{mn}}^t + \tilde{\phi}_{w_{mn}}}{\sum_v n_v^t + \tilde{\phi}_v}\right) \tag{2.15}$$

where n_t^m is the number of tokens in the mth document currently assigned to the tth topic, and n_v^t is the number of tokens assigned to the tth topic and the vth vocabulary value. These counts exclude the current assignment of z_{mn}.

Gibbs samplers are commonly run for a fixed number of iterations (though other stopping criteria can be considered), where a sampling "iteration" involves sampling a new topic assignment for every token in the corpus. A single iteration therefore involves N steps through the Markov chain, where N is the number of tokens in the corpus.

Conditioned on a set of topic assignments $\boldsymbol{z}$, there is a closed-form solution to the posterior distributions over the parameters, $p(\theta_m|\boldsymbol{z}, \tilde{\theta})$ and $p(\phi_t|\boldsymbol{z}, \tilde{\phi})$. As explained in Section 2.1.3, these posteriors are given by the Dirichlet distribution (Eq. 2.4):

$$\theta_m \sim \text{Dirichlet}(\boldsymbol{n^m} + \tilde{\theta}) \qquad\qquad \phi_t \sim \text{Dirichlet}(\boldsymbol{n^t} + \tilde{\phi}) \tag{2.16}$$

Thus, at each iteration of the Gibbs sampler, we can obtain conditional posteriors over the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. From these posteriors, we can also obtain point estimates of the expected values and maximizing values of these parameters conditioned on the current

topic assignments as the mean and mode of the Dirichlet distribution:

$$\mathbb{E}[\theta_{mt}] \propto n_t^m + \tilde{\theta}_t \qquad \mathbb{E}[\phi_{tv}] \propto n_v^t + \tilde{\phi}_v \tag{2.17}$$

$$\arg\max_{\theta_{mt}} p(\theta_m|\boldsymbol{z}, \tilde{\theta}_t) \propto n_t^m + \tilde{\theta}_t - 1 \qquad \arg\max_{\phi_{tv}} p(\phi_t|\boldsymbol{z}, \tilde{\phi}) \propto n_v^t + \tilde{\phi}_v - 1 \tag{2.18}$$

Typically the parameters are estimated as the means in Eq. 2.17. Each sample of the topic assignments $\boldsymbol{z}$ in the collapsed Gibbs sampler can be used to obtain posteriors over the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, conditioned on $\boldsymbol{z}$.

2.5.1.1.1 DERIVING THE SAMPLING DISTRIBUTION

The sampling distribution defined in Eq. 2.15 is distribution over the random variable z_{mn} conditioned on all other variables in the model. That is,

$$P(z_{mn}|\boldsymbol{z} - z_{mn}, \boldsymbol{w}, \tilde{\theta}, \tilde{\phi}) = \frac{P(\boldsymbol{z}, \boldsymbol{w}|\tilde{\theta}, \tilde{\phi})}{P(\boldsymbol{z} - z_{mn}, \boldsymbol{w}|\tilde{\theta}, \tilde{\phi})} \tag{2.19}$$

The joint $P(\boldsymbol{z}, \boldsymbol{w}|\tilde{\theta}, \tilde{\phi})$ can be factored as $P(\boldsymbol{z}|\tilde{\theta})P(\boldsymbol{w}|\boldsymbol{z}, \tilde{\phi})$. Recall that $P(\boldsymbol{z}|\tilde{\theta})$ and $P(\boldsymbol{w}|\boldsymbol{z}, \tilde{\phi})$ have already been defined as Eqs. 2.10–2.11 in terms of the Dirichlet-multinomial distribution. Plugging in these distributions and using the property that $\Gamma(x) = x\Gamma(x-1)$ leaves us with Eq. 2.15 after canceling terms.

2.5.1.1.2 WHAT IS THE SAMPLER LEARNING?

As an MCMC algorithm, a Gibbs sampler is simulating a random walk through a Markov chain whose stationary distribution corresponds to the posterior distribution over topic assignments. The Markov chain is traversed for cN time steps by the LDA collapsed Gibbs sampler (where N is the number of tokens), where c typically ranges from 1,000 to 10,000 for topic models. This is a minuscule fraction of the T^N states in the Markov chain (where T is the number of topics), which is the number of possible configurations of the topic variables.

Clearly, only a small portion of the posterior will thus be explored. While the samples will not form a close approximation to the entire posterior, they will form a reasonable approximation of some high-density region of the posterior.

This is useful behavior, because the sampler will discover (1) parameters that give high likelihood to the data (if not global maximizers); (2) parameters that give less-high likelihood, as alternatives to the best set of parameters; and (3) a distribution over the different parameter sets which approximates a local slice of the posterior, providing the user with information about the certainty of different sets of parameters.

Since only a local approximation of the posterior is learned, it is common to learn multiple models by running multiple Gibbs samplers with different random initializations. Typically a single model among the multiple trials is selected, usually based on its predictive abilities, as described in Section 2.6. Recent research has shown experimentally that many topics will be consistently learned by multiple Gibbs samplers, while others are

sensitive to the initialization (Chuang et al., 2015).

2.5.1.2 Gibbs sampling as stochastic optimization

A Gibbs sampler infers an approximate posterior (in the form of samples from the true posterior) which contains interesting information about different sets of parameters, but this information is not often provided to the user, and instead only a statistic of the posterior (such as the mean or mode) is used. Most commonly, the samples from the posterior are averaged to form a single point estimate of the posterior mean.[6] While not taking advantage of the full posterior, there is still an empirical reason to do this over MAP optimization techniques: when using the model to make predictions about new data, some experiments have shown that the mean of the approximated posterior is more robust than the locally optimal MAP estimate (Blei et al., 2003b).[7]

However, there is not always a clear difference between posterior inference via sampling and MAP estimation via optimization algorithms. As observed by Mimno (2012b), the way in which Gibbs samplers are sometimes used more closely resembles MAP estimation, when only a single high-probability sample is collected from the Gibbs sampler. It is in fact quite common to find instances of topic modeling in which only a single sample is used from the Gibbs sampler, and so often the goal appears to be to find a MAP esti-

[6] Combining samples is only reasonable to do if we assume the random walk will stay near one local mode of the posterior. Because topic models are not identifiable, the posterior has equally likely modes for all $T!$ permutations of the topic indices (at least under a symmetric prior), and averaging samples that correspond to different topic index permutations would not be meaningful.

[7] However, some experiments have found MAP estimates to be comparable to posterior means (Asuncion et al., 2009; Taddy, 2012).

mate rather than to infer the posterior, even if this goal is not explicitly stated. The Gibbs sampling algorithm can be used to find an approximate MAP estimate by taking the single sample that gives the highest likelihood among all variable configurations explored.

Viewed this way, Gibbs samplers are being used as a stochastic optimization algorithm for finding the MAP solution, and this optimization algorithm happens to work better than deterministic optimization algorithms like EM. Indeed, some early topic modeling papers motivated the use of Gibbs sampling by stating that it avoids converging to local optima, in contrast to EM (Griffiths and Steyvers, 2004; Li and McCallum, 2006).

There is also an interesting connection to stochastic optimization even when we average multiple samples to estimate the posterior mean. Parameter averaging has been shown to be an effective technique for common optimization algorithms including stochastic gradient descent (Ruppert, 1991; Polyak and Juditsky, 1992) and Perceptron (Collins, 2002). In these averaging variants, rather than using the single set of parameters that minimize the loss, a final set of parameters is produced by taking an average of the parameters from each iteration of the optimization algorithms. Collins (2002) explains that by averaging multiple sets of parameters before making predictions, this technique approximates a "voting" algorithm that averages the predictions made by each set of parameters, and this can lead to more robust predictions on unseen data (Freund and Schapire, 1999). This is an alternative explanation of why predictiveness of new data can be improved by averaging multiple samples from the approximated posterior.

2.5.2 Hyperparameter Learning

The posterior inference procedures described above treat the hyperparameters $\tilde{\theta}$ and $\tilde{\phi}$ as observed. Often the values of these hyperparameters are provided by the user as input. If the prior is set by the user, then typically symmetric priors are chosen, such that $\tilde{\theta}_t = \tilde{\theta}_{t'}$ and all $\tilde{\theta}_v = \tilde{\theta}_{v'}$.

The predictive performance of LDA can be affected by the choice of hyperparameters. Often users must try different values of hyperparameters and select the best model.

An alternative to setting the hyperparameters in advance is to infer them from the data. As with the other parameters of the model, the hyperparameters can be treated as random variables and have their posterior inferred. The same methods of posterior inference described above can be applied; for example, the hyperparameters can be sampled (Wallach, 2008). Rather than full posterior inference, however, a more common practice is to perform MAP estimation with optimization algorithms, which has been empirically shown to be a competitive approach to sampling the hyperparameter values (Wallach et al., 2009a). When hyperparameters are optimized, the approach is called *empirical Bayes* (Robbins, 1956).

Minka (2003) and Wallach (2008) describe various approaches for learning Dirichlet hyperparameters. For the models introduced in this thesis, we use gradient-based optimization for hyperparameters.

Our inference algorithm alternates between Gibbs sampling (conditioned on the priors) and optimization of the priors (conditioned on the samples). This forms a *Monte Carlo EM* algorithm, first used in topic modeling by Wallach (2006). The E-step involves estimating

the expected value of the latent variables by averaging samples from the posterior using Gibbs sampling. The M-step solves for the MAP estimate of the hyperparameters conditioned on these expected values. The algorithm used in this thesis does not completely optimize the hyperparameters in each M-step. Instead, we apply just one iteration of gradient ascent after each Gibbs sampling iteration (that is, a new value has been sampled for every random variable in the model). This approach changes the values gradually.

Note that the posterior distribution over $\boldsymbol{z}$, $\boldsymbol{\theta}$, and $\boldsymbol{\phi}$ is conditioned on $\tilde{\theta}$ and $\tilde{\phi}$, so if the hyperparameters change, the posterior changes. This means that Gibbs samples that are collected from iterations with different values of $\tilde{\theta}$ and $\tilde{\phi}$ are not sampled from the same conditional distribution. In this thesis, we only collect samples near the very end of the inference procedure, after the hyperparameters have stabilized. This avoids the "label switching" issue described in Footnote (6).

2.5.2.1 Optimization with gradient ascent

The hyperparameter optimization algorithms in this thesis apply a step of gradient ascent after each sampling iteration. Gradient ascent optimization uses the update rule: $\boldsymbol{x}^{i+1}=\boldsymbol{x}^{i} + \eta_t \nabla \mathcal{L}(\boldsymbol{x}^{i})$, for some vector $\boldsymbol{x}$, where $\boldsymbol{x}^{i}$ is the value at iteration i, η_i is some step size at iteration i, and $\mathcal{L}(\boldsymbol{x})$ is the log-likelihood of $\boldsymbol{x}$.

To optimize the hyperparameters of the topic distributions, we use the partial derivative

of $P(\boldsymbol{z}|\tilde{\theta})$ (Eq. 2.10) with respect to each $\tilde{\theta}_t$:

$$\frac{\partial \log P(\boldsymbol{z}|\tilde{\theta})}{\partial \tilde{\theta}_t} = \sum_{m=1}^{M} \Psi(n_t^m + \tilde{\theta}_t) - \Psi(\tilde{\theta}_t) + \Psi(\textstyle\sum_{t'=1}^{T} \tilde{\theta}_{t'}) - \Psi(\textstyle\sum_{t'=1}^{T} n_{t'}^m + \tilde{\theta}_{t'}) \quad (2.20)$$

where Ψ is the derivative of the log of the gamma function Γ, called the digamma function.

Similarly, to optimize the hyperparameters of the topic distributions, we use the partial derivative of $P(\boldsymbol{w}|\boldsymbol{z}, \tilde{\phi})$ (Eq. 2.11) with respect to each $\tilde{\phi}_v$:

$$\frac{\partial \log P(\boldsymbol{w}|\boldsymbol{z}, \tilde{\phi})}{\partial \tilde{\phi}_v} = \sum_{t=1}^{T} \Psi(n_v^t + \tilde{\phi}_v) - \Psi(\tilde{\phi}_v) + \Psi(\textstyle\sum_{v'=1}^{V} \tilde{\phi}_{v'}) - \Psi(\textstyle\sum_{v'=1}^{V} n_{v'}^t + \tilde{\phi}_{v'}) \quad (2.21)$$

In later chapters, the hyperparameters $\tilde{\theta}$ and $\tilde{\phi}$ will be replaced with functions of other parameters, but the derivatives will have a similar form.

2.5.3 Scaling Sampling

A challenge with applying topic models to corpora from the Web is scaling the inference algorithms to potentially massive datasets. This subsection describes faster variants of the standard Gibbs sampling algorithm for LDA.[8]

In the LDA Gibbs sampler, computing and sampling from the distribution over topic assignments (Eq. 2.15) requires $O(T)$ time, growing linearly with the number of topics T. This computation must be performed for all N tokens in the corpus, and therefore each

[8] Other research has worked on scaling other inference algorithms, such as variational inference (Hoffman et al., 2010).

Gibbs sampling iteration requires $O(NT)$ time. The complete Gibbs sampling algorithm requires $O(INT)$ time, where I is the number of sampling iterations. The complexity of each of these factors—the number of iterations, the number of tokens, and the number of topics—can be improved using techniques now described.

The number of iterations can be reduced using online learning and streaming algorithms. *Streaming* algorithms seek to iterate over the data only a single time, so $I = 1$. Streaming Gibbs sampling algorithms were first proposed for LDA by Canini et al. (2009) using particle filtering (Doucet et al., 2001). This approach was extended by Zhai and Boyd-Graber (2013) to allow the vocabulary to grow with new data. The memory complexity of the particle filtering algorithm was later reduced to constant time (from linear) by May et al. (2014) using reservoir sampling (Vitter, 1985). A drawback of these streaming algorithms is that they have been shown experimentally to perform worse than the standard batch samplers (Canini et al., 2009).

The amount of data that must be sampled by a processor can be reduced by using *distributed* algorithms, such that each processor only needs to sample tokens from a small partition of the full dataset. However, since Gibbs samplers are inherently sequential, they are not trivial to parallelize. Newman et al. (2009) presented two methods for distributed Gibbs sampling in LDA. The first method is to simply run Gibbs samplers independently on different processors for different partitions of the data. After each iteration, the counts of topic assignments are updated across all partitions. Since the processors only synchronize once per iteration (after N new topic assignments have been made), this algorithm

does not correctly sample from the true posterior. The authors suggested another approach in which the LDA model is adjusted to account for the distributed learning environment, such that each partition has its own set of parameters drawn from a globally shared prior. Experimentally, the simpler approximate version performed as well as the corrected model, so this approach has been widely adopted in practice.

Finally, another line of work has been to reduce the complexity of sampling new values for each token. Porteous et al. (2008) presented an algorithm that takes advantage of the fact that the sampling distribution for each token is sparse, meaning most of the probability mass is concentrated on a small number of topics. The algorithm computes the sampling distribution for only the most probable topics, and can potentially terminate without having to compute this for all T topics. The algorithm relies on bounds on the distribution to ensure that it does not prune away topics that it should have considered, and is guaranteed to sample from the true distribution. Similar ideas are used in the algorithm proposed by Yao et al. (2009), which decomposes the sampling distribution into different terms and only computes certain portions as necessary. The authors show that many of the computations can be cached, and the resulting algorithm is even faster than Porteous et al.'s. More recently, Li et al. (2014) showed that one can create an array, called an alias table (Walker, 1977), such that randomly picking an element from the array (which can be done in constant time) corresponds to sampling from a distribution. Applying this method to the LDA Gibbs sampler results in an algorithm whose time complexity grows only with the number of topics observed in a document, rather than the total number of topics T.

The topic models introduced in this thesis—Factorial LDA and SPRITE—use Gibbs samplers that are identical to the LDA Gibbs sampler once conditioned on the hyperparameters. This means that any of these techniques described in this subsection could be applied to these models as well. In Chapter 6, our implementation of SPRITE uses the approximate distributed algorithm of Newman et al. (2009) when applying the model to large social media corpora.

2.6 Evaluating Topic Models

It is important to be able to evaluate how "good" a topic model is, for some definition of goodness. This section will discuss possible definitions of goodness.

The purpose of a goodness metric is typically to choose among multiple models for a particular task and dataset. For example, one might need to decide whether to use LDA or a richer topic model, and one must therefore know which type of model is "better". Even within a single type of model, one often must choose between different parameterizations of the model, for example because multiple Gibbs samplers were run with different random initializations, or because multiple models were learned with different settings for the number of topics T. The process of choosing between competing models is called *model selection*.

The two main methods for evaluating topic models are: (1) measuring how well the model can *predict* data, and (2) measuring how *coherent* or interpretable the topics are. We

describe these methods now.

2.6.1 Model Predictiveness

A common definition of the goodness of a model is how high a likelihood it assigns to observed data, such as the text being modeled or auxiliary data that we think the model should be able to explain. This subsection describes this type of evaluation.

2.6.1.1 Predicting text

A standard method for evaluating any probabilistic model is to measure how well it can explain unseen data. In the case of topic models, the data is the text $\boldsymbol{w}$, and the likelihood of the text $P(\boldsymbol{w}|\boldsymbol{\theta}, \boldsymbol{\phi})$ can be used as an evaluation metric. Likelihood should be measured on held-out text, using parameters learned from the training corpus. Models that give higher likelihood to held-out text are said to be more predictive.

The likelihood is often reported in terms of the model *perplexity*, which is a monotonic function of likelihood, $\exp(-\frac{1}{N}\log P(\boldsymbol{w}|\boldsymbol{\theta}, \boldsymbol{\phi}))$ where N is the number of tokens in the corpus. Lower perplexity corresponds to higher likelihood. The advantage of perplexity as a measurement is that it is normalized to the size of the data and can thus be compared across different datasets.

One reason to measure the ability to predict unseen text is that this is sometimes the end task. For example, if one is building a model to predict the next word a user will type on a mobile device, then this is a direct way to evaluate a model's ability to perform that

task. More often, however, likelihood is measured as a general, task-independent form of evaluation, based on the assumption that the more accurately a model can predict unseen text, the more likely it is to be predictive in general.

Different measurements of perplexity have been proposed for topic models, as surveyed by Wallach et al. (2009b). The methods differ in the treatment of $\boldsymbol{\theta}$, the document-specific topic distributions, which must be estimated (or less commonly, marginalized over) for unseen documents before likelihood can be evaluated.

Perhaps the most straightforward method is to infer θ_m (for example, using Gibbs sampling) for each unseen document conditioned on the parameters $\boldsymbol{\phi}$ inferred from the training data, and use the mean of the estimated θ_m to compute the document log-likelihood. This method assumes that θ_m can be estimated before observing the document, which overstates the likelihood of the text. The "left-to-right" evaluation introduced by Wallach (2008) incrementally measures the likelihood of each token conditioned on the inferred topic assignments of tokens to the left of the current token in each document. In this method, the likelihood is calculated from data that was not used to estimate θ_m, which avoids the problem with the naive approach. A related method is the "document completion" evaluation used by Rosen-Zvi et al. (2004), in which θ_m is estimated from tokens in half of a document, and likelihood is measured on the remaining half. The perplexity experiments in this thesis use the document completion method; specifically, we train models on even-numbered tokens and evaluate on odd-numbered tokens.

2.6.1.2 Predicting metadata

Another type of predictive task is to use the inferred topic distributions to make predictions about other attributes associated with documents beyond the text. For example, if each mth document is associated with a discrete or continuous label y_m, one could train a classifier or regression model to predict $\boldsymbol{y}$ using the topic distributions $\boldsymbol{\theta}$ as features, as a low-dimensional alternative to bag of words features. The classification accuracy or regression error can be then used as a metric for comparing topic models. This type of evaluation was used in the original LDA paper of Blei et al. (2003b), and has been used frequently since then, including in the experiments in this thesis.

As with measuring perplexity of the text, one might be interested in measuring the ability to predict document labels either because it is the intended task of the user, or because it is a useful general-purpose evaluation that demonstrates the degree to which the model is learning generalizable concepts.

2.6.2 Topic Interpretability and Coherence

If topic models are used as exploratory tools, it is important that the inferred topics are meaningful to humans. Indeed, sometimes inferred topics are assumed to be so meaningful that they are directly used in analyses, for example to visualize historical trends in scientific topics (Hall et al., 2008) or literary topics Mimno (2012a). In such cases, it is important to be able to choose models that have high interpretability. This subsection describes various

methods for evaluating interpretability.

2.6.2.1 Human judgments of quality

Perhaps the best way to measure the interpretability of a topic model is to directly ask humans. Topic model researchers have therefore conducted a variety of quality evaluation experiments using human subjects.

One approach is to ask subjects to rate the quality of a topic, e.g., on a Likert scale (Newman et al., 2010; Paul and Dredze, 2012a), or to vote on topics aligned across models (Li and McCallum, 2006; Paul and Girju, 2009a). This is a direct measurement of topic quality, but a subjective one that should be averaged across judgments of many different annotators.

A less subjective evaluation is to ask subjects to perform a task whose difficulty depends on the topic quality. This is the idea behind *intrusion* tasks for topic models. Chang et al. (2009) introduced two types of intrusion tasks.

The first (and perhaps more widely adopted) is *word intrusion*, in which subjects are shown in random order several top words from one topic, and a top word from a different, randomly selected topic. Subjects are asked to identify the word from the "intruding" topic. If the topics are coherent, it should be obvious which word came from a different topic. If the topics are noisy, subjects will perform poorly, and thus performance at this task is indicative of the interpretability of the topics.

The second task is *topic intrusion*, in which users are shown a document and the most

probable topics within that document along with a topic that has low probability in the document. As with word intrusion, the task is to identify the low-probability topic.

Interestingly, Chang et al. (2009) showed that human judgments of quality across different models were in fact negatively correlated with perplexity measurements, which means that perplexity measurements are not a good proxy for topic quality. This thesis therefore evaluates the topic models in Chapters 3 and 5 with measures of both predictiveness and coherence.

2.6.2.2 Automatic coherence metrics

It is time-consuming and costly to conduct experiments with human subjects. As an alternative, researchers have proposed automatic metrics of topic coherence that have been shown to be highly correlated with human judgments.

Most coherence metrics measure the degree to which the words that appear within a topic also co-occur together in documents. If words rarely co-occur, they are probably unrelated, and should not be grouped in the same topic. The coherence of individual topics can be measured (e.g., for identifying and removing incoherent topics), and the average coherence across all topics can be used to evaluate the topic model overall.

We will now define the widely-used coherence metric of Mimno et al. (2011), which is what we use in our SPRITE evaluation in Chapters 5–6. Under this metric, the coherence

of the tth topic, $Coher(t)$, is defined as:

$$Coher(t) = \sum_{l=2}^{L} \sum_{j=1}^{l-1} \log \frac{DF(v_{tl}, v_{tj}) + 1}{DF(v_{tj})} \tag{2.22}$$

where $DF(v, w)$ is the document frequency of words v and w (the number of documents in which they both occur), $DF(v)$ is the document frequency of word v, and v_{ti} is the ith most probable word in topic t, L is the number of words considered in the topic (typically $L = 20$), and the 1 in the numerator is a smoothing coefficient.

If the document frequency counts were normalized to probabilities, this is equivalent to summing the log conditional probability of each of the top M words given each higher-ranked word. The intuition is that this score will be higher if the pairs of words in a topic are likely to co-occur.

A very similar metric was proposed by Newman et al. (2010), but used pointwise mutual information (PMI) rather than log conditional probabilities, which would add a $DF(v_{tl})$ term to the denominator of the equation above (if the frequencies were normalized). Mimno et al. (2011) found their variant to be better correlated with human judgments than PMI.

Lau et al. (2014) and Röder et al. (2015) provide summaries and empirical comparisons of additional coherence metrics.

2.6.2.3 Summarization quality

Another use of topic models that requires good interpretability is for document *summarization* (Goldstein et al., 2000), in which short text is generated to describe and summarize the key ideas in a large corpus. In particular, *extractive* summarization is when segments of text from a corpus are extracted to be representative of the corpus. Topic models can be used for extractive summarization by identifying segments of text that have high probability under different topics (Titov and McDonald, 2008; Haghighi and Vanderwende, 2009; Dredze et al., 2008). A useful way to evaluate topic model quality is to evaluate the quality of the "summaries" extracted by a topic model. This approach is used in Chapter 4.

Extracted summaries can either be evaluated intrinsically, by asking human annotators to rate the quality of the summaries, or by comparing the extracted summaries to existing "reference" summaries written by humans. When comparing generated summaries to reference summaries, one can either use human judgments to rate the comparative quality, or one can use automated metrics, such as the ROUGE score (Lin, 2004), which measures n-gram overlap between the two summaries.

Chapter 3

Factorial Latent Dirichlet Allocation

This chapter introduces an extension to LDA called *factorial LDA* (FLDA), which is a multi-dimensional topic model, as described in Section 2.4.2. This chapter is based on material from from Paul and Dredze (2012a).

Standard topic models such as LDA implicitly model a single factor, usually interpreted as thematic content (Blei et al., 2003b), while FLDA can model an arbitrary number of factors. To use the example from Section 2.4.2, in a corpus of restaurant reviews, one might imagine a set of latent aspects such as FOOD QUALITY and SERVICE, as well as latent sentiment values of POSITIVE and NEGATIVE. FLDA would then provide the ability to learn word distributions for pairs of (aspect, sentiment) values, such as (SERVICE, POSITIVE). While standard topic models associate each word token with a single latent topic variable, FLDA can associate each token with a latent vector of multiple factors, such as (aspect, sentiment) in reviews or (topic, political ideology, author gender) in political commentaries.

We extend LDA to this factored setting with the use of *structured priors*, defining the Dirichlet distributions of LDA as functions of additional parameters that are used to enforce the multi-dimensional structure of FLDA.

A key challenge in this model is to ensure consistency across different word distributions which have the same components. For example, the word distributions associated with the (aspect, sentiment) pairs (SERVICE,POSITIVE) and (SERVICE,NEGATIVE) should both give high probability to words about restaurant service, while (SERVICE,POSITIVE) and (FOOD QUALITY,POSITIVE) should both give high probability to positive sentiment words. We address this issue by formulating the priors over word distributions in a way that ties these related distributions together, such that the prior for a pair's word distribution is a function of common parameters for each of the two components of the pair (or more generally, for K components of a K-tuple).

Another hurdle is that as we increase the number of factors in a multi-dimensional model, we have an exponential increase in the number of word distributions assumed by the model, and not all of these will be well-supported in a corpus, as there will not be enough data to infer an exponentially increasing number of parameters. We address this in a modification to FLDA that incorporates sparsity in the set of possible tuples, allowing the model to avoid learning word distributions for tuples that are unlikely to appear in a corpus. This modification is similarly incorporated via a structured prior that reduces the prior probability of certain tuples in documents.

We will demonstrate through experiments that both of these prior structures lead to

improvements in performance.

In the next section, we introduce the general form of factorial LDA. We then discuss our inference procedure (Section 3.2) and share experimental results evaluating FLDA in a general setting (Section 3.3). The next chapter (Chapter 4) will describe extensions to FLDA focused on specific health science applications.

3.1 Model Definition

Recall that LDA assumes we have a set of T latent components (called "topics" in the context of text modeling), and each document has a discrete distribution over these topics, parameterized by θ_m for the mth document. Each word token w_{mn} is associated with a topic variable z_{mn}, and each topic value points to a discrete distribution over words, parameterized by ϕ_t for the tth topic.

Under LDA, a document is generated by choosing the topic distribution θ_m from a global Dirichlet prior. For each token we sample a topic value t from this distribution before sampling a word w from the tth word distribution ϕ_t. Without additional structure, LDA tends to learn distributions which correspond to semantic topics (such as SPORTS or ECONOMICS) (Chang et al., 2009) which dominate the choice of words in a document, rather than sentiment, perspective, or other aspects of document content.

Imagine that instead of a one-dimensional vector of T topics, we have a two-dimensional matrix of T_1 components along one dimension and T_2 components along the other. This

structure is desired if we assume a corpus is composed of two different factors, and the two dimensions might correspond to factors such as news topic and political perspective (if we are modeling newspaper editorials), or research topic and discipline (if we are modeling scientific papers). Individual cells of the matrix would correspond to pairs such as (ECONOMICS,CONSERVATIVE) or (GRAMMAR,LINGUISTICS) and each is associated with a word distribution $\phi_{\vec{t}}$ for the 2-tuple $\vec{t}$.

Let us expand this idea further by assuming K factors modeled with a K-dimensional array, where each cell of the array points to a word distribution corresponding to that particular K-tuple. For example, in addition to topic and perspective, we might want to model a third factor of the author's gender in newspaper editorials, yielding triples such as (ECONOMICS,CONSERVATIVE,MALE). Conceptually, each K-tuple $\vec{t}$ functions as a topic in LDA (with an associated word distribution $\phi_{\vec{t}}$) except that K-tuples imply a structure, e.g., the pairs (ECONOMICS,CONSERVATIVE) and (ECONOMICS,LIBERAL) are related. This is the idea behind *factorial LDA* (FLDA).

At its core, our model follows the basic template of LDA, but θ_m is over elements of a multi-dimensional array rather than a one-dimensional vector. Under FLDA, each document has a distribution θ_m over K-tuples, specifically, over cells in the K-dimensional array where each dimension has T_k components. That is, there are K *factors*, and each factor contains multiple *components*. An example of a type of factor is sentiment, while POSITIVE and NEGATIVE are examples of components within the sentiment factor.

For each document, we again sample the distribution θ_m from a Dirichlet prior. For each

token, we sample a value for each of the K latent components—that is, we sample a full K-tuple $\vec{t}$. Once the tuple is chosen, we sample a word from the tuple's corresponding word distribution $\phi_{\vec{t}}$. As we've described it so far, this is equivalent to the standard LDA model with $\prod_k^K T_k$ topics. However, LDA is unlikely to learn parameters that actually reflect the multi-dimensional concept described above—why would word distributions relate to multiple concepts? We do this by forcing the distributions to reflect multiple dimensions such that the priors are functions of the different factors. We modify FLDA by adding structure to the Dirichlet priors of both the tuple distributions for each document and the word distributions for each topic. We modify the priors in the following ways to induce a factored structure:

- We model the intuition that tuples which share components should share other properties. For example, we expect the word distributions for (ECONOMICS,CONSERVATIVE) and (ECONOMICS,LIBERAL) to both give high probability to words about economics, while the pairs (ECONOMICS,LIBERAL) and (ENVIRONMENT,LIBERAL) should both reflect words about liberalism.

- Similarly, we want each document's distribution over tuples to reflect the same type of consistency. If a document is written from a liberal perspective, then we believe that pairs of the form (*,LIBERAL) are more likely to have high probability than pairs with CONSERVATIVE as the second component.

This consistency across factors is encouraged by sharing parameters across the word

and topic prior distributions in the model: this encodes our *a priori* assumption that distributions which share components should be similar. The prior for each tuple's word distribution is a function of parameters for each of the components of the tuple, and the prior for each document's tuple distribution is similarly a function of each tuple's components.

As in Sections 2.2 and 2.3.1, we will define the model formally by describing the story behind the process that generates data under this model. The generative story for FLDA is as follows (we explain the priors afterward):

1. Draw the various hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$ from 0-mean normal distributions
2. For each tuple $\vec{t} = (t_1, t_2, \ldots, t_K)$:
   (a) Let $\tilde{\phi}_{\vec{t}v} = \exp\left\{\omega^{(B)} + \omega_v^{(0)} + \sum_k \omega_{t_k v}^{(k)}\right\}$ for each word v
   (b) Sample word distribution, $\phi_{\vec{t}} \sim$ Dirichlet($\tilde{\phi}_{\vec{t}}$)
3. For each document m:
   (a) Let $\tilde{\theta}_{m\vec{t}} = \exp\left\{\alpha^{(B)} + \left(\sum_k \alpha_{0t_k}^{(k)} + \alpha_{mt_k}^{(k)}\right)\right\}$ for each tuple $\vec{t}$
   (b) Sample distribution over tuples, $\theta_m \sim$ Dirichlet($\tilde{\theta}_m$)
   (c) For each token n in document m:
      i. Sample component tuple $\vec{z}_{mn} \sim \theta_m$
      ii. Sample word $w_{mn} \sim \phi_{\vec{z}_{mn}}$

The graphical model is illustrated in Figure 3.2(a). Figure 3.1 shows an illustration of how the weight vectors $\boldsymbol{\omega}^{(0)}$ and $\boldsymbol{\omega}^{(k)}$ are combined to form $\tilde{\phi}$ for a particular tuple that

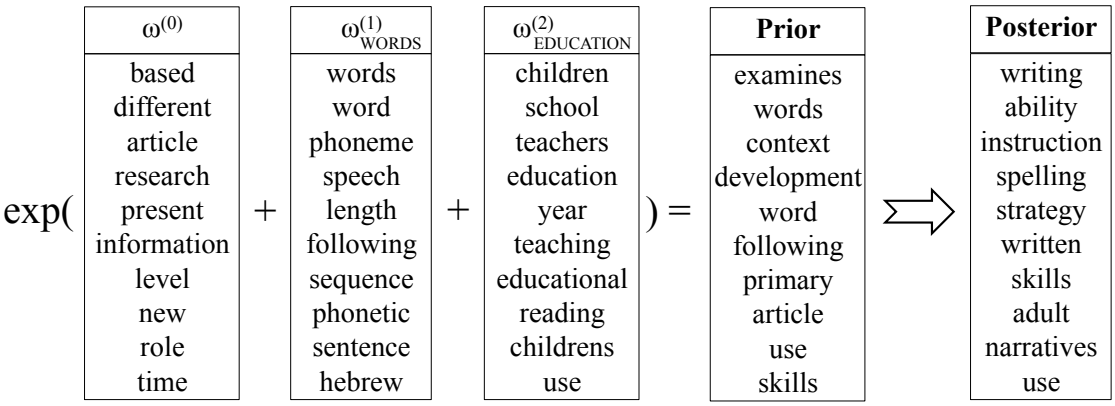


Figure 3.1: An illustration of word distributions in FLDA with two factors when applying FLDA to a collection of scientific articles from various research disciplines, including linguistics and education research (using the CLEP dataset described in Section 3.3). We learn weights ω corresponding to a topic we call WORDS and the discipline EDUCATION as well as background words. These weights are combined to form the Dirichlet prior, and the distribution for (WORDS,EDUCATION) is drawn from this prior: this distribution appears to describe writing education. When referring to the "prior" and "posterior" in this figure, we more concretely are referring to the mean of the prior and the mean of the posterior.

was inferred by our model. The words shown have the highest weight after running our inference procedure, using the data described in the experiments section (Section 3.3).

We assume all $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ parameters are independent and normally distributed around 0, which is equivalent to ℓ_2 regularization during optimization. Each type of parameter ($\omega^{(B)}$, $\omega^{(0)}$, $\omega^{(k)}$, $\alpha^{(B)}$, $\alpha_0^{(k)}$, $\alpha_m^{(k)}$) can have a different variance for the normal prior, which is a user-defined parameter in the experiments used in this thesis.

The next subsection explains these new priors in more detail.

3.1.1 Interpreting the Parameters

As discussed above, the difference between FLDA and LDA is that structure has been added to the Dirichlet priors for the word and topic distributions. We use a form of Dirichlet-multinomial regression (Mimno and McCallum, 2008) (described in Section 2.3.3.1) to formulate the priors for $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ in terms of the log-linear functions defined the generative story. Unlike in LDA, each document has its own prior $\tilde{\theta}_m$ and each word distribution has its own prior $\tilde{\phi}_{\vec{t}}$, rather than globally shared priors. We will now describe these priors in more detail.

3.1.1.1 Prior over word distributions

We formulate the priors of $\boldsymbol{\phi}$ to encourage word distributions to be consistent across components of each factor. Intuitively, our *a priori* assumption is that a word distribution for a particular tuple should have commonalities with other tuples that share the same components. To achieve this goal, we link the priors for tuples that share common components by utilizing a log-linear parameterization of the Dirichlet prior over $\phi_{\vec{t}}$. By placing restrictions only in the prior, we allow the posterior over $\phi_{\vec{t}}$ to deviate and learn interactions among factors.

Formally, we place a Dirichlet($\tilde{\phi}_{\vec{t}}$) prior over $\phi_{\vec{t}}$, the word distribution for tuple $\vec{t} = (t_1, t_2, \ldots, t_K)$. The Dirichlet vector $\tilde{\phi}_{\vec{t}}$ controls the mean and precision of the prior. It is a function of three types of hyperparameters:

- $\omega^{(B)}$ is a global bias scalar which adjusts the overall precision of the Dirichlet.

- $\boldsymbol{\omega}^{(\mathbf{0})}$ is a vector over the vocabulary which reflects the relative prior likelihood of different words. These values adjust the mean of the Dirichlet in addition to the precision and adjust the default prior likelihood of each word in every word distribution.

- Most importantly, each $\omega^{(k)}_{t_k v}$ is a weighting parameter for the vth word for component t_k of the kth factor. By increasing the weight of a particular $\omega^{(k)}_{t_k v}$, we increase the prior of word v in $\phi_{\vec{t}}$ for all $\vec{t}$ that contain component t_k. It is through these parameters the word distributions for different tuples with shared components are tied together.

As an example, consider the effect of this prior on a two-dimensional model for research topic and scientific discipline. For the first factor (topic), we create T_1 vectors $\boldsymbol{\omega}^{(\mathbf{1})}$ that reflect the prior for words in each topic, and similarly, T_2 vectors $\boldsymbol{\omega}^{(\mathbf{2})}$ for each discipline. The tuple (BILINGUALISM,LINGUISTICS) would then combine the corresponding vectors $\boldsymbol{\omega}^{(\mathbf{1})}_{\text{BILINGUALISM}}$ for the topic and $\boldsymbol{\omega}^{(\mathbf{2})}_{\text{LINGUISTICS}}$ for the discipline, along with $\omega^{(B)}$ and $\omega^{(0)}$ to obtain a tuple-specific prior. The distribution for this pair would more likely include words common to both BILINGUALISM and LINGUISTICS, but could also reflect words especially prominent under this pairing (e.g., "acquisition", as in language acquisition).

3.1.1.2 Prior over topic distributions

We use a similar formulation for the prior over $\boldsymbol{\theta}$. Recall that we want documents to naturally favor tuples that share components, i.e., favoring both (ECONOMICS,CONSERVATIVE) and (EDUCATION,CONSERVATIVE) if the document favors CONSERVATIVE in general.

To address this, we let θ_m be drawn from Dirichlet($\tilde{\theta}_m$), where instead of a corpus-wide prior, each document has a vector $\tilde{\theta}_m$ which reflects the independent contributions of the factors via a log-linear function. This function contains three types of hyperparameters:

- $\alpha^{(B)}$ is a global bias scalar which adjusts the overall precision of the Dirichlet.
- $\alpha_{0t_k}^{(k)}$ indicates the bias for the kth factor's component t_k across the entire corpus, which enables the model to favor certain components *a priori*, as they appear in any tuple across the entire corpus. These values adjust the mean of the Dirichlet priors. Modifying these parameters can favor tuples that include e.g., the discipline EDUCATION across all documents.
- Most importantly, $\alpha_{mt_k}^{(k)}$ is the weight parameter for the kth factor's component t_k specifically in document m. This allows documents to favor certain components over others, such as the perspective CONSERVATIVE in a specific document.

The $\alpha_m^{(k)}$ parameters can independently adjust the prior probability of components in each factor among all tuples. For example, assigning a high value to $\alpha_{m,\text{BILINGUALISM}}^{(1)}$ for the topic factor (when jointly modeling research topic and discipline) will yield higher prior probability for pairs of the form (BILINGUALISM,*). Thus, the model can independently favor components in each factor that influence all tuples.

One could achieve similar behavior by having K different distributions θ_{mk} and sampling each component of a tuple from these. However, using a single distribution turns out

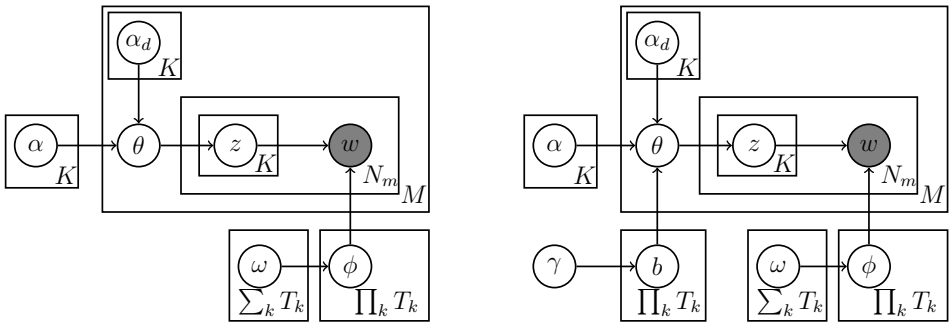


Figure 3.2: The graphical model plate diagrams for (a) Factorial LDA and (b) its sparse variant.

to have desirable properties for inference once we introduce sparsity to the model, as we will explain at the end of Section 3.1.2.

3.1.2 Adding Sparsity

As the dimensionality of the array of tuples increases (that is, the number of factors increases), we are going to encounter problems of overparameterization, because the model will likely contain more tuples than are observed in the data. This is not just a learning difficulty, but also a model interpretability problem. For example, if we are jointly modeling research topic and scientific discipline, it is easy to imagine research topics that do not appear across all research disciplines. If we are modeling papers from the disciplines of linguistics and education, there will be a topic of SEMANTICS that does not appear in education literature and a topic of PUBLIC POLICY that does not appear in linguistics papers. It is therefore problematic for the model to assume the existence of pairs like (SEMAN-

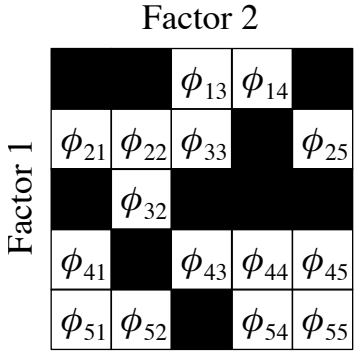


Figure 3.3: A conceptualization of bringing sparsity to factorial LDA. Ideally, only a subset of possible tuples would be associated with word distributions, since otherwise there may be too many parameters to learn and some tuples may not make sense as concepts. In practice, we learn ϕ for all tuples, but assign some of them very low probability.

TICS,EDUCATION), as such word distributions will not be interpretable.

An important variant of factorial LDA therefore allows for *sparsity* in the set of tuples. When originally presented in Paul and Dredze (2012), sparsity was described as part of the general FLDA model. In this thesis, we separately describe the most basic version of FLDA and its sparse variant, and will compare the sparse and dense models.

We can handle sparsity by having an auxiliary multi-dimensional array which encodes a sparsity pattern over tuples. This is illustrated in Figure 3.3. The priors over the document tuple distributions are augmented with this sparsity pattern. These priors model the belief that the Cartesian product of factors should be sparse.

Specifically, we assume a K-dimensional binary array $\boldsymbol{b}$, where an entry $b_{\vec{t}}$ corresponds to tuple $\vec{t}$. If $b_{\vec{t}} = 1$, then $\vec{t}$ is active: that is, we are allowed to chose $\vec{t}$ to generate a token and we learn $\phi_{\vec{t}}$; otherwise we do not. We modify the prior over θ_m to include a binary mask of the tuples: $\theta_m \sim \text{Dirichlet}(\boldsymbol{b} \odot \tilde{\theta}_m)$, where $\odot$ is the Hadamard (cell-wise) product.

If the values of $\boldsymbol{b}$ lie in $\{0, 1\}$, then θ_m will not include tuples for which $b_{\vec{t}} = 0$, otherwise the prior will remain unchanged, and $\boldsymbol{b}$ would therefore function as a selector of the set of tuples in the model. This could be achieved by modeling $\boldsymbol{b}$ as the outcome of a Beta-Bernoulli model (Griffiths and Ghahramani, 2006), for example.

However, it is difficult to learn binary $\boldsymbol{b}$, as changes to the values yield large changes to the model, as encountered by Gormley et al. (2010) when doing inference for shared components topic models (Section 2.4.1.2). To simplify learning, we relax the constraint that $\boldsymbol{b}$ must be binary and instead allow $b_{\vec{t}}$ to be real-valued in $(0, 1)$. This is a common approximation used in other models, such as artificial neural networks and deep belief networks.[1]

To encourage sparsity, we place a "U-shaped" Beta(γ_0, γ_1) prior over $b_{\vec{t}}$, with $\gamma < 1$, which yields a density function that is concentrated around the boundaries 0 and 1. (The Beta distribution is the bivariate version of the Dirichlet distribution, so it has the same properties described in Section 2.1.2.) Empirically, we will show that this effectively learns a sparse binary $\boldsymbol{b}$. The effect is that the prior assigns tiny probabilities to some tuples instead of strictly 0.

For completeness, we describe the full generative story of the sparse FLDA variant,

[1] We arrived at this approximation after experimenting with other approaches. Our original model treated $\boldsymbol{b}$ as a hard constraint and assigned zero probability to variable configurations with any $\vec{z}_{mn} = \vec{t}$ for any $\vec{t}$ such that $b_{\vec{t}} = 0$. This made it non-trivial to infer $\boldsymbol{b}$ values in the Gibbs sampler, because changing a value $b_{\vec{t}} = 1$ to $b_{\vec{t}} = 0$ entailed changing the $\vec{z}_{mn}$ values of any tokens currently assigned to $\vec{t}$. We attempted to do this by running a Gibbs sampler on all such tokens to sample new values, and this new joint configuration of many variables was used as a proposal in the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970). This was based on the *split-merge* algorithm of Jain and Neal (2000), which uses Gibbs sampling to make Metropolis-Hastings proposals that make large changes to cluster configurations. However, we found that the proposals were almost always rejected, and this sampler was far too slow.

with the original steps grayed out:

1. Draw the various hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$ from 0-mean normal distributions
2. For each tuple $\vec{t} = (t_1, t_2, \ldots, t_K)$:
   (a) Let $\tilde{\phi}_{\vec{t}v} = \exp\left\{\omega^{(B)} + \omega_v^{(0)} + \sum_k \omega_{t_k v}^{(k)}\right\}$ for each word v
   (b) Sample word distribution, $\phi_{\vec{t}} \sim \text{Dirichlet}(\tilde{\phi}_{\vec{t}})$
   (c) Sample sparsity "bit" $b_{\vec{t}} \sim \text{Beta}(\gamma_0, \gamma_1)$
3. For each document m:
   (a) Let $\tilde{\theta}_{m\vec{t}} = \exp\left\{\alpha^{(B)} + \left(\sum_k \alpha_{0t_k}^{(k)} + \alpha_{mt_k}^{(k)}\right)\right\}$ for each tuple $\vec{t}$
   (b) Sample distribution over tuples, $\theta_m \sim \text{Dirichlet}(\boldsymbol{b} \odot \tilde{\theta}_m)$
   (c) For each token n in document m:
      i. Sample component tuple $\vec{z}_{mn} \sim \theta_m$
      ii. Sample word $w_{mn} \sim \phi_{\vec{z}_{mn}}$

Figure 3.2(b) shows the graphical model.

As an alternative to modifying the prior over θ_m, one could use $\boldsymbol{b}$ to modify θ_m directly; that is, $\vec{z}$ is sampled proportionally to $b_{\vec{t}}\,\theta_{m\vec{t}}$. However, an advantage of incorporating this in the Dirichlet prior is that this allows us to use the same collapsed Gibbs sampler that is used for LDA (Section 2.5.1.1), since the model is identical to LDA once conditioned on the priors.

Creating a sparsity pattern over tuples is also a motivation for the decision to let θ_m be a single distribution over tuples, rather than having a separate θ_{mk} distribution for each factor. The sparsity parameters $\boldsymbol{b}$ create a dependence between the K components in a tuple. If we had K different parameters θ_{mk}, they could not be analytically collapsed out due to the dependence created by $\boldsymbol{b}$.

3.1.3 Comparison to Related Work

The name factorial LDA is borrowed from factorial hidden Markov models (Ghahramani and Jordan, 1997). While a standard hidden Markov model (HMM) posits a Markov chain of latent variables, a factorial HMM posits multiple Markov chains, and each emission variable depends on its parent variables from each chain. This idea is closely related to FLDA, under which each word token variable depends on multiple latent variables. The parameterization of FLDA using log-linear priors is completely different, however.

Section 2.4.2 described common multi-dimensional models such as the topic aspect model (TAM) (Paul and Girju, 2010b), multi-view LDA (Ahmed and Xing, 2010), and SAGE (Eisenstein et al., 2011). All of these models consider only two dimensions, although a later variant of SAGE modeled three factors in historic documents: topic, time, and location (Wang et al., 2012).

Multi-dimensional topic modeling is also related to factored language models (Bilmes and Kirchhoff, 2003), which are a family of n-gram language models in which multiple factors of word tokens (such as the word class and other features like the word stem) are

jointly modeled. These models do not include latent variables. Some latent variable models have combined multiple factors, in particular models that combine topics and syntax. For example, Griffiths et al. (2005) and Darling et al. (2012) presented models that combine LDA with hidden Markov models, to infer both topic and syntax information.

Other work has applied topic models to multi-dimensional text, though not with a multi-dimensional model. For example, Mei et al. (2007) created a topic-like model of aspect and sentiment. However, each word distribution was associated with either an aspect or a sentiment value, rather than the pairing of both, as in a multi-dimensional model. Zhang et al. (2009) applied a basic topic model (Hofmann, 1999) to multi-dimensional data for online analytical processing (OLAP) data, but not with a joint model. Instead, a one-dimensional topic model was applied to different slices of data.

A non-probabilistic approach considered different dimensions of clustering using spectral clustering (Ng et al., 2001), in which K different clusterings are obtained by considering K different eigenvectors (Dasgupta and Ng, 2010). For example, product reviews can be clustered not only by topic, but also by sentiment and author attributes. However, this approach clusters each dimension independently, whereas FLDA jointly models the interactions between the different dimensions.

3.1.3.1 Relation to product-of-experts models

Perhaps the closest model to FLDA is the topic aspect model (TAM), described in Section 2.4.2.1. In TAM, each token has a latent topic variable and a latent aspect variable, and

each document has its own topic distribution and aspect distribution. The topic and aspect values are independently drawn from their respective distributions in each document. If $\theta_m^{(1)}$ is the distribution over topics and $\theta_m^{(2)}$ is the distribution over aspects, then the joint probability of the ith topic and jth aspect is proportional to $\theta_{mi}^{(1)}\theta_{mj}^{(2)}$. This could be described as a product-of-experts model (Hinton, 2002), in which different distributions are multiplied together and renormalized to form a joint distribution.[2] (In this case, renormalization is not necessary, but we are using this terminology because it will be reused later in this subsection.)

Factorial LDA, in contrast, has a fully parameterized joint distribution, rather than giving each factor its own distribution. However, the Dirichlet prior over this joint distribution is in fact similar to a product-of-experts, since the log-linear function is implicitly a product of its exponentiated parameters, and the prior contains parameters for each factor independently. Therefore, the prior for each document's distribution over tuples is centered around a product of factor-specific weights for the document, and if the Dirichlet precision is very high, then the posterior will be concentrated around this product, resulting in similar behavior to TAM.

For the word distributions, however, TAM provides no structure at all to tie together word distributions for pairs that share components, and all ϕ were assumed conditionally independent. This would result in word distributions that were not always sensible to the user. An important contribution of FLDA is the use of priors to tie together word distribu-

[2] A product-of-experts model is similarly used to define the topic distributions in syntactic topic models (Boyd-Graber and Blei, 2008). Here, the topic distribution is proportional to the product of a document-specific topic preference as well as a syntactic topic preference.

tions with the same components. As with the priors over topic distributions, the log-linear prior over each topic's word distribution is a product of weights from each factor independently.

A product-of-experts model was used to define word distributions in shared components topic models (SCTM) (Gormley et al., 2010), in which each topic's word distribution is defined as a normalized product of distributions from a set of underlying components that are combined to form topics. Such an approach could be used for FLDA, but then a tuple's word distribution would not contain information specific to the combination of components. By learning parameters that are specific to the tuple, but using a prior that is a product of weights from each factor independently, FLDA achieves a "happy medium" between the various possibilities. If the Dirichlet precision is very high, then the word distributions will be more like SCTM (where the word distributions are deterministically defined by the product of components), and if the precision is low and close to uniform, then the word distributions will be more like TAM (where all word distributions are independent). By learning ω, our model can determine the optimal amount of cohesion among the ϕ.

3.1.3.2 Relation to structured sparsity

Section 2.4.3 described different types of topic model sparsity and how FLDA's sparsity fits in. We additionally note that among sparsity-inducing regularizers, one that closely relates to FLDA's sparsity is the *group lasso* (Meier et al., 2008). While the standard lasso (which applies ℓ_1 regularization to linear regression) will drive vector elements to 0, the

group lasso will drive entire vectors to 0. The group lasso is an example of *structured sparsity* (Huang et al., 2009; Martins et al., 2011). While the group lasso cannot be directly applied to the parameterization used by FLDA, the concept is related because in FLDA we want to make an entire set of parameters (a word distribution) sparse.

3.2 Inference and Optimization

To learn the parameters of FLDA, we use a Monte Carlo EM routine, as outlined in Section 2.5.2. Conditioned on the structured priors, we use a collapsed Gibbs sampler to sample from the posterior of the tuple assignments, from which the posterior of the model parameters—each document's distribution over tuples and each tuple's distribution over words—can be computed. Conditioned on the sampler's estimate of $\mathbb{E}[\vec{\mathbf{z}}]$, we optimize the Dirichlet parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$, as well as the sparsity pattern $\boldsymbol{b}$ for the sparse variant.

3.2.1 Latent Variable Sampling

The latent variables $\vec{\mathbf{z}}$ are sampled using the standard collapsed Gibbs sampler for LDA (Section 2.5.1.1), with the exception that the basic Dirichlet priors have been replaced with our structured priors for θ and ϕ. The sampling distribution is the same as LDA's if one treats each K-tuple as an individual "topic", with $\prod_k T_k$ total topics. This "flat" representation will be used when FLDA is generalized by SPRITE in Chapter 5, but here we make the tuple structure explicit.

Thus, the sampling equation for $\vec{z}_{mn}$ for token w_{mn}, given all other latent variable assignments $\vec{\boldsymbol{z}}$, the corpus $\boldsymbol{w}$ and the parameters ($\boldsymbol{\alpha}$, $\boldsymbol{\omega}$, and $\boldsymbol{b}$) becomes:

$$p(\vec{z}_{mn} = \vec{t} \mid \vec{\boldsymbol{z}} - \vec{z}_{mn}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\omega}, \boldsymbol{b}) \propto \left(n_{\vec{t}}^{m} + b_{\vec{t}}\, \tilde{\theta}_{m\vec{t}}\right)\left(\frac{n_{w_{mn}}^{\vec{t}} + \tilde{\phi}_{\vec{t}w_{mn}}}{\sum_v n_v^{\vec{t}} + \tilde{\phi}_{\vec{t}v}}\right) \tag{3.1}$$

where n_x^y denotes the number of times x occurs in y, excluding the count of the current token. If the sparsity pattern is not included, then let $b_{\vec{t}} = 1$.

3.2.2 Optimizing the Structured Priors

After each Gibbs sampling iteration, our algorithm takes one iteration of gradient ascent for the parameters of the structured priors. This subsection defines the gradient with the partial derivatives of the corpus likelihood with respect to $\boldsymbol{\alpha}$, $\boldsymbol{\omega}$, and $\boldsymbol{b}$. The gradient has a similar form to the gradient with respect to the LDA hyperparameters in Section 2.5.2.

3.2.2.1 Dirichlet Parameters

The partial derivatives of the corpus log likelihood $\mathcal{L}$ with respect to the $\boldsymbol{\alpha}$ parameters are:

$$\frac{\partial \mathcal{L}}{\partial \alpha_{mt_k}^{(k)}} = -\frac{\alpha_{mt_k}^{(k)}}{\sigma^2} + \sum_{\vec{t} \in \mathcal{T}(k,t_k)} \tilde{\theta}_{m\vec{t}} \Big(\Psi(n_{\vec{t}}^m + \tilde{\theta}_{m\vec{t}}) - \Psi(\tilde{\theta}_{m\vec{t}}) + \Psi\left(\textstyle\sum_{\vec{t}'} \tilde{\theta}_{m\vec{t}'}\right) - \Psi\left(\textstyle\sum_{\vec{t}'} n_{\vec{t}'}^m + \tilde{\theta}_{m\vec{t}'}\right) \Big) \tag{3.2}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_{0t_k}^{(k)}} = -\frac{\alpha_{0t_k}^{(k)}}{\sigma^2} + \sum_{m=1}^{M} \sum_{\vec{t} \in \mathcal{T}(k,t_k)} \tilde{\theta}_{m\vec{t}} \Big(\Psi(n_{\vec{t}}^m + \tilde{\theta}_{m\vec{t}}) - \Psi(\tilde{\theta}_{m\vec{t}}) + \Psi\left(\textstyle\sum_{\vec{t}'} \tilde{\theta}_{m\vec{t}'}\right) - \Psi\left(\textstyle\sum_{\vec{t}'} n_{\vec{t}'}^m + \tilde{\theta}_{m\vec{t}'}\right) \Big) \tag{3.3}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha^{(B)}} = -\frac{\alpha^{(B)}}{\sigma^2} + \sum_{m=1}^{M} \sum_{\vec{t} \in \mathcal{T}(*)} \tilde{\theta}_{m\vec{t}} \Big(\Psi(n_{\vec{t}}^m + \tilde{\theta}_{m\vec{t}}) - \Psi(\tilde{\theta}_{m\vec{t}}) + \Psi\left(\textstyle\sum_{\vec{t}'} \tilde{\theta}_{m\vec{t}'}\right) - \Psi\left(\textstyle\sum_{\vec{t}'} n_{\vec{t}'}^m + \tilde{\theta}_{m\vec{t}'}\right) \Big) \tag{3.4}$$

where we let $\mathcal{T}(k, t_k) = \{\vec{z} : z_k = t_k, 1 \leq z_j \leq T_j \; \forall j \neq k\}$ denote the set of tuples such that the kth component has value t_k, and $\mathcal{T}(*) = \{\vec{z} : 1 \leq z_k \leq T_k \; \forall k\}$ is the set of all valid tuples.

The σ^2 in the first term of each derivative is the variance of the Gaussian prior of the hyperparameters. These derivatives do not include sparsity. For the sparse variant of FLDA, each $\tilde{\theta}_{m\vec{t}}$ should be multiplied by $b_{\vec{t}}$.

Next, the partial derivatives of the corpus log likelihood with respect to the $\boldsymbol{\omega}$ parame-

ters are:

$$\frac{\partial \mathcal{L}}{\partial \omega_{t_k v}^{(k)}} = -\frac{\omega_{t_k v}^{(k)}}{\sigma^2} + \sum_{\vec{t} \in \mathcal{T}(k,t_k)} \tilde{\phi}_{\vec{t}v} \left(\Psi(n_v^{\vec{t}} + \tilde{\phi}_{\vec{t}v}) - \Psi(\tilde{\phi}_{\vec{t}v}) + \Psi\left(\sum_{v'} \tilde{\phi}_{\vec{t}v'}\right) - \Psi\left(\sum_{v'} n_{v'}^{\vec{t}} + \tilde{\phi}_{\vec{t}v'}\right)\right) \tag{3.5}$$

$$\frac{\partial \mathcal{L}}{\partial \omega_{v}^{(0)}} = -\frac{\omega_{v}^{(0)}}{\sigma^2} + \sum_{\vec{t} \in \mathcal{T}(*)} \tilde{\phi}_{\vec{t}v} \left(\Psi(n_v^{\vec{t}} + \tilde{\phi}_{\vec{t}v}) - \Psi(\tilde{\phi}_{\vec{t}v}) + \Psi\left(\sum_{v'} \tilde{\phi}_{\vec{t}v'}\right) - \Psi\left(\sum_{v'} n_{v'}^{\vec{t}} + \tilde{\phi}_{\vec{t}v'}\right)\right) \tag{3.6}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \omega^{(B)}} = &-\frac{\omega^{(B)}}{\sigma^2} + \\ &\sum_{v=1}^{V} \sum_{\vec{t} \in \mathcal{T}(*)} \tilde{\phi}_{\vec{t}v} \left(\Psi(n_v^{\vec{t}} + \tilde{\phi}_{\vec{t}v}) - \Psi(\tilde{\phi}_{\vec{t}v}) + \Psi\left(\sum_{v'} \tilde{\phi}_{\vec{t}v'}\right) - \Psi\left(\sum_{v'} n_{v'}^{\vec{t}} + \tilde{\phi}_{\vec{t}v'}\right)\right) \end{aligned} \tag{3.7}$$

3.2.2.2 Sparsity Pattern

Recall that each value of $\boldsymbol{b}$ lies in $(0, 1)$. For mathematical convenience, we reparameterize $\boldsymbol{b}$ in terms of the logistic function σ, such that $b_{\vec{t}} = \sigma(\hat{b}_{\vec{t}})$, where $\sigma(x) = \frac{1}{1+\exp(-x)}$. This allows us to optimize $\hat{b} \in \mathbb{R}$ to obtain $b \in (0, 1)$, keeping this as an unconstrained optimization problem.

The derivative of $\sigma(x)$ has the form $\sigma(x)\sigma(-x)$. For each tuple $\vec{t}$, the partial derivative

of $\mathcal{L}$ with respect to $\hat{b}_{\vec{t}}$ is:

$$\frac{\partial \mathcal{L}}{\partial \hat{b}_{\vec{t}}} = (\gamma_0 - 1)\sigma(-\hat{b}_{\vec{t}}) + (\gamma_1 - 1)(-\sigma(\hat{b}_{\vec{t}})) + \left[\sum_{m=1}^{M} \sigma(\hat{b}_{\vec{t}})\sigma(-\hat{b}_{\vec{t}})\ \tilde{\theta}_{m\vec{t}} \times \right. \tag{3.8}$$
$$\left. \left(\Psi(n^m_{\vec{t}} + \sigma(\hat{b}_{\vec{t}})\tilde{\theta}_{m\vec{t}}) - \Psi(\sigma(\hat{b}_{\vec{t}})\tilde{\theta}_{m\vec{t}}) + \Psi\left(\textstyle\sum_{\vec{u}} \sigma(\hat{b}_{\vec{u}})\right) \tilde{\theta}_{m\vec{u}}) - \Psi\left(\textstyle\sum_{\vec{u}} n^m_{\vec{u}} + \sigma(\hat{b}_{\vec{u}})\right) \tilde{\theta}_{m\vec{u}}) \right) \right]$$

where the γ values are the Beta parameters. The first two terms are a result of the Beta(γ_0, γ_1) prior over $b_{\vec{t}}$.

3.3 Experiments

This section describes experiments performed to conduct a general evaluation of FLDA's potential utility, and to understand the behavior of the model under different settings. Chapter 4 will investigate the utility of FLDA for specific applications in health science.

3.3.1 Experimental Details

We experiment with two datasets of research articles. The first is a collection of 5000 computational linguistics abstracts from the ACL Anthology, denoted 'ACL'. The second combines these abstracts (C) with several journals in the fields of linguistics (L), education (E), and psychology (P). We use 1000 articles from each discipline (4000 documents total), denoted 'CLEP'. For both corpora, we keep an additional 1000 documents for development and 1000 for test (uniformly representative of the 4 CLEP disciplines, with 250 documents

each). Both datasets were created by the author (Paul, 2009). Stop words are removed from the data.[3]

We used $\vec{T} = (*, 2, 2)$ for ACL (three factors) and $\vec{T} = (*, 4)$ for CLEP (two factors) for various numbers of "topics" $T_1 \in \{5, \ldots, 50\}$. While we cannot say *a priori* what each factor will represent, we observed that when T_1 is large, components along this factor can be interpreted as topics. Therefore, we set $T_1 > T_{k>1}$ and refer to this factor as topic. For the evaluation here, we focus on the unsupervised setting, although indirect forms of supervision are considered in Chapter 4.

We will compare the sparse FLDA model against simpler models by ablating parts of the model. If we remove the structured word priors and sparsity, we are left with a basic multi-dimensional model, referred to as the 'base' model. We will compare against models where we add the structured word priors ('W') and array sparsity ('S') in isolation, and finally the complete FLDA model with sparsity (referred to as 'SW'). All variants are identical except that we fix all $\boldsymbol{\omega}^{(\boldsymbol{k})} = \mathbf{0}$ to remove structured word priors and fix $\boldsymbol{b} = \mathbf{1}$ to remove sparsity.

We also compare against the topic aspect model (TAM) (Paul and Girju, 2010b), a two-dimensional model, using the public implementation. TAM is similar to the 'base' two-factor FLDA model except that FLDA has a single θ_m per document with priors that are independently weighted by each factor, whereas TAM has K independent θs, with a different θ_{mk} for each factor. If the Dirichlet precision in FLDA is very high, then it will

[3]The stop word list, used for all experiments in this thesis, is available at:
`http://cs.jhu.edu/~mpaul/data/stop_words.txt`

exhibit similar behavior as having separate θs. TAM only models two dimensions, so we only run this on the two-dimensional CLEP dataset.

For hyperparameters, we set $\gamma_0 = \gamma_1 = 0.1$ in the Beta prior over $b_{\vec{t}}$, and we set $\sigma^2 = 10$ for $\boldsymbol{\alpha}$ and $\sigma^2 = 1$ for $\boldsymbol{\omega}$ in the normal prior over weights. Bias parameters ($\alpha^{(B)}$, $\omega^{(B)}$) are initialized to -5 for weak initial priors. Our sampling algorithm alternates between a full pass over tokens and a single gradient step on the parameters using a constant step size of 10^{-2} for $\boldsymbol{\alpha}$; 10^{-3} for $\boldsymbol{\omega}$ and $\boldsymbol{b}$). Results were averaged from five trials of randomly initialized Gibbs samplers, which were each run for 10,000 iterations.

3.3.2 Perplexity

To evaluate the predictiveness of the models, we measure perplexity on held-out data. We fix all parameters learned from the training data, and we need to learn document-specific parameters on the new data. Specifically, we optimize each $\alpha^{(m,k)}$ on the test documents to infer each θ_m. We use the "document completion" method where we infer θ_m based on half of a document, and we measure the perplexity of the remaining half given our inferred θ_m (Section 2.6.1). We run our MCEM algorithm on the test data for 200 iterations to infer the document parameters, and measure the average perplexity over an additional 10 iterations.

Figure 3.4 shows that the structured word priors yielded lower perplexity, while results for sparse models were mixed. On ACL, sparsity consistently improved perplexity once the number of topics exceeds 20, while on CLEP sparsity worsened perplexity. On CLEP,

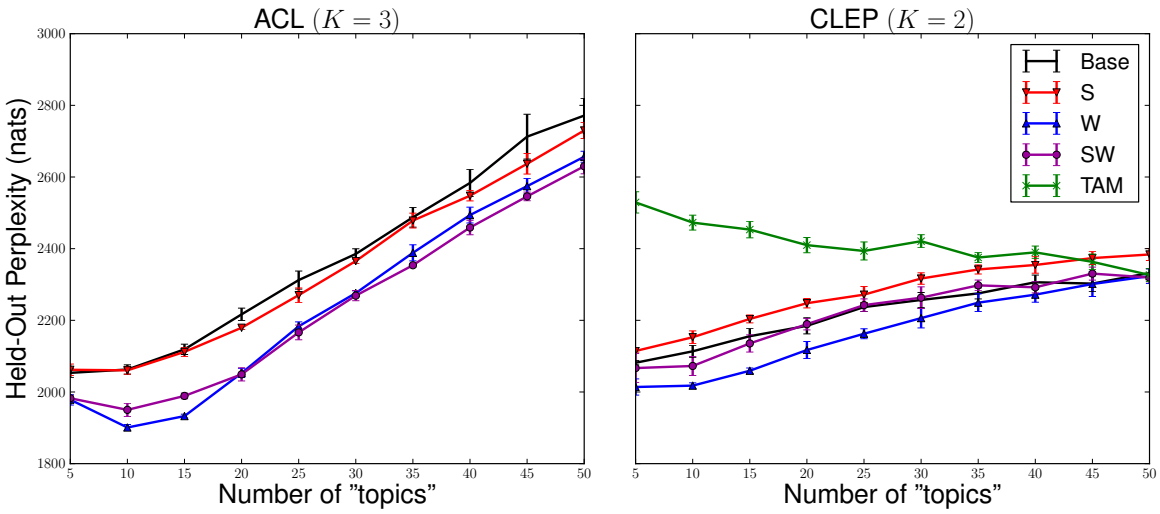


Figure 3.4: The document completion perplexity on two datasets. Lower is better. Models with "W" use structured word priors, and those with "S" use sparsity. Error bars indicate 90% confidence intervals. When pooling results across all numbers of topics $\geq$20, we find that S is significantly better than Base with $p = 1.4 \times 10^{-4}$ and SW is better than W with $p = 5 \times 10^{-5}$ on the ACL corpus.

we found that TAM performs worse than FLDA with a lower number of topics (which is what we found to work best qualitatively, based on our own observations), but catches up as the number of topics increases.

3.3.3 Human Judgments

Perplexity may not correlate with human judgments (Chang et al., 2009), which are important for FLDA since structured word priors and array sparsity are motivated in part by semantic coherence. Therefore, we conducted human evaluations to understand what latent factors represent, how factors interact, and how they are tied. We measured interpretability based on the notion of relatedness: among components that are inferred to belong to the

same factor, how many actually make sense together? Seven annotators provided judgments for two related tasks. First, we presented annotators with two word lists—the ten most frequent words assigned to each tuple—that that are assigned to the same topic (the first component), along with a word list randomly selected from a tuple with a different topic. Annotators are asked to choose the word list that does not belong, i.e., an intrusion test (Chang et al., 2009). If the two tuples from the same topic are strongly related, the random list should be easy to identify. In some cases, the intruder list can be determined based on other clues even when the other lists are not related. For example, if two "junk" lists are paired with a "good" list, it is clear the third one is out of place.

Second, annotators are presented with pairs of word lists from the same topic and asked to judge the degree of relation using a 5-point Likert scale. Since our modeling assumptions are that different tuples that share components should have some commonalities, we want to measure the degree to which this is true. While more subjective, this explicitly measures relatedness.

We ran these experiments on both corpora with 20 topics. For the two models without the structured word priors, we use a symmetric prior (by optimizing only $\omega^{(B)}$ and fixing $\omega^{(0)} = \mathbf{0}$), since symmetric word priors can lead to better interpretability (Wallach et al., 2009a).[4] In the sparse variants, we excluded tuples with $b_{\vec{t}} \leq 0.5$.

Across all datasets and models, annotators labeled 362 triples in the intrusion experiment and 333 pairs in the scoring experiment. The results (Table 3.1) differ slightly from

[4]We used an asymmetric prior for the perplexity experiments, which gave slightly better results.

	ACL	CLEP	ACL	CLEP
	Intrusion Accuracy		Relatedness Score (1–5)	
TAM	n/a	46%	n/a	2.29 ± 0.26
Baseline	39%	38%	2.35 ± 0.31	2.55 ± 0.37
Sparsity (S)	51%	43%	2.61 ± 0.37	2.53 ± 0.48
Word Priors (W)	**76%**	45%	3.56 ± 0.36	2.59 ± 0.33
Combined (SW)	73%	**67%**	**3.90** ± 0.37	**2.67** ± 0.55

Table 3.1: Results from human judgments. The best scoring model for each dataset is in bold. 90% confidence intervals are indicated for scores; scores were more varied on the CLEP corpus.

the perplexity results. The word priors help in all cases, but much more so on ACL. The models with sparsity are generally better than those without, even on CLEP, in contrast to perplexity where sparse models did worse. This suggests that removing tuples with small $b_{\vec{t}}$ values removes nonsensical tuples.

Overall, the judgments are worse for the CLEP corpus; this appears to be a difficult corpus to model due to high topic diversity and low overlap across disciplines. TAM is judged to be worse than all FLDA variants when directly scored by annotators. The intrusion performance with TAM is better than or comparable to the ablated versions of FLDA, but worse than the full model. It thus appears that both the structured priors and sparsity yield more interpretable word clusters.

3.3.4 Analysis of Sparsity Patterns

We now examine the learned sparsity patterns: how much of $\boldsymbol{b}$ is close to 0 or 1? Figure 3.5 shows a histogram of $b_{\vec{t}}$ values (ACL with 20 topics, 3 factors) pooled across

five sampling chains. The majority of values are close to 0 or 1, effectively capturing a sparse binary array. The higher variance near 0 relative to 1 suggests that the model prefers to keep bits "on"—and give tuples tiny probability—rather than "off." This suggests that a model with a hard constraint might struggle to "turn off" bits during inference.

While we fixed the Beta parameters in our experiments, these can be tuned to control sparsity. The model will favor more "on" than "off" bits by setting $\gamma_1 > \gamma_0$, or vice versa. When $\gamma > 1$, the Beta distribution no longer favors sparsity; we confirmed empirically that this leads to $b_{\vec{t}}$ values that are closer to 0.8 or 0.9 rather than 1. In contrast, setting $\gamma \ll 0.1$ yields more extreme values near 0 and 1 than with $\gamma = 0.1$ (e.g., .9999 instead of .991), but this does not greatly affect the number of non-binary values. Thus, a sparse prior alone cannot fully satisfy our preference that $\boldsymbol{b}$ is binary.[5]

3.3.5 Empirical Comparison to LDA

Because the FLDA Gibbs sampler is so similar to the LDA sampler—the only difference is the step of hyperparameter optimization— the runtimes of samplers for LDA and FLDA are on the same order, for a comparable number of word distributions. For example, $\vec{T} = (20, 2, 2)$ is the same as $T = 80$ topics in LDA. For a comparable number of word distributions, our FLDA implementation is between one and two times slower per iteration than our own comparable LDA implementation, with hyperparameter optimization using the methods in Paul (2012). However, the number of word distributions increases exponen-

[5] However, in Section 5.3.2.1 of Chapter 5 we show that such priors can be strengthened to satisfy these constraints.

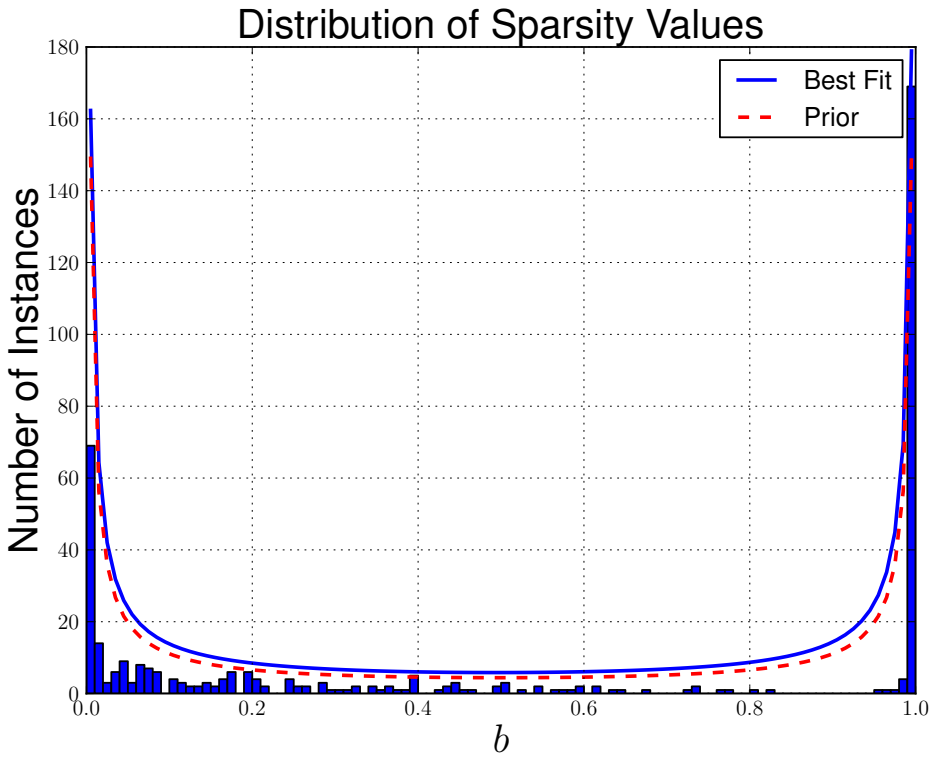


Figure 3.5: The distribution of sparsity values learned on the ACL corpus with $\vec{T} = (20, 2, 2)$. The dashed curve shows the Beta prior that was used for these values, while the solid curve shows the best-fitting Beta distribution to these values.

tially in the number of factors in FLDA, so in practice this can make it slower than LDA, since more word distributions might be needed than in LDA.

We did not observe a consistent pattern regarding the perplexity of the two models. Averaged across all numbers of topics, the perplexity of LDA was 97% the perplexity of FLDA on ACL and 104% on CLEP, using a comparable number of word distributions for LDA and FLDA.

"Topic"				"Approach"		"Focus"	
"SPEECH"	"I.R."	"M.T."	...	"EMPIRICAL"	"THEORETICAL"	"METHODS"	"APPLICATIONS"
speech	document	translation	...	task	theory	word	user
spoken	retrieval	machine	...	tasks	description	algorithm	research
recognition	documents	source	...	performance	formal	method	project
state	question	mt	...	improve	forms	accuracy	technology
vocabulary	web	parallel	...	accuracy	treatment	best	processing
recognizer	answering	french	...	learning	linguistics	sentence	science
utterances	query	bilingual	...	demonstrate	syntax	statistical	natural
synthesis	answer	transfer	...	using	ed	previously	development

Topic		SPEECH		DATA		MODELING		GRAMMAR	
Focus		METHODS	APPL.	METHODS	APPL.	METHODS	APPL.	METHODS	APPL.
Approach	EMPIRICAL	**(b=0.20)**	**(b=1.00)** dialogue spoken speech dialogues understanding task recognition	**(b=1.00)** corpus data training model tagging annotated test	**(b=1.00)** data corpus annotation annotated corpora collection xml	**(b=1.00)** models model approach shown error errors statistical	**(b=0.50)**	**(b=1.00)** parsing parser syntactic tree parse dependency treebank	**(b=0.57)** grammar parsing based robust component processing linguistic
	THEORETICAL	**(b=0.99)** speech words recognition prosodic written phonological spoken	**(b=0.00)**	**(b=0.07)**	**(b=0.02)**	**(b=1.00)** rules rule model shown models right left	**(b=0.01)**	**(b=1.00)** grammar parsing grammars structures paper formalism based	**(b=1.00)** grammar grammars formalism parsing based efficient unification

Figure 3.6: Example FLDA output from the ACL corpus with $\vec{T} = (20, 2, 2)$. Above: The top words (based on their ω values) for a few components from three factors. Below: A three-dimensional table showing a sample of four topics (i.e., components of the first factor) with their top words (based on their ϕ values) as they appear in all combinations of factors. The components in the top table are combined to create 3-tuples in the bottom table. Shaded cells ($b \leq 0.5$) are inactive. The names of factors and their components in quotes are manually assigned through post-hoc analysis.

3.3.6 Qualitative Analysis

To illustrate model behavior, we include a sample of output on the ACL corpus in Figure 3.6, selecting topics that represent a diverse range of sparsity patterns. We consider the component-specific weights for each factor $\omega_{t_k}^{(k)}$, which present an "overview" of each component, as well as the tuple-specific word distributions $\phi_{\vec{t}}$.

Upon examination, we determined that the first factor (T_1= 20) corresponds to *topic*, the second (T_2= 2) to *approach* (EMPIRICAL or THEORETICAL), and the third (T_3= 2) to

focus (METHODS or APPLICATIONS). The top row shows words common across all components for each factor. The bottom row shows specific $\phi_{\vec{t}}$. Consider the topic SPEECH: the triple (SPEECH,METHODS,THEORETICAL) emphasizes the linguistic side of speech processing (*phonological*, *prosodic*, etc.) while (SPEECH,APPLICATIONS,EMPIRICAL) is predominantly about dialogue systems and speech interfaces. Both of these triples include *speech* and *spoken* among the highest probability words, which receive high prior probability because these have the highest ω values for the speech topic.

We also see tuple sparsity (shaded tuples, in which $b_{\vec{t}} \leq 0.5$) for unsupported tuples. For example, under the topic of DATA, a mostly empirical topic, tuples along the THEORETICAL component are inactive.

In addition to the top words, we also examined the documents associated with various tuples. Figure 3.7 shows the three documents with the highest tuple proportion $\theta_{m\vec{t}}$ for various example tuples, showing the titles of the papers from the ACL corpus. This figure was generated using a different Gibbs sampling run than Figure 3.6, so some of the learned tuples differ, but the same general concepts were learned by components. For this figure, we showed documents for all tuples regardless of the sparsity $\boldsymbol{b}$ values.

As with the top words, the top documents generally show interpretable patterns. For example, the papers under (SPEECH,APPLICATIONS,*) are mostly about dialog, with papers describing dialog systems under the EMPIRICAL approach, and papers with dialog theory (e.g., social goals and conversation misunderstanding) under the THEORETICAL approach.

Linguistics-oriented papers generally favor the THEORETICAL component. For exam-

SPEECH		
	METHODS	APPLICATIONS
EMPIRICAL	• The Lincoln Continuous Speech Recognition System: Recent Developments and Results • Minimizing Speaker Variation Effects for Speaker-Independent Speech Recognition • Improved Acoustic Modeling for Continuous Speech Recognition	• University of Colorado Dialog Systems for Travel and Navigation • The Collection and Preliminary Analysis of a Spontaneous Speech Database • Evaluation of the CMU ATIS System
THEORETICAL	• Incremental generation of spatial referring expressions in situated dialog • A context-dependent algorithm for generating locative expressions in physically situated env. • Referring Expression Generation Using Speaker-based Attribute Selection and Trainable Realization	• Abductive explanation of dialogue misunderstandings • Social Goals in Conversational Cooperation • Using Linguistic, World, and Contextual Knowledge in a Plan Recognition Model of Dialogue
GRAMMAR		
	METHODS	APPLICATIONS
EMPIRICAL	• Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based LFG approximations • Dependency-Based N-Gram Models for General Purpose Sentence Realisation • Extracting Syntactic Features from a Korean Treebank	• PRINCIPAR–An Efficient, Broad-coverage, Principle-based Parser • EXEMPLARS: A Practical, Extensible Framework For Dynamic Text Generation • Thistle and Interarbora
THEORETICAL	• XMG: an expressive formalism for describing tree-based grammars • Generative Power of CCGs with Generalized Type-Raised Categories • Parsing Strategies with 'Lexicalized' Grammars: Application to Tree Adjoining Grammars	• Categorial Grammars for Strata of Non-CF Languages and Their Parsers • Semantic Caseframe Parsing and Syntactic Generality • A View of Parsing
DOCUMENT STRUCTURE		
	METHODS	APPLICATIONS
EMPIRICAL	• Support Vector Machines for Query-focused Summarization trained and evaluated on Pyramid data • Integrating Cross-Lingually Relevant News Articles and Monolingual Web Documents in Bilingual Lexicon Acquisition • A Text Categorization Based on Summarization Technique	• Pedagogically Useful Extractive Summaries for Science Education • A Machine Learning Approach to Extract Temporal Information from Texts in Swedish and Generate Animated 3D Scenes • Using Linguistic Knowledge in Automatic Abstracting
THEORETICAL	• Discourse Relations and Discourse Markers • Discursive Usage of Six Chinese Punctuation Marks • Text Segmentation with Multiple Surface Linguistic Cues	• Towards Generating Procedural Texts: an exploration of their rhetorical and argumentative structure • Quantitative Portraits of Lexical Elements • Reassessing Rhetorical Abstractions and Planning Mechanisms

Figure 3.7: Example document titles representative of various 3-tuples learned by FLDA on the ACL corpus with $\vec{T} = (20, 2, 2)$. These titles are from the documents with the highest tuple proportion $\theta_{m\vec{t}}$ for the indicated tuple.

ple, the paper titled "Thistle and Interarbora" presents an application of manipulating linguistic diagrams, so it naturally fits the triple (GRAMMAR,APPLICATIONS,THEORETICAL). Under the DOCUMENT STRUCTURE topic, many of the THEORETICAL papers are about discourse structure, while many EMPIRICAL papers are about document summarization.

3.4 Summary

This chapter presented factorial LDA (FLDA), a multi-dimensional text model that can incorporate an arbitrary number of factors. To encourage the model to learn the desired patterns, we developed two new types of structure for the model priors: word priors that share features across factors, and a sparsity variable that restricts the set of active tuples. We have shown both qualitatively and quantitatively that FLDA is capable of discovering interpretable patterns in multi-dimensional spaces, and we find that our new priors improve model predictiveness and coherence.

Chapter 4

Applications of Factorial LDA

This chapter shows how factorial LDA (FLDA) can be applied to specific problems in the domain of health science. We demonstrate the utility of FLDA for two applications involving text from the Web: learning information about drugs (e.g., dosage and side effects) from informal user reports (Section 4.1) and learning about perceived quality of healthcare from patient reviews (Section 4.2).

This chapter provides experimental evaluation of the models for application-specific tasks. For the drug application, we show that FLDA can extract text describing various aspects of drug use that closely matches literature written by domain experts. For the healthcare application, we show that features learned by FLDA are significantly correlated with external measures of healthcare quality across U.S. states. These experiments suggest that FLDA is learning meaningful information that can be of value to researchers in health domains.

While the contributions of this chapter are more applied, a technical challenge that arises when using FLDA for specific applications is the need to learn specific concepts that are relevant to the application. Since topic models are typically unsupervised, the inferred distributions may not correspond to the concepts needed for the application. This chapter will introduce a novel hierarchical prior over the model parameters informed by domain knowledge, guiding the models toward the desired concepts (Sections 4.1.3 and 4.2.3).

This chapter is broken into two main sections, one for each application, followed by a concluding section. Within each application section, we follow the same outline. We first describe the task and provide motivating background information about the scientific problem. We then describe the FLDA model used for the task—what the factors and components correspond to—and then we show how to extend the priors to incorporate domain knowledge specific to each application. Lastly, we show experimental results including standard topic model evaluation—predictive experiments and example topic output—as well as domain-specific evaluation for each application.

4.1 Drug Information Summarization from Web Forums

Our first task is to mine information about illicit drugs from user forums, an important clinical research problem as explained in Section 4.1.1. This section is based on material from Paul and Dredze (2013) and Paul and Dredze (2012b).

We use a three-dimensional factorial LDA model that jointly models different types of drug-related information: the type of drug, the route of administration, and the aspect of drug use. To learn these concepts, we create priors over the word distributions using labeled data (Section 4.1.3). This is done by training a simplified supervised model on a subset of the corpus and using the learned parameters as normal distribution priors over the FLDA parameters.

We then demonstrate the model's utility in exploring a corpus in a targeted manner by using it to automatically extract interesting excerpts from the text, a simple form of extractive multi-document summarization (Goldstein et al., 2000), in which representative portions of text are extracted from large corpora. In the same way that topic models can be used for aspect-specific summarization (Titov and McDonald, 2008; Haghighi and Vanderwende, 2009), we use FLDA to extract snippets corresponding to fine-grained information patterns. Our results in Section 4.1.4.3 demonstrate that the multi-dimensional modeling approach targets more informative text than a simpler baseline.

4.1.1 Task and Motivation

Ilicit drug use imposes a significant burden on the health infrastructure of the United States and other countries. Accurate information on drugs, usage profiles and side effects are necessary for supporting a range of healthcare activities, such as addiction treatment programs, toxin diagnosis, prevention and awareness campaigns, and public policy. These activities rely on up-to-date information on drug trends as substance popularity changes in

response to legislative efforts and market trends. For example, hospitals and poison control centers among others must remain informed on the pharmacological and toxicological effects of new and popular drugs (Hill and Thomas, 2011). Understanding usage patterns can inform outreach strategies (Bruneau et al., 2012).

It is increasingly difficult to keep up with current drug information, as distribution and information-sharing of novel drugs is easier than ever via the Web (Wax, 2002). For the third consecutive year, a record number of new drugs were detected in Europe in 2011 (EMCDDA, 2012). About two-thirds of these new drugs were synthetic cannabinoids (used as legal marijuana substitutes), which led to 11,000 hospitalizations in the U.S. in 2010 (SAMHSA, 2012). Treatment is complicated by the fact that novel substances like these may have unknown side effects and other properties.

Accurate information on drug trends can be obtained by speaking directly with users, e.g., focus groups and interviews (Reyes et al., 2012; Hout and Bingham, 2012), but such studies are slow and costly, and can fail to identify the emergence of new drug classes, such as mephedrone (Dunn et al., 2011).

More recently, researchers have begun to recognize clinical value in information obtained from the Web (Corazza et al., 2011). By (manually) analyzing YouTube videos, Drugs-Forum (discussed below), and other social media websites and online communities, researchers have uncovered details about the use, effects, and popularity of a variety of new and emerging drugs (Morgan et al., 2010; Corazza et al., 2012; Gallagher et al., 2012), and comprehensive drug reviews now include non-standard sources such as Web forums in

addition to standard sources (Hill and Thomas, 2011). The EU Psychonaut Project has created a large database of recreational drug information from online forums (Schifano et al., 2006).

However, while online forums contain a wealth of information, organizing and understanding forums requires significant effort. We propose automated tools to aid in the exploration and analysis of this data.

In this section, we will show how factorial LDA can be used to learn information about drugs from online forums. Topic models have been used for targeted browsing of corpora (Eisenstein et al., 2012; Chaney and Blei, 2012) as well as extractive summarization of text (Titov and McDonald, 2008; Haghighi and Vanderwende, 2009), and we will consider FLDA for both of these purposes. Section 4.1.4.2 will show examples of information learned from this dataset using FLDA, and Section 4.1.4.3 will show how informative text snippets can be extracted using this model.

4.1.1.1 Dataset

Our data set is taken from Drugs-Forum (`drugs-forum.com`), a website that has been active for more than 10 years with over 100,000 members and more than 1 million monthly readers. The site is an information hub where people can freely discuss recreational drugs with psychoactive effects, ranging from coffee to heroin, hosting information and discussions on specific drugs, as well as drug-related politics, law, news, recovery and addiction. Site users are primarily drug users, but also include researchers, parents,

Factor	Components
Drug	ALCOHOL AMPHETAMINES BETA-KETONES CANNABINOIDS CANNABIS COCAINE DMT DOWNERS DXM ECSTASY GHB HERBAL ECSTASY KETAMINE KRATOM LSA LSD NOOTROPICS OPIATES PEYOTE PHENETHYLAMINES SALVIA TOBACCO
Route	INJECTION ORAL SMOKING SNORTING
Aspect	CHEMISTRY (Pharmacology, TEK) CULTURE (Culture, Setting, Social, Spiritual) EFFECTS (Effects) HEALTH (Health, Overdose, Side effects) USAGE (Dose, Storing, Weight)

Table 4.1: The three factors of drug information that we model. The components of each factor are shown in smallcaps. The forum tags shown in parentheses are manually grouped together to form aspects.

officials, NGOs, lawyers, doctors, journalists and addiction specialists. With current information on a variety of drugs and an extensive archive, Drugs-Forum provides an ideal information source for public health researchers (Corazza et al., 2012). We limit our study to the English portion of the website.

Discussion threads are organized into numerous subforums, including drugs, the law, addiction, and current events. Since we are modeling drug use, we focus on the drug forums. Each thread is assigned to a drug-specific subforum and each thread has a user-specified tag, which can indicate categories like "Effects" as well as routes of administration like "Oral." We organized the tags and subforum categorizations into factors and components, as shown in Table 4.1. We make use of these tags in Section 4.1.3.

4.1.2 Factorial LDA for Drug Information

For this application, we will use a three-dimensional ($K = 3$) factorial LDA model where the three factors correspond to (1) the drug type, (2) the route of administration, and (3) other aspects of drug use (e.g., dosage, side effects). This allows us to model information jointly related to all three factors, because the effects and other aspects of drugs can vary by route of administration. For example, oral consumption of drugs often produces longer lasting but milder effects than injection or smoking. Many mephedrone users report nose bleeds and nasal pain as a health effect of snorting the drug: this could be modeled as the triple (MEPHEDRONE,SNORTING,HEALTH), a particular combination of all three factors.

Concretely, the prior for the $\vec{t}$th word distribution (step 2a of the generative story in Section 3.1) is defined as:

$$\tilde{\phi}_{\vec{t}v} = \exp\left(\omega^{(B)} + \omega_w^{(0)} + \omega_{t_1 v}^{(\text{drug})} + \omega_{t_2 v}^{(\text{route})} + \omega_{t_3 v}^{(\text{aspect})}\right) \tag{4.1}$$

where $\omega^{(B)}$ is a corpus-wide bias scalar, $\omega_v^{(0)}$ is a corpus-specific bias for word v, and $\omega_{t_k v}^{(k)}$ is a bias parameter for word v for component t_k of the kth factor. That is, each drug, route, and aspect has a weight vector over the vocabulary, and the prior for a particular triple is influenced by the weight vectors of each of the three factors.

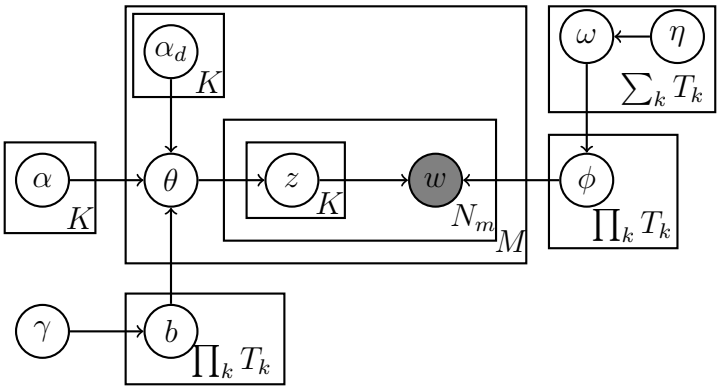


Figure 4.1: The graphical model for FLDA augmented with priors η learned from labeled data (see Section 4.1.3).

The prior for the tuple distribution in each mth document is defined as:

$$\begin{aligned}\tilde{\theta}_{m\vec{t}} = \exp\big(\alpha^{(B)} &+ \alpha_{0t_1}^{(\text{drug})} + \alpha_{mt_1}^{(\text{drug})} \\ &+ \alpha_{0t_2}^{(\text{route})} + \alpha_{mt_2}^{(\text{route})} \\ &+ \alpha_{0t_3}^{(\text{aspect})} + \alpha_{mt_3}^{(\text{aspect})}\big)\end{aligned} \tag{4.2}$$

where $\alpha^{(B)}$ is a global bias parameter, while the α_0 vectors are corpus-wide weight vectors and α_m are document-specific weight vectors over the components of each factor.

The way we are describing this assumes that factors and their components will respond to very specific concepts (drugs, routes, and aspects), yet FLDA is unsupervised. In the next subsection, we will describe a method of semi-supervision that aligns the various components with the intended concepts.

4.1.3 Incorporating Prior Knowledge

In an unsupervised setting, there is no reason FLDA would actually infer parameters corresponding to the three factors we have been describing. However, the forums include metadata that can help guide the model: the messages are organized into forums corresponding to drug type (factor 1), and some threads are tagged with labels corresponding to routes of administration and other aspects (factors 2 and 3). We manually grouped tags into components for the aspect factor. For example, the forum tags "Dose", "Storing", or "Weight" are categorized as the USAGE aspect. We also manually selected tags for routes of administration, but only one tag was needed for each route, unlike aspects whose tags were finer-grained. Table 4.1 shows the factors and components in our model.

One could use these tags as labels in a simple supervised model—this will be our experimental baseline (Section 4.1.4.3). However, this approach has limitations in that most documents are missing labels (less than a third of our corpus contains one of the labels in Table 4.1) and many messages discuss several components, not just the one implied by the tag. For example, a message tagged as "Side effects" may talk about both side effects and dosage. While a standard supervised classifier would attribute all words to a single tag, FLDA learns per-token assignments.

We will instead use the tags to inform the priors over the FLDA word distribution parameters. We do this with a two-stage approach. First, we use the tags to train parameters of a supervised model. We then use the learned parameters as priors over the corresponding FLDA parameters.

In particular, we will place priors on the $\boldsymbol{\omega}$ vectors, the Dirichlet hyperparameters which influence the word distributions. Suppose that we are given a vector $\eta^{(0)}$ which is believed to contain desirable values for $\omega^{(0)}$, the weight vector over words in the corpus, and similarly we are given vectors $\eta_i^{(f)}$ over the vocabulary for the ith component of factor f, which are believed to be good values for $\omega_i^{(f)}$. One option is to fix ω to η, forcing the component weights to match the provided weights. However, in our case $\boldsymbol{\eta}$ will only be an approximation of the optimal component parameters since it is estimated from incomplete data (only some messages have tags) and the $\boldsymbol{\eta}$ vectors are learned using an approximate model (see below). Instead, these weight vectors will merely guide learning as prior knowledge over model parameters $\boldsymbol{\omega}$. While FLDA assumes each $\boldsymbol{\omega}$ is drawn from a 0-mean normal distribution, we alter the means of the appropriate $\boldsymbol{\omega}$ parameters to use $\boldsymbol{\eta}$:

$$\omega_v^{(0)} \sim \mathcal{N}(\eta_v^{(0)}, \sigma^2) \qquad\qquad \omega_{iv}^{(k)} \sim \mathcal{N}(\eta_{iv}^{(k)}, \sigma^2) \tag{4.3}$$

Recall that $\omega_v^{(0)}$ are corpus-wide bias parameters for each vth word and $\omega_{iv}^{(k)}$ are component-specific parameters for each word. This yields a hierarchical prior in which $\boldsymbol{\eta}$ parameterizes the prior over $\boldsymbol{\omega}$, while $\boldsymbol{\omega}$ parameterizes the prior over $\boldsymbol{\phi}$ (the word distributions). The resulting $\boldsymbol{\omega}$ parameters can vary from the provided priors to adapt to the data. An example of learned parameters is shown in Figure 4.2, illustrating the hierarchical process behind this model.

COCAINE	SNORTING	HEALTH	
η (Prior over ω)			
coke	snort	kidney	
cocaine	snorting	hcv	
crack	snorted	pains	
cola	nose	symptoms	COCAINE
blow	nasal	guidelines	SNORTING
lines	drip	diet	HEALTH
ω (Prior over ϕ)			ϕ (Posterior)
coke	snort	symptoms	nose
cocaine	snorting	long-term	cocaine
crack	snorted	depression	coke
cola	passages	disorder	blood
rocks	nostril	schizophrenia	water
coca	insufflating	severe	pain

Figure 4.2: Example of parameters learned by FLDA on the drug forum data. The highest weight words in the ω and η vectors for three components are shown on the left. These are combined to form the prior for the word distribution ϕ. The tripling of (COCAINE,SNORTING,HEALTH) results in high probability words about nose bleeds and nasal damage.

4.1.3.1 Learning priors with a supervised model

In various applications, priors can come from many different sources, such as labeled data (Jagarlamudi et al., 2012). We learn the prior means $\boldsymbol{\eta}$ from tagged messages. However, these parameters imply a latent division of responsibility for observed words: some are present because of the tag while others are general words in the corpus. As a result, the parameters must be estimated with an appropriate model.

We learn these parameters from the tagged messages using a model based on SAGE (Eisenstein et al., 2011), which we described in Section 2.3.3.2. SAGE models word distributions as log-linear combinations of background and topic word distributions. While we described SAGE as an alternative to topic models, Eisenstein et al. (2011) also showed how

it can be used as a supervised Naive Bayes-like model. We take advantage of this flexibility to model the property that a document has multiple factors which are given as labels across the entire document—the drug type and the tag, which could correspond to a component of either the route or aspect factors. We posit the following model of text generation per document:

$$P(\text{word} = v|\text{drug} = i, \text{factor} f = j) = \frac{\exp(\eta_v^{(0)} + \eta_{iv}^{(\text{drug})} + \eta_{jv}^{(f)})}{\sum_{v'} \exp(\eta_{v'}^{(0)} + \eta_{iv'}^{(\text{drug})} + \eta_{jv'}^{(f)})} \tag{4.4}$$

This log-linear model has a similar form as Eq. 4.1, but with two factors instead of three, and it is a distribution rather than a Dirichlet vector. As in SAGE, we fix $\eta^{(\mathbf{0})}$ to be the observed vector of corpus log-frequencies over the vocabulary, which acts as an "overall" weight vector, while parameter estimation yields $\eta_i^{(f)}$, the parameters for the ith component of factor f. SAGE also models sparsity on the weights via a Laplace prior, but such sparsity is not used in the $\boldsymbol{\omega}$ vectors of FLDA, so we instead use a 0-mean normal prior over $\boldsymbol{\eta}$ to more closely match FLDA.

These $\boldsymbol{\eta}$ parameters, learned from a supervised model that is simpler than FLDA, are then used as the mean of the normal priors over $\boldsymbol{\omega}$.

As with the hyperparameter optimization of FLDA, gradient ascent can be used to estimate the parameters of our supervised model. The partial derivative of the log-likelihood

$\mathcal{L}$ with respect to the parameter $\eta_{iw}^{(\text{drug})}$ is:

$$\frac{\partial \mathcal{L}}{\partial \eta_{iv}^{(\text{drug})}} = \sum_{f} \sum_{j \in f} c(i, j, v) - \pi(i, j, v) c(i, j, *) \tag{4.5}$$

where $c(i, j, v)$ is the number of times vth word appears in documents labeled with i (drug) and j (tag), and $\pi(i, j, w)$ denotes the probability given by Eq. 4.4. The partial derivative of each $\eta_j^{(f)}$ is similar; the summation is over different components.

4.1.3.2 Alternative approaches to prior knowledge

The approach described above, in which one model's parameters are used to create priors for another, is not necessarily the most natural or efficient, as it requires the use of two different models in a pipeline, nor is it necessarily the most effective approach, since the model used to learn the prior (SAGE) is different from the target model (FLDA). We will briefly discuss some alternative approaches that could be considered.

One alternative is to keep the two-step approach described above, but using FLDA rather SAGE to learn the $\boldsymbol{\eta}$ parameters used for priors, so that the model is the same and there is less of a mismatch between $\boldsymbol{\eta}$ and good $\boldsymbol{\omega}$ values. This could be done by learning FLDA on the labeled data using a Gibbs sampler that is constrained to only tuples that are consistent with the labels (that is, the drug and the route or aspect match the label). Adding constraints to the possible Gibbs sampling assignments is a form of *posterior regularization* (Ganchev et al., 2010). The $\boldsymbol{\omega}$ values learned from the labeled data could then be used to

create priors for ω in the second FLDA model.

Rather than learning two separate FLDA models, one could train a single FLDA model on both the labeled and unlabeled data, which is perhaps a more natural model. As described in the previous paragraph, the Gibbs sampling assignments could be constrained for the labeled data, to help learn the correct tuples. This is perhaps the most obvious approach, but is not likely to work well by itself, because the amount of unlabeled data is much larger than the labeled data, so the sampler counts from the labeled data may not by large enough to influence the full sampler. The two-step approach was originally used to avoid this problem, because the priors learned from the labeled data can influence the model for the unlabeled data independent of their difference in size. However, other techniques could also be used to help learning with more unlabeled data than labeled. For example, the Gibbs sampler could initially be run on only labeled data, and larger amounts of unlabeled data could gradually be added during sampling, biasing the sampler toward the labeled data. This is similar to Nigam et al.'s (2000) approach to combining labeled and unlabeled data.

4.1.4 Experiments

Our corpus consists of messages from `drugs-forum.com`, as described in Section 4.1.1.1. The site categorizes threads into many drug-specific subforums which are categorized hierarchically. We treat higher-level categories with pharmacologically similar drugs as a single drug type (e.g., OPIOIDS, AMPHETAMINES); for others we took the finest-

granularity subforum as the drug type. We selected 22 popular drugs and from these forums, from which we crawled 410K messages. We selected a subset of tags to form components for the route and aspect factors, as shown in Table 4.1. (Some tags were too general or infrequent to be useful.) We also included a GENERAL component in the latter two factors to model word usage which does not pertain to a particular route or aspect; the prior parameters $\boldsymbol{\eta}$ for these components were simply set to $\mathbf{0}$.

4.1.4.1 Experimental details

Because FLDA does not rely on tagged data (the tags are only used to create priors), we can run inference on larger sets of data. The drawback is that despite these priors, it is still mostly unsupervised and we want to be careful to ensure the model will learn the patterns we care about. We thus add some reasonable constraints to the parameter space to guide the model further.

First, we treat the drug type as an observed variable based on the subforum the message comes from. For example, only tuples of the form (SALVIA,$*$,$*$) can be assigned to tokens in the salvia subforum. Second, we restrict the set of possible routes of administration that can be assigned to tokens in particular drug forums, since most drugs can be taken through only a subset of routes. For example, marijuana is typically smoked or eaten orally, but not injected. We therefore restrict each drug's allowable set of administration routes to those which are tagged (e.g., with "Oral" or "Snorting") in at least 1% of that drug's data. Even though FLDA can learn sparsity, we find that we can improve this by manually specifying

these known sparsity constraints. Similar ideas are used in Labeled LDA (Ramage et al., 2009), in which tags are used to restrict the space of allowed topics in a document.

Each message in a thread was considered a separate document, and we only used documents with at least five word tokens after stop-words, punctuation and low-frequency words were removed.

All FLDA instances were run with 5000 iterations alternating between a sweep of Gibbs sampling followed by a step of gradient ascent on the hyperparameters. While we do not use the tags as strict labels during sampling, we initialized the Gibbs sampler so that each token in a document is assigned to its label given by the tag, when available. In the absence of tags, we initialized tokens to the GENERAL components. We initialized $\boldsymbol{\omega}$ to its prior mean (Eq. 4.3), while the regularization variance and the initialization of bias $\omega^{(B)}$ are chosen to optimize likelihood on the held-out development set.

We optimized the hyperparameters and sparsity array using gradient descent after each Gibbs sweep. We use a decreasing step size of $a/(i+1000)$, where i is the current iteration and $a = 10$ for $\boldsymbol{\alpha}$ and $a = 1$ for $\boldsymbol{\omega}$ and the sparsity values. To learn the priors $\boldsymbol{\eta}$, we ran our version of SAGE for 100 iterations of gradient ascent with a fixed step size of 0.1.

4.1.4.2 Topic model validation

4.1.4.2.1 QUANTITATIVE EVALUATION

We experimented with two predictive tasks to measure the degree to which the semi-supervised FLDA variant improves over the standard unsupervised model. Note that we

Model	Perplexity	Accuracy	MRR
Unsupervised	1765	14%	0.37
Semi-supervised	1730	41%	0.62

Table 4.2: Quantitative comparison of the unsupervised FLDA model with the semi-supervised variant proposed in this chapter (Section 4.1.3) on the drug data.

say "unsupervised" to refer to the standard FLDA model that does not include the prior knowledge described in Section 4.1.3 (nor are the tuple assignments initialized based on the prior), but we still applied the constraints to the parameter space (on the selection of drug and route of administration) described above.

For these experiments, we randomly selected a total of 100K messages to train the topic model and evaluated on 25K held-out messages.

First, we computed perplexity of held-out text. Second, we measured how well the model can predict the observed tags of threads, both in accuracy (how often the true tag was the model's most likely component) as well as the mean reciprocal rank (MRR) of the true tags. For the unsupervised model, we used a post-hoc greedy matching to determine which model components corresponded to which tag, based on the Jensen-Shannon divergence between each component's marginal distribution and the distribution defined by the prior.

Table 4.2 shows that the semi-supervised model provides better predictive abilities. While we would expect the model to have higher accuracy at predicting tags, it is also notable that it has lower perplexity of text, as this means these concepts better generalize to new text.

4.1.4.2.2 QUALITATIVE ANALYSIS

In addition to these quantitative experiments, Table 4.3 shows examples of word distributions learned. We present examples of the resulting tuples by selecting the top 6 words for each tuple. We focus on common drugs so that they can be interpreted by the reader, but the next subsection will focus on the novel drugs that motivated this application, as described in Section 4.1.1.

The structured output itself appears more informative than a flat list of topics to a researcher. This output breaks down words for each drug into route of administration and aspect. For example, the cocaine component distinguishes words by routes: smoking ("pipe", "rock") vs. snorting ("nose", "powder"), and aspects: chemistry ("acetone", "water") vs. health ("addiction", "brain"). Additionally, the labels for drug, route, and aspect are not assigned manually, but taken from the prior; this saves time and clarifies the output. The tuples clearly correspond to the labeled components.

An examination of output reveals several patterns of drug use, such as:

- **Cocaine:** The delivery methods reveal different types of cocaine. The SMOKING component has the words "crack" and "rock", while the SNORTING component has the words "coke", "powder" and "lines".

- **Cannabis:** The oral method includes words about marijuana brownies; the tuple (CANNABIS, ORAL, CHEMISTRY) contains words related to baking, such as "butter" and "milk", which are particular to this delivery method.

- The **culture** components reflect differences in the culture surrounding drugs. ECSTASY contains words related to raves and nightclubs, and OPIOIDS, which includes heroin, has words about addiction and street life ("money", "dealer", "junkie").

- The **health** components highlight health issues surrounding different types of drugs. COCAINE and OPIOIDS both include words about addiction, while CANNABIS includes words about mental health ("mental", "anxiety", "psychosis"). We also find health words that are specific to certain delivery methods: the tuple (COCAINE, SNORTING, HEALTH) includes words about nose and sinus damage, and (CANNABIS, SMOKING, HEALTH) includes the words "cancer", "lung", and "lungs".

4.1.4.3 Information summarization

We wish to demonstrate that our modified FLDA model can be used to discover useful information in the text. One way to demonstrate this is by using the model to extract relevant snippets of text from the forums. This is the basis of our evaluation experiments in this subsection. Our goal is not to build a complete summarization system, but rather to use the model to direct researchers to interesting messages.

While we model all 22 drugs, our summarization experiments will focus on five drugs which have been studied only relatively recently: mephedrone and MDPV (β-ketones), Bromo-Dragonfly (synthetic phenethylamines), Spice/K2 (synthetic cannabinoids), and salvia divinorum. We consider these drugs in particular because these are the five drugs

Route of Administration	Aspect: GENERAL	CHEMISTRY	CULTURE	EFFECTS	HEALTH	USAGE
	CANNABIS					
ORAL	weed	butter	friend	trip	sleep	time
	high	oil	went	experience	cannabis	pot
	eat	heat	night	lsd	dreams	hours
	eating	water	friends	hallucinations	memory	gram
	brownies	milk	home	psychedelic	effects	half
	work	mix	room	intense	experience	grams
SMOKING	tobacco	pipe	said	time	smoking	smoke
	joint	glass	marijuana	smoked	marijuana	bong
	weed	bowl	drug	weed	smoke	hit
	joints	water	police	felt	cannabis	bowl
	smoke	bottle	law	first	cancer	hits
	roll	hole	store	high	cause	smoking
	COCAINE					
SMOKING	crack	water	went	friend	body	
	smoke	soda	thought	time	eat	
	smoking	baking	house	weed	weight	
	pipe	freebase	car	smoking	eating	
	hit	spoon	shit	high	food	
	rock	rock	home	says	help	
SNORTING	nose	dry	smell	feel	nose	coke
	window	filter	card	coke	pain	line
	water	plate	bathroom	heart	damage	lines
	nasal	paper	coke	felt	blood	nose
	spray	powder	white	feeling	cocaine	small
	mouth	fine	bag	time	problem	cut
	MDMA (ECSTASY)					
GENERAL	time	serotonin	music	mdma	drug	pills
	really	mdma	rolling	experience	drugs	mdma
	first	effects	rave	time	mdma	pill
	feel	dopamine	people	people	people	test
	friend	brain	great	experiences	effects	ecstasy
	doesnt	receptors	mp3	feeling	depression	pure
	LSD (ACID)					
GENERAL	time	lsd	music	trip	experience	lsd
	acid	effects	tripping	experience	people	blotter
	friends	mescaline	movie	tripping	mind	blotters
	trip	psychedelic	love	time	think	dose
	friend	receptors	listening	first	lsd	taste
	felt	visual	watch	trips	way	dox
	OPIOIDS					
GENERAL	dont	pods	heroin	feeling	depression	dose
	know	tea	life	feel	drug	tolerance
	people	opium	years	time	drugs	opiates
	think	poppy	time	felt	treatment	opiate
	really	seeds	day	really	patients	high
	youre	pod	money	high	effects	doses
INJECTION	needle	water	dope	minutes		codeine
	vein	filter	time	later		pills
	veins	solution	shit	added		apap
	injecting	liquid	bag	seconds		liver
	blood	powder	know	hours		cwe
	hit	heat	going	10		acetaminophen

Table 4.3: Example output from a sample of pertinent routes of administration from five drug types. Darkened boxes indicate sparse tuples in which $b < 0.2$.

for which technical reports were created by the EU Psychonaut Project (Schifano et al., 2006), an online database of novel and emerging drugs, whose information is collected by reading drug websites, including Drugs-Forum. Extensive technical reports were written about these five popular drugs, and we can use these reports to produce reference summaries for our experiments.

Of these five drugs, only salvia has its own subforum; the others belong to subforums representing the broader categories shown in parentheses. We simply model the drug type as a proxy for the specific drug, as most of the drugs in each category have similar effects and properties. The first two drugs are both in the same subforum, so for the purpose of our model we treat mephedrone and MDPV as the single drug type, β-ketones. These two drugs are grouped together during summarization, but the corresponding reference summaries incorporate excepts from the technical reports on both drugs.

Of the four drug types being considered for summarization, our data set contains 12K messages with one of the tags in Table 4.1 and 30K without. Of those without tags, we set aside 5K as development data. There are also over 300K messages (140K tagged) from the remaining 18 drug types: some of these messages are utilized when training FLDA. Even though we only consider four drug types in our experiments, our intuition is that it can be beneficial to model other drugs as well, because this will help to learn parameters for the various aspects and routes of administration. For instance, our model of the effects of mephedrone can be informed by also modeling the effects of other stimulants such as cocaine.

4.1.4.3.1 MODEL DESCRIPTIONS

Our baseline model is a unigram language model trained on the subset of messages which are tagged. We treat the drug subforum as a label for the drug factor, and each message's tag is used as a label for either the route or aspect factor. For example, the word distribution for the pair (SALVIA,EFFECTS) is estimated as the empirical distribution from messages posted in the salvia forum and tagged with "Effects." We use pseudocount smoothing where the pseudocount size is chosen to optimize likelihood on the held-out development set.

This is a two-dimensional model, since we explicitly model pairs such as (MEPHEDRONE, SNORTING) or (SALVIA,EFFECTS). However, we also created word distributions for triples such as (SALVIA,ORAL,EFFECTS) by estimating the unigram model from the union of documents with these tags, e.g., documents tagged with either (SALVIA,ORAL) or (SALVIA, EFFECTS).

We use FLDA as a three-dimensional model which explicitly models triples, but we also obtain distributions for pairs such as (SALVIA,EFFECTS) by marginalizing across all distributions of the form (SALVIA,$*$,EFFECTS). We trained FLDA on two different data sets, yielding the following models:

- **FLDA-1:** We use the 12K messages with tags and fill the set out with 13K messages with tags uniformly sampled from the 18 other drugs, for a total of 25K messages.
- **FLDA-2:** We use all 37K messages (many without tags) and fill the set out with 63K

Algorithm 1 Our extractive summarization algorithm. The SUMMARIZE function takes as input a set of text snippets $\mathcal{D}$, where each snippet $\boldsymbol{d} \in \mathcal{D}$ is a vector (length V) of word counts in the snippet, as well as a tuple-specific word distribution $\phi_{\vec{t}}$. The function returns a set $\mathcal{S} \subset \mathcal{D}$ of 5 snippets that have low KL-divergence with the target word distribution.

function SUMMARIZE($\mathcal{D}, \phi_{\vec{t}}$)
  $\mathcal{S} \leftarrow \emptyset$
  for $i \leftarrow 1$ to 5 **do**
    $\boldsymbol{s} \leftarrow \text{argmin}_{\boldsymbol{d} \in \mathcal{D}}$ SCORE($\boldsymbol{d}, \mathcal{S}, \phi_{\vec{t}}$)
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{s}\}$
  return $\mathcal{S}$
function SCORE($\boldsymbol{d}, \mathcal{S}, \phi_{\vec{t}}$) ▷ KL-divergence between candidate summary and $\phi_{\vec{t}}$
  $\boldsymbol{n} \leftarrow \mathbf{0}$
  for $\boldsymbol{s} \in \mathcal{S} \cup \{\boldsymbol{d}\}$ **do**
    for $v \leftarrow 1$ to V **do**
      $n_v \leftarrow n_v + s_v$ ▷ $\boldsymbol{n}$ is the count vector of the candidate summary $\mathcal{S} \cup \{\boldsymbol{d}\}$
  for $v \leftarrow 1$ to V **do**
    $\pi_v \leftarrow \frac{n_v}{\sum_{v'} n_{v'}}$ ▷ π is the empirical word distribution of the candidate summary
  return $KL(\pi || \phi_{\vec{t}})$

messages with tags uniformly sampled from the 18 other drugs, for a total of 100K messages.

4.1.4.3.2 SUMMARY GENERATION

We created twelve reference summaries by editing together excerpts from the five Psychonaut Project reports (Psychonaut, 2009). Each reference is matched to drug-specific pairs and triples. For example, a paragraph describing the differences in effects of salvia between smoking and oral routes was matched to distributions for (SALVIA,EFFECTS), (SALVIA,SMOKING,EFFECTS), (SALVIA,ORAL,EFFECTS). Descriptions of creating tinctures and blotters for oral consumption were matched to (SALVIA,ORAL,CHEMISTRY). We consider pairs in addition to triples because not all summaries correspond to particular

routes or aspects.

For each tuple-specific word distribution (a pair or a triple), we create a "summary" by extracting a set of five text snippets which minimize KL-divergence to the target word distribution. We consider all overlapping text windows of widths $\{10,15,20\}$ in the corpus as candidate snippets. Following Haghighi and Vanderwende (2009), we greedily add snippets one by one with the lowest KL-divergence at each step until we have added five. The pseudocode is shown in Algorithm 1.

We only considered candidate snippets within the subforum for the particular drug, and snippets are based on the preprocessed topic model input with no stop words. Before presenting snippets to users, we then map the snippets back to the raw text by taking all sentences which are at least partly spanned by the window of tokens. Because each reference may be matched to more than one tuple, there may be more than five snippets which correspond to a reference.

Recall that the reports used as reference summaries were themselves created by reading Web forums. Our hypothesis is that FLDA could be used as an exploratory tool to expedite the creation of these reports. Thus in our evaluation we want to measure how useful the extracted snippets would be in informing the writing of such reports. We performed both human and automatic evaluation on the summaries generated by FLDA (variants 1 and 2) as well as our baseline. We also included randomly selected snippets as a control (five per reference).

Example output is shown in Table 4.4.

Reference Text	**System Snippet**
Mephedrone (β-ketones/Bath salts)	
It is recommended by users that Mephedrone be taken on an empty stomach. Doses usually vary between 100mg–1g.	• If it is SWIY's first time using Mephedrone SWIM recommends a 100mg oral dose on an empty stomach.
Reported negative side effects include: • Loss of appetite. • Dehydration and dry mouth • Tense jaw, mild muscle clenching, stiff neck, and bruxia (teeth grinding) • Anxiety and paranoia • Increase in mean body temperature (sweating/Mephedrone sweat and hot flushes) • Elevated heart rate (tachycardia) and blood pressure, and chest pains • Dermatitis like symptoms (Itch and rash)	• Neutral side effects: Lack of appetite, occasional loss of visual focus, [...] weight loss, possible diuretic. Negative side effects: Grinding teeth, "Cotton mouth", unable to acheive orgasm • Aside from his last session he has never experienced any negative symptoms at all, no raised heart beat, vasoconstriction , sweating, headaches, paranoia e.t.c nothing at all except sometimes cold hands the next day. • lot of people report that anxiety and paranoia are some of the side effects of taking mephedrone [...] is it also possible that alot of the chest pains people are experiencing is due to anxiety? • moisturize the affected areas of skin twice daily with E45 or a similar unperfumed dermatalogical lotion.
Salvia divinorum	
Sublingual ingestion of the leaf (quid): reduces intensity of effects and can taste disgusting. When Salvia is consumed as a smokeable formulation the duration of the trip lasts 30 minutes or less, whereas if Salvia is consumed sublingually the effects lasts for 1 hour or more.	• The taste of sublingual salvia is foul and it is easy to have a dud trip unless large amounts of it are used. • SWIM has heard from many other users that chewing the fresh leaves of the Salvia plant allow for a much longer and mellower trip. [...] SWIM has read that a trip this way can last anywhere from a half on hour or longer.
Dried leaves and/or salvia extract are smoked (using a butane lighter) either by pipe (considered to be the most effective but is considered to be quite painful) or water bong.	• 2. Use a water pipe. Its harsh and needs to be smoked hot so this should be self explanatory. 3. Use a torch style lighter [...] Salvinorin A has a VERY high boiling point (around 700 degrees F I believe) so a regular bic just wont do it

Table 4.4: Example drug snippets generated by FLDA along with the corresponding reference text. For space, the references and snippets shown have been shortened in some cases. "SWIM" and "SWIY" stand for "someone who isn't me/you" and are used to avoid self-incrimination on the Web forum.

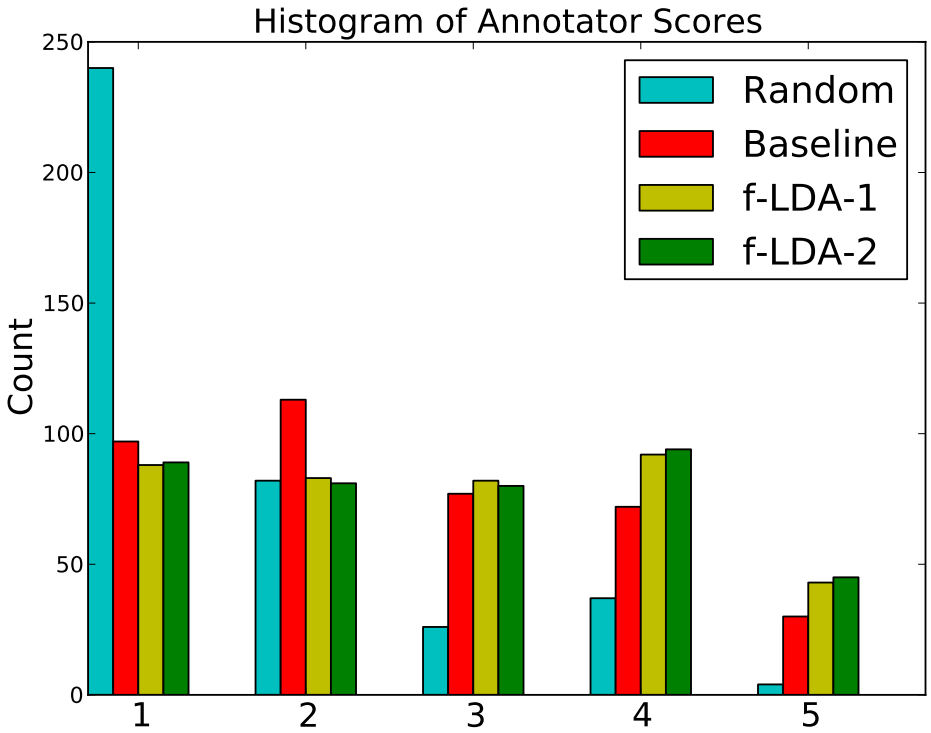


Figure 4.3: The distribution of annotator scores for the summarization task (Section 4.1.4.3.3). The "Random" counts have been scaled to fit the same range as the other systems, since fewer random snippets were shown to annotators.

4.1.4.3.3 HUMAN JUDGMENTS OF QUALITY

Three annotators were presented snippets from all four systems alongside the corresponding reference text. Within each set corresponding to a reference summary, the snippets were shown in a random order. Annotators were asked to judge each snippet independently on a 5-point Likert scale as to how useful each snippet would be in writing the reference text.

The distribution of scores is shown in Figure 4.3 and summarized in Table 4.5. Annotators generally agreed on the relative quality of snippets: the average correlation of scores between each pair of annotators was 0.49. Snippets produced by FLDA were given more

high scores and fewer low scores than the baseline, while the two FLDA variants were rated comparably. The breakdown is more interesting when we compare scores for snippets that were matched to word distributions for pairs versus word distributions for triples. The gap in scores between FLDA and the baseline increases when we look at the scores for only triples: FLDA beats the baseline by a margin of 0.45 for snippets matched to triples and 0.21 for pairs. This suggests that we produce better triples by modeling them jointly. For triples, FLDA-2 (which uses more data) beats FLDA-1 (which uses only tagged data), while the reverse is true for pairs, suggesting that more data is helpful for modeling triples.

While some of the randomly selected control snippets happened to be useful, the scores for these snippets were much lower than those extracted through model-based systems. This suggests that exploring the forums in a targeted way (e.g., through our topic model approach) would be more efficient than exploring the data in a non-targeted way (akin to the random approach).

Finally, we asked two expert annotators (faculty members in psychiatry and behavioral pharmacology, who have used drug forums in the past to study emerging drugs) to rate the snippets corresponding to mephedrone/MDPV. The best FLDA system had an average score of 2.57 compared to a baseline score of 2.45 and random score of 1.63.

4.1.4.3.4 AUTOMATIC EVALUATION OF RECALL

The human judgments effectively measured a form of precision, as the quality of snippets were judged by their correspondence to the reference text, without regard to how much

	Random	**Baseline**	**FLDA-1**	**FLDA-2**
	Annotator Scores			
Mean	1.67	2.55	2.79	**2.81**
Pairs only	n/a	2.58	**2.79**	2.72
Triples only	n/a	2.50	2.80	**2.95**
	ROUGE			
1-gram	.112	.326	**.355**	.327
2-gram	.023	.072	**.085**	.084

Table 4.5: Summary quality evaluation across four systems.

of the reference text was covered by all snippets. We also used the automatic evaluation metric ROUGE (Lin, 2004) as a rough estimate of summary recall: this metric computes the percentage of n-grams in the reference text that appeared in the generated summaries.

We computed ROUGE for both 1-grams and 2-grams. When computing n-gram counts, we applied Porter's stemmer to all tokens. We excluded stop words from 1-gram counts but included them in 2-gram counts where we care about longer phrases.

Results are shown in Table 4.5. We find that FLDA-1 has the highest score for both 1- and 2-grams, suggesting that it is extracting a more diverse set of relevant snippets. When performing a paired t-test across the 12 reference summaries, we find that FLDA is better than the baseline with p-values 0.14 and 0.10 for 1-gram and 2-gram recall, respectively. FLDA's recall advantage may come from the fact that it learns from a larger amount of data and it may learn more diverse word distributions by directly modeling triples. FLDA-1 had slightly better recall (under ROUGE), while FLDA-2 was slightly better according to the human annotators.

4.2 Measuring Healthcare Quality from Online Reviews

Our second application analyzes patient reviews of doctors using a two-dimensional factorial LDA model that jointly captures topic and sentiment. This section is based on material from Paul et al. (2013) and Wallace et al. (2014).

Our aim is to elucidate the issues that most affect consumer sentiment regarding interactions with their doctor using a dataset of over 50,000 online doctor reviews. Reviews include ratings along various aspects: staffing, helpfulness and knowledgeability. We use FLDA to infer the text associated with strong sentiment along these aspects to illustrate the factors that most influence patient satisfaction.

Following the semi-supervised approach of the previous application, we use a small set of manual annotations created for a previously conducted qualitative study of online provider reviews (López et al., 2012) to bias the model parameter estimates. Additionally, we show that user ratings from reviews can be incorporated into priors to additionally guide the parameters. Our experiments in Section 4.2.4 show that FLDA learns intuitive concepts that are correlated with real-world measures of healthcare quality.

4.2.1 Task and Motivation

Individuals are increasingly turning to the Web for healthcare information. A recent survey (Fox and Duggan, 2013) found that 72% of internet users have looked online for health information in the past year, and one in five for reviews of particular treatments or doctors. In a random sample of 500 urologists, online reviews were found to have been written about 80% of them (Ellimoottil et al., 2012). These numbers will likely increase in coming years.

The shift toward online health information consumption and sharing has produced a proliferation of health-related, user-generated content, including online doctor reviews. Such reviews have clear value to patients, but they are also valuable in that taken *en masse* they may reveal insights into factors that affect patient satisfaction. For example, in an analysis of online healthcare provider reviews, López et al. (2012) noted that comments regarding interpersonal manner and technical competence tended to be more positive, whereas comments about systems issues (e.g., regarding office staff) tended to be more mixed.

There has been a flurry of recent research concerning online physician-rating websites (Segal et al., 2012; Emmert et al., 2013; López et al., 2012; Galizzi et al., 2012; Ellimoottil et al., 2012). We have already discussed the work by López et al. (2012): perhaps their most interesting finding was that reviews often concern aspects beyond the patient-doctor relationship (e.g., office staff). They also found that well-perceived bedside manner was a key to successful patient-doctor interactions. Segal et al. (2012) analyzed the relationship between high-volume surgeons (who perform many operations) and online reviews. Not-

ing that surgeons who perform more procedures tend to have better clinical outcomes and safety records, they found that they could both identify high-volume surgeons using online reviews, and that high-volume surgeons tended to receive more praise.

A drawback to existing explorations of online provider reviews is that they have been primarily qualitative in nature. This approach limits the potential scope of analysis, and has precluded conduct of the sort of larger-scale analyses necessary to comprehensively elucidate the content of online doctor reviews.

In this section, we apply factorial LDA to the application of understanding online doctor reviews. We will use a two-dimensional version of FLDA to model the dimensions of topic and sentiment, and as with the model for the drug forum data in the previous section (4.1.3), we will create informed priors using labeled data to ground the model parameters in concepts of interest.

Topic models have previously been used to identify themes in doctor reviews (Brody and Elhadad, 2010), but this previous work did not also model sentiment and did not take advantage of labeled data as we do here. Models that can tease out sentiment across different aspects from free-form text would also facilitate automatic monitoring of performance.

In addition to experimenting with topic models for identifying themes and sentiment, in Section 4.2.4.3 we will compare the information learned by FLDA to ground truth data regarding healthcare quality across the 50 U.S. states.

ratings	*review text*
5 5 5	Dr. *X* has a gentle and reassuring manner with the kids, her office staff is prompt, pleasant, responsive, and she seems very knowledgeable.
1 2 1	We were told outright that my wife, without question, did not have a uterine infection. She was discharged. 4 hours later she was very sick. We went back to triage and lo and behold, a uterine infection.

Table 4.6: A positive and negative review from the doctor reviews corpus. Ratings correspond to *helpfulness*, *staff* and *knowledgeability*, respectively; higher numbers convey positive sentiment.

4.2.1.1 Dataset

Our dataset is composed of 52,226 reviews (with 55.8 word tokens on average) downloaded from RateMDs.com, a website of doctor reviews written by patients. Our dataset covers 17,681 unique doctors. Reviews contain free text and numerical scores across different aspects of care (Table 4.6). To achieve wide geographical coverage (since we analyze content across geography in Sections 4.2.4.3–4.2.4.4), we crawled reviews from states with equal probability, i.e., uniformly sampled a US state and then crawled reviews from that state.

4.2.2 Factorial LDA for Reviews

For modeling the doctor reviews, we use a two-dimensional instantiation of factorial LDA, with the two factors corresponding to topic and sentiment. The sentiment factor has two components: POSITIVE and NEGATIVE. This model yields word distributions for different topics paired with positive or negative sentiment values.

The priors for word distributions and tuple distributions are given by:

$$\tilde{\phi}_{\vec{t}v} = \exp\left(\omega^{(B)} + \omega_v^{(0)} + \omega_{t_1 v}^{(\text{topic})} + \omega_{t_2 v}^{(\text{sentiment})}\right) \tag{4.6}$$

$$\tilde{\theta}_{m\vec{t}} = \exp\left(\alpha^{(B)} + \alpha_{0t_1}^{(\text{topic})} + \alpha_{mt_1}^{(\text{topic})} + \alpha_{0t_2}^{(\text{sentiment})} + \alpha_{mt_2}^{(\text{sentiment})}\right) \tag{4.7}$$

Other work has used topic models to jointly model topic (sometimes called aspect) and sentiment. Extending LDA to account for aspect-specific sentiment, Titov and McDonald (2008) considered the general task of jointly modeling text and aspect ratings for sentiment summarization, exploiting supervision by leveraging existing aspect labels. Mei et al. (2007) proposed a mixture model that combines topic and sentiment components through a switch variable; Lu et al. (2009) used a similar approach to summarize sentiment of eBay feedback. However, none of these models use word distributions for the conjunction of topic and sentiment, the way FLDA models (topic, sentiment) pairs.

4.2.3 Incorporating Prior Knowledge

As with the previous task of modeling drug forums, we must incorporate prior knowledge into the model in order to learn factors that correspond to topic and sentiment. We will use the same approach as the previous section, using labeled data to learn priors over the word distributions. Additionally, we will show how user ratings in the review metadata can be used to influence priors over the tuple distributions in documents.

Interpersonal manner		**Technical competence**		**Systems issues**	
positive	*negative*	*positive*	*negative*	*positive*	*negative*
shows empathy, professional, communicates well	poor listener, judgmental, racist	good decision maker, follows up on issues, knowledgeable	poor decision maker, prescribes the wrong medication, disorganized	friendly staff, short wait times, convenient location	difficult to park, rude staff, expensive
professional	judgmental	follows up on issues	unorganized	short wait times	rude staff
gives information	racist	knowledgeable	not punctual	convenient location	expensive

Table 4.7: Illustrative tags underneath the three main aspects identified by López et al. (2012). These aspects are used to create informative priors for our model of doctor reviews, as described in Section 4.2.3.

4.2.3.1 Priors from labeled documents

We make use of an annotated corpus of 842 online reviews previously created by López et al. (2012). The reviews were annotated along three dimensions, illustrated in Table 4.7, using a code set developed by a physician-led team.

López et al. identified three main aspects: *interpersonal manner*, *technical competence* and *systems issues*. Examples of the first include personal demeanor and bedside disposition; the second refers to (perceived) medical quality, and the third refers to logistical issues such as the location of the physician's facility. Our model will use these aspects as the three components of our model's topic factor. (We will also experiment with more than three components, but only the first three will have informed priors.) The dataset also includes annotations of positive or negative sentiment, which will form the two components of the sentiment factor.

We followed the same procedure described in Section 4.1.3 to create priors from these labels. We used a SAGE-style model to estimate parameters from the small set of labeled

reviews of López et al. (2012):

$$P(\text{word} = v|\text{topic} = i, \text{sentiment} = j) = \frac{\exp(\eta_v^{(0)} + \eta_{iv}^{(\text{topic})} + \eta_{jv}^{(\text{sentiment})})}{\sum_{v'} \exp(\eta_{v'}^{(0)} + \eta_{iv'}^{(\text{topic})} + \eta_{jv'}^{(\text{sentiment})})} \tag{4.8}$$

The $\boldsymbol{\eta}$ parameters are then used as the means of normal priors over the $\boldsymbol{\omega}$ parameters in FLDA.

We use the three high-level topic labels described above from the López et al. dataset as components: interpersonal manner (INTERPERSONAL), technical competence (TECHNICAL), and systems issues (SYSTEMS). Each review in the labeled data is labeled with (topic, sentiment) pairs such as (TECHNICAL,NEGATIVE). Some documents have multiple labels; rather than complicating the model to handle label attribution, in these cases we simply duplicate the document for each label, so that each training instance has only one label.

When using FLDA with more than three topics ($T_1 > 3$), we set the corresponding $\boldsymbol{\eta}$ values to $\mathbf{0}$, so they are not influenced by labeled data, since we have no labeled data for such topics.

4.2.3.2 Priors from review scores

The reviews in the RateMDs corpus contain user ratings (integers ranging from 1 to 5) for three categories: knowledgeability, staff, helpfulness. (There was also a rating for punctuality, but this did not directly map to one of the three López et al. aspects, so we

did not incorporate this into our model.) As a novel extension to FLDA for the purpose of this topic-sentiment task, we attempt to leverage these ratings to further guide the model in inferring the different topic and sentiment pairs. The ratings are not quite what we want to model, but provide valuable side information. In this subsection, we show how to incorporate these user ratings into the document priors.

These rating categories naturally correspond to similar labels as in the López et al. dataset, albeit only roughly. We created the following category-to-topic mapping:

- 'Knowledgeability' : TECHNICAL
- 'Staff' : SYSTEMS
- 'Helpfulness' : INTERPERSONAL

For each pair $\vec{t}$ in the mth document, we use the user ratings to create rating variables $r_{m\vec{t}}$ centered around the middle value of 3: for each topic, we set the value of $r_{m\vec{t}}$ for the positive sentiment to be the original user rating minus 3, while the $r_{m\vec{t}}$ value for the negative sentiment is the negation of the positive. For example, if the user rating for 'Staff' was 2, then $r_{m,\text{SYSTEMS,POS}} = -1$ and $r_{m,\text{SYSTEMS,NEG}} = 1$, while if the user rating for 'Helpful' was 3, then the r_m variables for both the positive and negative INTERPERSONAL pairs would be 0. These r variables can thus be used to bias the document's pair distribution toward or away from pairs that have a high or low user rating. The r values are simply set to 0 for topics beyond the first three, for which we do not have ratings.

We incorporate $r_{m\vec{t}}$ into the document's prior over pair distributions, so that topics with

high ratings are *a priori* more likely to contain that topic paired with positive sentiment and less likely to contain that topic paired with negative sentiment. Specifically, we modify the log-linear equation in Eq. 4.7 to include an additional term containing $r_{m\vec{t}}$:

$$\tilde{\theta}_{m\vec{t}} = \exp\left(\alpha^{(B)} + \alpha_{0t_1}^{(\text{topic})} + \alpha_{mt_1}^{(\text{topic})} + \alpha_{0t_2}^{(\text{sentiment})} + \alpha_{mt_2}^{(\text{sentiment})} + \rho\, r_{m\vec{t}}\right) \tag{4.9}$$

where $\rho > 0$ is a scalar parameter that controls how strongly the rating variable should influence the prior.

We optimize ρ to maximize likelihood. For mathematical convenience, we first re-parameterize ρ as $\exp(\hat{\rho})$, allowing us to optimize $\hat{\rho} \in \mathbb{R}$ rather than $\rho \in \mathbb{R}_{>0}$. We also place a 0-mean normal prior on $\hat{\rho}$. The partial derivative of the corpus likelihood $\mathcal{L}$ with respect to $\hat{\rho}$ is:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{\rho}} = &- \frac{\hat{\rho}}{\sigma^2} + \sum_{m=1}^{M} \sum_{\vec{t} \in \mathcal{T}(*)} r_{m\vec{t}}\, \exp(\hat{\rho})\, \tilde{\theta}_{m\vec{t}} \times \\ &\left(\Psi(n_{\vec{t}}^m + \tilde{\theta}_{m\vec{t}}) - \Psi(\tilde{\theta}_{m\vec{t}}) + \Psi(\textstyle\sum_{\vec{u}} \tilde{\theta}_{m\vec{u}}) - \Psi(\sum_{\vec{u}} n_{\vec{u}}^m + \tilde{\theta}_{m\vec{u}})\right)\end{aligned} \tag{4.10}$$

where $n_{\vec{t}}^m$ is the number of times the pair $\vec{t}$ appeared in document m, given the current state of the Gibbs sampler. We optimize this with gradient ascent along with the other hyperparameters of the model.

4.2.4 Experiments

4.2.4.1 Experimental details

All of our Gibbs samplers are run for 5000 iterations with a gradient ascent step size of 10^{-3}. The variance of the normal prior over the parameters was $\sigma^2 = 1$ for $\boldsymbol{\alpha}$ and ρ, $\sigma^2 = 0.5$ for $\boldsymbol{\omega}$. For experiments that compared to LDA, the model was run for the same number of iterations and the Dirichlet hyperparameters of LDA were optimized for likelihood.

We initialized $\alpha^{(B)} = -2$ and $\omega^{(B)} = -6$, the other $\boldsymbol{\omega}$ parameters were initialized to their corresponding $\boldsymbol{\eta}$ values when applicable, and all other hyperparameters were initialized to 0. (The selection of hyperparameters is discussed more in Section 6.4.) Finally, to tilt the model parameters slightly toward the correct sentiment values, we initialized $\alpha_{m,\text{POS}}^{(\text{sentiment})} = 0.1$ if the average user rating across the three categories was ≥ 3 and -0.1 otherwise, with $\alpha_{m,\text{NEG}}^{(\text{sentiment})} = -\alpha_{m,\text{POS}}^{(\text{sentiment})}$.

When training the SAGE-like supervised model on the labeled data, our gradient ascent algorithm was run for 1000 iterations with a step size of 10^{-2}. The normal prior variance over $\boldsymbol{\eta}$ was $\sigma^2 = 0.1$.

4.2.4.2 Topic model validation

4.2.4.2.1 Quantitative evaluation

We validated the FLDA model using two predictive tasks: perplexity of held-out text and predicting the user ratings of held-out reviews.

Our model utilizes two extensions to FLDA. In our experiments, we compare this full model to ablated versions:

- 'B': baseline model without extensions (unsupervised FLDA);
- 'W': model with word priors from labeled data ($\boldsymbol{\eta}$);
- 'R': model with document priors from user ratings ($\boldsymbol{r}$);
- 'WR': full model with both extensions.

We also compared against LDA with comparable numbers of word distributions. For example, when comparing against FLDA with 3 topics, we use 6 topics in LDA, because FLDA in this case has 6 word distributions for the 3 topics paired with 2 sentiment values.

Results are from 5 fold cross-validation where within each fold, we perform 10 inference trials through randomly initialized sampling chains on the training set (80% of the data) and selecting inferred parameters with the lowest perplexity on the held-out set (20% of the data). For inference on the held-out set, we fix all except for document-specific parameters. We run the sampler for 1000 iterations, and then average the parameters sampled from 100 iterations. No information about the user ratings is used during inference on the test set; the 'R' extensions used by the models only apply during training.

	$T_1 = 3$	$T_1 = 6$	$T_1 = 9$
B	**1048.3** ($\pm$ 6.0)	912.1 ($\pm$ 5.2)	842.1 ($\pm$ 5.1)
W	1076.2 ($\pm$ 6.5)	**907.5** ($\pm$ 7.2)	**827.6** ($\pm$ 6.4)
R	1055.0 ($\pm$ 10.3)	917.5 ($\pm$ 5.8)	841.6 ($\pm$ 5.1)
WR	1076.4 ($\pm$ 5.4)	921.0 ($\pm$ 5.3)	835.8 ($\pm$ 5.8)
LDA	1062.2 ($\pm$ 6.2)	936.4 ($\pm$ 5.5)	861.2 ($\pm$ 5.3)

Table 4.8: Average perplexity of held-out data during cross-validation using various models, $\pm$ standard deviation.

Perplexity results are mixed and inconsistent (Table 4.8). The baseline model is the best model when $T_1 = 3$ topics, but the worst when $T_1 = 9$. The 'W' model is the best for 6 and 9 topics, but nearly the worst for 3. We cannot conclude that any of these models consistently do the best at predicting unseen documents.

We also evaluated whether the distributions learned by FLDA are predictive of the user ratings associated with the reviews. To do so, we encoded each review by its distribution over the inferred (topic, sentiment) pairs. Specifically, we represent each review as a feature vector of length $2\times|T_1|$, representing every (topic, sentiment) pair. The tth feature of a review is set to θ_{mt}. Because we have an ordinal outcome (ratings are integers from 1 to 5), we used ordinal logistic regression to predict ratings.

We also experimented with using a standard bag-of-words (BoW) feature encoding, where features were the 1000 most frequently occurring words, and conjunctions of BoW features and the FLDA distribution representation.

Figure 4.4 shows the prediction error of the various models for the three rating categories. The proposed full model almost always outperforms the other models, and the models with extensions almost always outperform the baseline FLDA model in terms of

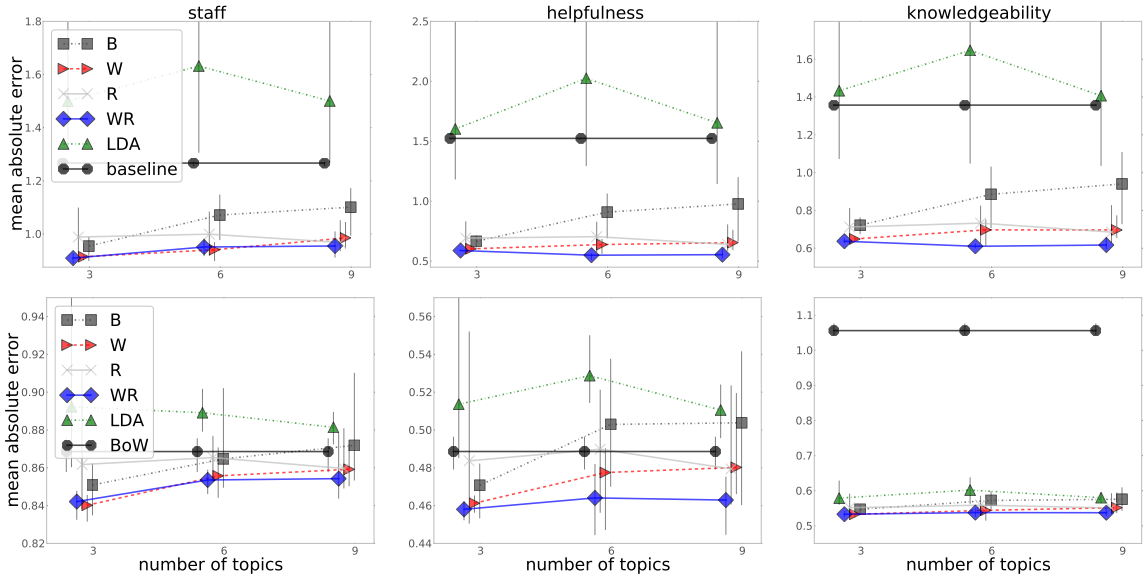


Figure 4.4: Mean absolute errors (markers) and ranges (vertical lines) over five folds with respect to predicting the sentiment scores of held out reviews for three aspects (staff, helpfulness and knowledgeability). **B**: FLDA without priors; **W**: priors over words; **R**: priors using ratings; **WR**: priors over words and ratings. Results include 3, 6, and 9 topics (x-axis). Top row: predictions made using only features representing the inferred distribution over (topic, sentiment) pairs; *baseline* corresponds to simply predicting the observed mean score for each aspect. Bottom row: adding bag-of-words (BoW) features; we also show results using standard BoW representation (with no topic information). Results for each model show the performance achieved when the inferred topic distributions are added to the BoW representations.

prediction. All FLDA models can predict the user ratings substantially better than LDA, though one difference is that the LDA results improve with more topics, while the opposite happens with FLDA, perhaps because the FLDA priors are enriched with information that only helps the first 3 topics. Additionally, the two 'W' models typically had lower variance than others, perhaps because the word priors lead to more consistency in the inferred parameters. Exact numbers for prediction from the topic output are shown in Table 4.9.

	staff	helpfulness	knowledgeability
Baseline	1.27	1.52	1.36
LDA	1.50	1.60	1.43
B	0.95	0.67	0.72
W	0.91	0.60	0.65
R	0.99	0.69	0.71
WR	0.91	0.59	0.64

Table 4.9: Mean absolute error of rating prediction using FLDA distributions as features with $T_1 = 3$ on the doctor reviews dataset.

4.2.4.2.2 QUALITATIVE ANALYSIS

Our model uncovers several interesting salient patterns, shown in Figure 4.5. Consider the general sentiment terms: *rude* and *asked* are the top two most negative words according to $\boldsymbol{\eta}$, highlighting the importance of communicative/interpersonal skills. Indeed, it would seem that poor communicative skills is generally the most complained about aspect of patient care. Generally positive terms are (unsurprisingly) dominated by superlatives (e.g., *wonderful*). Additionally, we find that the words associated with topics generally match what one would expect: the interpersonal topic includes words like *manner* and *caring*; the technical topic contains words about surgeries and other operations; and the systems topic contains words about the hospital and office, such as *appointment*, *staff*, and *nurse*.

Increasing beyond 3 topics yield more specific words in each topic. The interpersonal topic with T_1=9 included the words *unprofessional*, *arrogant*, *attitude*, *cold*, and *condescending* (negative), along with *compassionate* and *understanding* (positive). When examining the topics beyond the first 3, we find that the model learns clusters of more specific topics such as dentistry and family matters, but some of these topics are noisier and the

positive and negative distributions for the same topic index sometimes do not even correspond to the same theme. The fact that the topics beyond the first 3 are less salient may explain why the rating prediction with FLDA was generally worse for $T_1 > 3$.

It is also interesting to consider how words appear across different topic or sentiment categories. For example, *rude* is obviously associated with negative sentiment, and it shows up in both the system and interpersonal topics. We find that it has higher probability in the system topics, thus it appears patients more often describe staff as rude than doctors. Different forms of *listen* appear in the different sentiment categories of the interpersonal topic; *listen* (as in "s/he doesn't listen") appears in negative, while *listens* (as in "s/he listens") appears in positive.

Certain issues appear to be associated with specific polar sentiment. For example, *medication* and *prescription* is mentioned more in negative contexts—patients remark when they get a wrong prescription, but a correct prescription is unremarkable. Bedside manners are primarily mentioned in positive contexts. Systems issues related to appointments and wait times are primarily mentioned in negative contexts; this agrees with López et al.'s remark that many patients were concerned with wait times (2012).

We also observed sentiment-specific differences in the language used by patients to reference their doctor: the word *dr* has a high prior for the positive sentiment, and upon inspection we noticed that users writing positive reviews were more likely to mention the doctor by name and title ("Dr. X"), while addressing the doctor by name only ("X") or no name ("s/he," "the doctor") was more common with negative reviews. More gener-

NEGATIVE	POSITIVE	INTERPERSONAL	TECHNICAL	SYSTEMS
η (prior over ω)				
rude	thorough	insurance	thorough	office
asked	great	visit	gave	receptionist
pain	best	felt	prescription	staff
told	dr	years	specialist	appointment
room	ive	listen	pain	friendly
dont	caring	caring	knowledgeable	waiting
didnt	friendly	doctor	primary	make
receptionist	hes	condition	question	minutes
ω (prior over ϕ)				
told	recommend	patients	surgery	staff
said	wonderful	care	pain	time
pain	highly	manner	went	office
didnt	knowledgeable	family	hospital	questions
wrong	professional	help	told	wait
dont	kind	caring	months	helpful
tell	great	treatment	old	nice
test	dr	patient	husband	feel
left	best	bedside	said	great

ϕ (word distribution for pair)					
INTERPERSONAL		TECHNICAL		SYSTEMS	
NEGATIVE	POSITIVE	NEGATIVE	POSITIVE	NEGATIVE	POSITIVE
doctor	dr	pain	dr	office	dr
care	doctor	told	surgery	time	time
medical	best	went	first	doctor	staff
patients	years	said	son	appointment	great
doesnt	caring	dr	life	rude	helpful
help	care	surgery	surgeon	staff	feel
know	patients	later	daughter	room	questions
patient	patient	didnt	recommend	didnt	office
dont	recommend	months	baby	visit	really
treatment	family	years	thank	wait	friendly
problem	excellent	hospital	pregnancy	insurance	doctor
tests	knowledgeable	left	husband	minutes	nice
doctors	highly	weeks	old	dr	love
listen	doctors	needed	child	waiting	going
medication	manner	days	delivered	called	recommend
condition	kind	work	results	dont	wonderful
people	bedside	blood	job	first	comfortable

Figure 4.5: The highest-weight words for the hyperparameters η and ω (left), and the highest probability words for each (topic, sentiment) pair (right) for the full model with $T_1 = 3$ topics on the doctor reviews dataset.

Supervision type	Initialization	Perplexity
Unsupervised ($\eta = 0$)	Random	1397.00 ($\pm$ 10.15)
Seed words ($\eta = 1$)	Sampled	1386.90 ($\pm$ 6.90)
Seed words ($\eta = 1$)	Max	1379.51 ($\pm$ 7.82)
Seed words ($\eta = 10$)	Sampled	1452.94 ($\pm$ 10.26)
Seed words ($\eta = 10$)	Max	1452.78 ($\pm$ 8.72)
All words (η learned)	Sampled	1394.66 ($\pm$ 11.53)
All words (η learned)	Max	1379.19 ($\pm$ 6.4)

Table 4.10: The FLDA document completion perplexity ($\pm$ standard deviation) when using different levels of word-level supervision on the doctor reviews data.

ally, we noticed that specific mentions of people appear in positive contexts. For example, the technical/operations topic includes many words describing family members (*husband*, *daughter*). In the systems issues topic, *staff* has a higher probability in the positive distribution than *office* (a more abstract institution), whereas this pattern is reversed in the negative distribution.

4.2.4.2.3 How much supervision is needed?

While we have shown that adding various forms of available supervision can improve performance, there is a cost in added model complexity and development time. We now briefly investigate what can be learned without the incorporation of the extensive prior knowledge considered above.

We first experimented with using the standard FLDA model with no supervision and analyzed what it learns, using $\vec{T} = (2, 3)$ to correspond to the structure used above with two sentiment components and three topics. We then experimented with a lighter-weight form of supervision using a small set of *seed words*, in which we used $\boldsymbol{\eta}$ priors over $\boldsymbol{\omega}$ component

NEGATIVE	POSITIVE	INTERPERSONAL	TECHNICAL	SYSTEMS
Top component words with no supervision				
told	helpful	excellent	hospital	time
went	knowledgeable	highly	life	wait
said	caring	recommend	old	patients
weeks	family	wonderful	husband	questions
months	treatment	caring	second	office
second	best	great	high	appointment
didnt	excellent	best	first	staff
left	wonderful	professional	saw	doesnt
first	recommend	knowledgeable	home	dont
saw	help	helpful	heart	patient
hours	great	pain	helped	phone
later	knowledge	job	knew	waiting
hospital	kind	family	weeks	calls
old	years	kind	man	ask
wanted	patient	recommended	months	times
NEGATIVE	POSITIVE	INTERPERSONAL	TECHNICAL	SYSTEMS
Seed words used for supervision				
horrible	best	manner(s)	knowledge(able)	office
worst	great	bedside	medication(s)	staff
wrong	wonderful	listen(s/ed)	diagnos(e/ed/is)	appointment(s)
rude	excellent	explain(s/ed)	treatment(s)	phone
unprofessional	amazing	caring	test(s)	insurance
Top component words with seed word supervision (excluding seed words)				
told	recommend	questions	life	rude
didnt	professional	patients	surgery	experience
said	highly	doctor	care	work
wait	family	helpful	problems	results
tell	dr	feel	condition	pay
times	extremely	time	problem	left
know	helpful	really	treated	service
dont	experience	kind	able	called
problem	years	patient	physician	went
help	friendly	hes	years	new
time	recommended	cares	husband	money
asked	kind	concerns	hospital	said
day	care	makes	medicine	told
spent	area	recommend	referred	asked
called	surgery	talk	told	times

Figure 4.6: The highest weight component ω values for sentiment and topic values learned by FLDA on the doctor reviews corpus, learned with no supervision and with seed word supervision, with η set to 10 and -10 for seed word priors.

weights, as above, except rather than learning $\boldsymbol{\eta}$ from labeled data, we hand-defined η_{iv} values for a small number of words, leaving the values at 0 for all other words. This form of supervision would be much easier for a practitioner to invoke, as it just requires the selection of a handful of words that reflect a practitioner's prior belief about what concepts should be learned.

We selected seeds to correspond to the two sentiment values and three aspect values used above, to guide the model toward the same concepts. We selected five seed words (plus additional forms of the words) for each component by examining the top 50 ω and ϕ values from the unsupervised model and choosing words thought to be representative of each concept. The chosen seed words are shown in Figure 4.6 along with example output.

We set the η_{iv} prior mean for each seed word to λ in the target component and to $-\lambda$ for the other components in the same factor, comparing values of $\lambda = 1$ and $\lambda = 10$.

We compared two different methods of initializing the Gibbs sampler assignments. In one version, the sampler assignments are initialized proportionally to the $\tilde{\phi}$ prior (referred to as "Sampled" initialization). In the other version, the assignments are initialized to the tuple with the highest score under the prior (referred to as "Max" initialization). Both initialization methods have the effect that seed words are more likely to be initialized to tuples with the desired components.

Figure 4.6 shows the top words (according to the highest ω weights) for each component for both the unsupervised model and the seed word model with $\lambda = 10$, which we qualitatively observed to be a closer match to the target concepts than $\lambda = 1$. With the

seed word model, we found that the "Max" initialization approach resulted in much better components, so this version is shown in the figure.

We observe that the unsupervised model actually learns concepts that are quite close to the full model in Figure 4.5. The top component words for POSITIVE, NEGATIVE, TECHNICAL, and SYSTEMS are all quite close to the full model. The unsupervised model does not appear to learn a meaningful interpersonal manners concept, and the INTERPERSONAL component is not very distinguishable from the POSITIVE component in the other factor, so there is not a clear factorization in this model.

Generally, the top words in the seed word model are the seed words themselves, as might be expected, so we excluded these from the figure. In all cases, the concepts line up well with the seed words, and the components are similar to the full model in Figure 4.5. It thus appears that the simple seed word method is sufficient for learning the desired concepts. However, we found that initialization was very important for incorporating seed words. When using the "Sampled" initialization instead of "Max", the components look similar to the unsupervised model, and it is not clear that this helps qualitatively.

We also measured the document completion perplexity of these different models, averaging perplexity from 10 randomly initialized Gibbs sampling trials, shown in Table 4.10. The seed word model with $\lambda = 1$ does significantly better than the unsupervised model, with $p = .018$ and $p = .0004$ under a two-sided t-test for the "Sampled" and "Max" variants, respectively. The seed word model with $\lambda = 10$ actually has substantially worse perplexity, despite much better interpretability than the $\lambda = 1$ version qualitatively. This

may be because a prior weight of 10 is much higher than would be naturally learned, which worsens perplexity. For comparison, we also measured perplexity of the model using the full $\boldsymbol{\eta}$ supervision for all words (the 'W' model above), which has comparable perplexity to the $\lambda = 0$ seed word model (they are not significantly different).

The "Max" initialization results in better perplexity than the "Sampled" initialization for both the seed word model with $\lambda = 1$ ($p = .038$) and the full model ($p = .002$). This agrees with our qualitative finding that the "Max" initialization works better.

To conclude, we have found qualitatively that interpretable components can be learned with very simple supervision based on seed words, and even reasonable, but imperfect, components can be learned with no supervision. We also found that it is crucial to initialize the Gibbs sampler assignments for seed words to the correct tuples, rather than using random sampling.

Another application of seed word priors is discussed later in Section 6.1.3.1.

4.2.4.3 Healthcare quality prediction

As an experiment to show what can be learned using FLDA, we evaluated whether the distributions inferred by FLDA from online reviews are predictive of real-world information. We explore associations between external U.S. state-level healthcare statistics and the distributions over (topic, sentiment) pairs learned by FLDA. We considered three measurements of healthcare quality taken from the Dartmouth Atlas of Healthcare (Goodman et al., 2011): the percentage of patients who saw their primary care physician (PCP) within

fourteen days of discharge, mortality rates, and mean monetary expenditure.

Similar to the rating prediction task in Section 4.2.4.2, here we used regression models to predict the state-level attributes. To create features from FLDA, we used the average tuple distribution θ_m across all reviews within each U.S. state.

We experimented with two linear regression models. The first uses the average review rating for each state as the only feature. The second model uses the FLDA features (6 total, for each topic-sentiment pair) in addition to the rating feature.

Our goals were (1) to determine if information in the reviews was significantly predictive of the external measurements, and (2) to determine if information from the text (as inferred by FLDA) is more predictive than the user ratings alone. For goal (1), we use a t-test to determine if the regression model significantly predicts the state-level outcomes (that is, the regression coefficients are significantly non-zero). For goal (2), we use likelihood ratio (LR) tests to compare the two models with and without FLDA features. If adding FLDA model output results in statistically significantly better models, it indicates that this model output contains information not readily available from the raw metadata.

First, we considered the percentage of patients who visited a PCP within fourteen days of hospital discharge following an acute event in 2010 (the most recent available data). This is considered a positive measure of adequate healthcare access and coordination of care. We found that the regression model with the full feature set was significantly predictive of this measure ($p = 0.03$). Furthermore, the model with FLDA features explains more of the variance in this outcome than the RateMDs ratings alone (LR test $p = .10$; R-squared of

.13 when using RateMD ratings only and .21 when including model output).

We also considered the overall Medicare state mortality rates from 2007 (the most recent available data). However, neither model was significantly predictive of this measure. This is not surprising: across multiple measures of patient satisfaction, even with rigorous population sampling, there is no consistent association with mortality (Fenton et al., 2012; Schneider et al., 2001; Sequist et al., 2008).

Finally, we considered the cost of care across states, in terms of health care expenditure per capita. We again found that including the topic modeling output explains more variance in the outcome across states than the RateMDs ratings alone (LR test $p = .02$). Including topic modeling output results in an R-squared of .25 with respect to cost while using only the RateMDs ratings results in an R-squared of only .03.

Online doctor review data across states is thus associated with patient likelihood of receiving and attending a post-hospitalization appointment with his or her PCP and with higher cost of care. Moreover, the text of the reviews, modeled as latent topic and sentiment categories, contains information beyond the user ratings that have been considered in previous studies (Segal et al., 2012). These results demonstrate that meaningful information is contained in online reviews, and that text modeling (via factorial LDA) is beneficial over simply using structured data (ratings).

4.2.4.4 Understanding patient perceptions

While the previous experiment showed that online user reviews can be used to measure knowledge captured through other systems (healthcare quality metrics), we now use the output of FLDA to conduct analyses that are not captured by existing databases. In particular, we examine at a high level the *perceptions* of patients, as indicated by the content of their reviews, and visualizing how the review content varies across the United States. Understanding patient perceptions of quality, and understand which aspects of care are important to patients, will play an important role in improving healthcare quality (Sofaer and Firminger, 2005; Greaves et al., 2013).

Figure 4.7 shows the average proportion of review text tokens assigned to each of the three topics according to FLDA, and Figure 4.8 shows the same, but for sentiment instead of aspect. While the previous subsection showed that these proportions are correlated with existing metrics, the proportions are interesting to examine.

A few patterns emerge upon examination of these figures. The proportions tend to exhibit geographic clustering (that is, nearby states often have similar values), which is evidence that these patterns are not random.

Positive sentiment ranges from approximately 54% to 64%, meaning the majority of text is positive in all states. This suggests that the average healthcare experience is satisfying, though not by wide margins. These sentiment values are more balanced than in product reviews, which are known to have a strong positive bias (Hu et al., 2009). States in the Midwest typically have the highest proportion of positive sentiment, while states in the

West and Southwest have high negative sentiment.

The SYSTEMS topic proportions range from approximately 35% to 43%, INTERPERSONAL ranges from 34% to 46%, and TECHNICAL ranges from 17% to 24%. This shows that issues unrelated to the technical competence of the doctor's care—systems issues related to the office and the doctor's manner—are roughly twice as impactful on a patient's perspective on care than technical skill, at least as defined by the amount of review text devoted to these issues. It is likely that these two issues are easier for a patient to perceive and judge than a doctor's competence. There is also geographic variation in the discussion of different topics, with system issues being discussed much more commonly in the South, Southwest, and Mid-Atlantic regions, interpersonal manners discussed more in the Northeast and Pacific Northwest, and technical competence discussed more in the West and Midwest.

These findings suggest that patient perceptions (which our results show are primarily based on issues unrelated to technical quality) and care outcomes (which are defined by concrete metrics, as considered in the previous subsection) are quite different, and thus information found in online reviews, which contains perceptions of quality more than objective quality measures, can complement traditional care metrics.

4.3 Summary

This chapter presented two semi-supervised instantiations of factorial LDA designed for specific health science applications: learning information about drug use from online forums and measuring healthcare quality from online reviews of doctors. We enriched the basic FLDA model with additional priors to guide the model toward specific concepts for the target applications. This included priors over the word distributions (for both applications) and priors over the tuple distributions (for the doctor reviews application), leveraging labeled data and metadata. Experiments with the drug forum data showed that FLDA can extract information that is comparable to information summarized by humans, and experiments with the doctor review data showed that FLDA learns information that significantly predictive of real-world measurements.

Through these two applications, this chapter demonstrated the utility of factorial LDA, while also introducing general methods for incorporating domain knowledge and metadata into the model, which could be applied to new applications in the future.

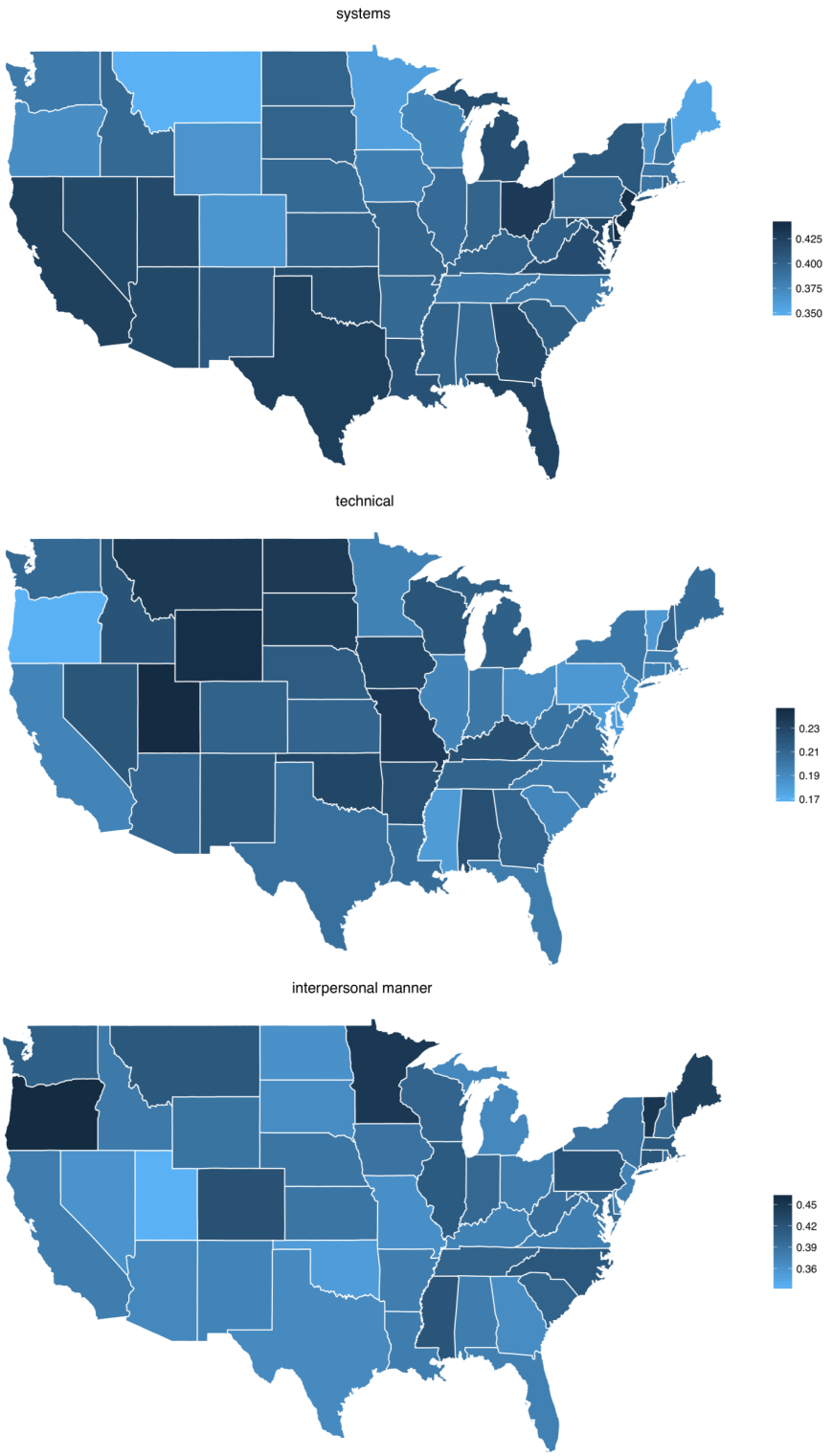


Figure 4.7: The proportion of doctor review text related to three aspects of healthcare across U.S. states, as inferred by Factorial LDA.

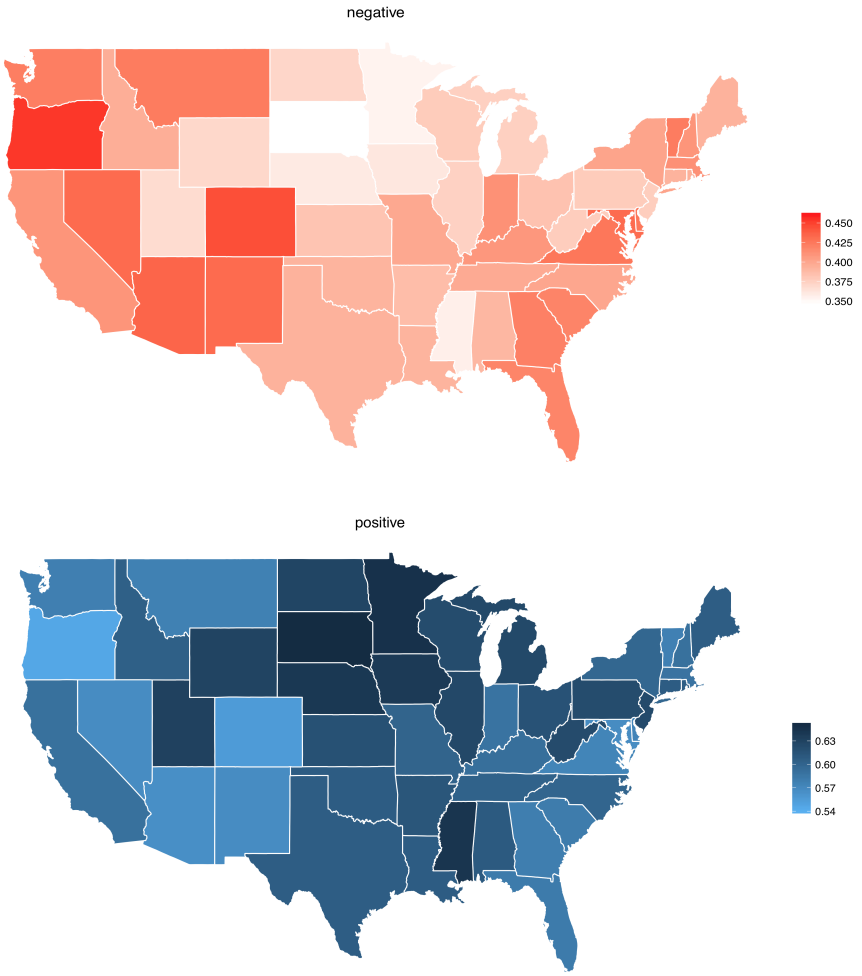


Figure 4.8: The proportion of doctor review text related to positive and negative sentiment across U.S. states, as inferred by Factorial LDA.

Chapter 5

SPRITE: A Family of Topic Models with Structured Priors

The previous two chapters introduced and showcased factorial LDA (FLDA), a probabilistic model that organizes word distributions into multiple dimensions, using structured priors to create a factored structure. As described in Section 2.4, topics can be organized in many different ways, and a factorization is just one type of structure that has been considered for topic models.

This chapter will introduce SPRITE, a family of topic models that provides a flexible framework for encoding priors for how topics should be structured. In addition to factorizations, SPRITE can incorporate many other types of structure that have been considered in previous work, including hierarchies (Blei et al., 2003a; Mimno et al., 2007), correlations between topics (Blei and Lafferty, 2007; Li and McCallum, 2006), preferences over

word choices (Andrzejewski et al., 2009; Paul and Dredze, 2013), and associations between topics and document attributes (Ramage et al., 2009; Mimno and McCallum, 2008). This chapter builds on material from Paul and Dredze (2015).

SPRITE extends the idea of structured priors used by factorial LDA. In FLDA, weight vectors called *components* were combined to form priors for tuple-specific word distributions and document-specific tuple distributions. In SPRITE, this idea is generalized to allow component weight vectors to be combined in arbitrary ways to create different structures. We call this family of models **SPRITE**: **S**tructured-**PRI**or **T**opic mod**E**ls. We will show that SPRITE is general enough to capture the behavior of a diverse variety of topic models, including FLDA.

In Section 5.1, we define the form of the model, and Subsections 5.1.1–5.1.4 will describe the constraints and priors that can be placed on the component hyperparameters to induce different structures. Section 5.2 will show how various existing topic models are special cases of SPRITE, focusing in particular on factorial LDA in Subsection 5.2.1, for which the SPRITE representation has advantages over the original formulation. Section 5.3 describes posterior inference and parameter optimization. Section 5.4 presents experiments that compare various structures that can be created with SPRITE. The next chapter (Chapter 6) will provide further evaluation of SPRITE through application-specific models.

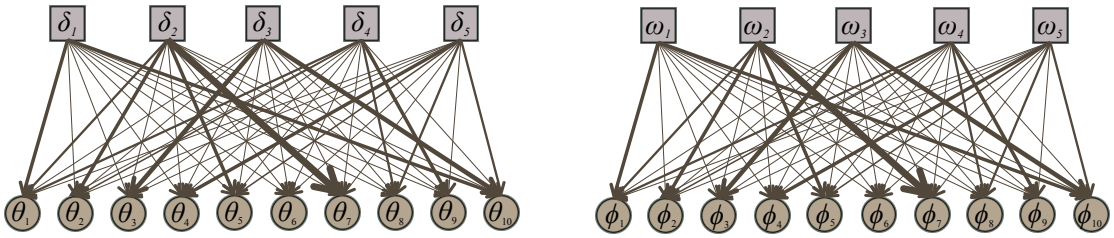


Figure 5.1: An illustration of the relationship between document components and document parameters (left) and between topic components and topic parameters (right) in SPRITE. Edge weights are used to illustrate the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

5.1 Model Definition

Like FLDA, SPRITE is based on LDA. SPRITE adds structure to the generation of the Dirichlet parameters. The priors for these parameters are modeled as log-linear combinations of underlying *components*. Components are real-valued vectors of length equal to the vocabulary size V (for priors over word distributions) or length equal to the number of topics T (for priors over topic distributions).

For example, we might assume that topics about sports like baseball and football share a common prior with general words about sports, defined by a sports-themed component. A fine-grained topic about steroid use in sports might be created by combining components about broader topics like sports, medicine, and crime. By modeling the priors as combinations of components that are shared across all topics, we can learn interesting connections between topics, where components provide an additional latent layer of structure for understanding corpus content.

As we'll show in the next section, by imposing certain requirements on which components feed into which topics (or documents), we can induce a variety of model structures. For example, if we want to model a topic hierarchy, we require that each topic depend on exactly one parent component. If we want to jointly model topic and ideology in a corpus of political documents (as we do in Chapter 6), we make topic priors a combination of one component from each of two groups: a topical component and an ideological component, resulting in ideology-specific topics like "conservative economics".

Components are used to construct priors as follows. For the topic-specific word distributions $\boldsymbol{\phi}$, there are $C^{(\phi)}$ *topic components*. The tth topic's prior over ϕ_t is a weighted combination (with coefficient vector β_t) of the $C^{(\phi)}$ components, where the cth topic component is denoted ω_c. For the document-specific topic distributions $\boldsymbol{\theta}$, there are $C^{(\theta)}$ *document components*. The mth document's prior over θ_m is a weighted combination (with coefficient vector α_m) of the $C^{(\theta)}$ components, where the cth document component is denoted δ_c. These variables are illustrated in Figure 5.1.

Once conditioned on these priors, the model is otherwise identical to LDA. The graphical model is shown in Figure 5.2. The generative story is:

1. Generate hyperparameters: $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\delta}$, $\boldsymbol{\omega}$ (see Section 5.1.1)

2. For each document m, generate parameters:

   (a) $\tilde{\theta}_{mt} = \exp(\sum_{c=1}^{C^{(\theta)}} \alpha_{mc}\, \delta_{ct})$, $1 \leq t \leq T$

   (b) $\theta_m \sim \text{Dirichlet}(\tilde{\theta}_m)$

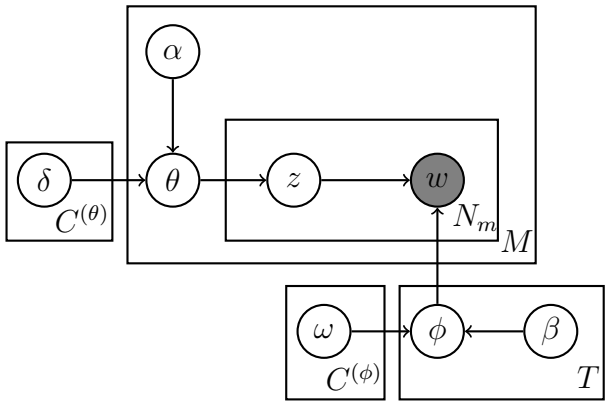


Figure 5.2: The graphical model plate diagram for SPRITE.

3. For each topic t, generate parameters:

   (a) $\tilde{\phi}_{tv} = \exp(\sum_{c=1}^{C^{(\phi)}} \beta_{tc}\, \omega_{cv})$, $1 \leq v \leq V$

   (b) $\phi_t \sim \text{Dirichlet}(\tilde{\phi}_t)$

4. For each token n in each document m, generate data:

   (a) Topic (unobserved): $z_{mn} \sim \theta_m$

   (b) Word (observed): $w_{mn} \sim \phi_{z_{mn}}$

Steps 2a and 3a are closely related to steps 2a and 3a of the FLDA generative story in Section 3.1.

To illustrate the role that components can play, consider an example in which we are modeling research topics in a corpus of NLP abstracts (as we do in Section 5.4). Consider three speech-related topics: signal processing, automatic speech recognition, and dialog systems. If conceptualized as a hierarchy, these topics might belong to a higher level category of spoken language processing. SPRITE allows the relationship between these three

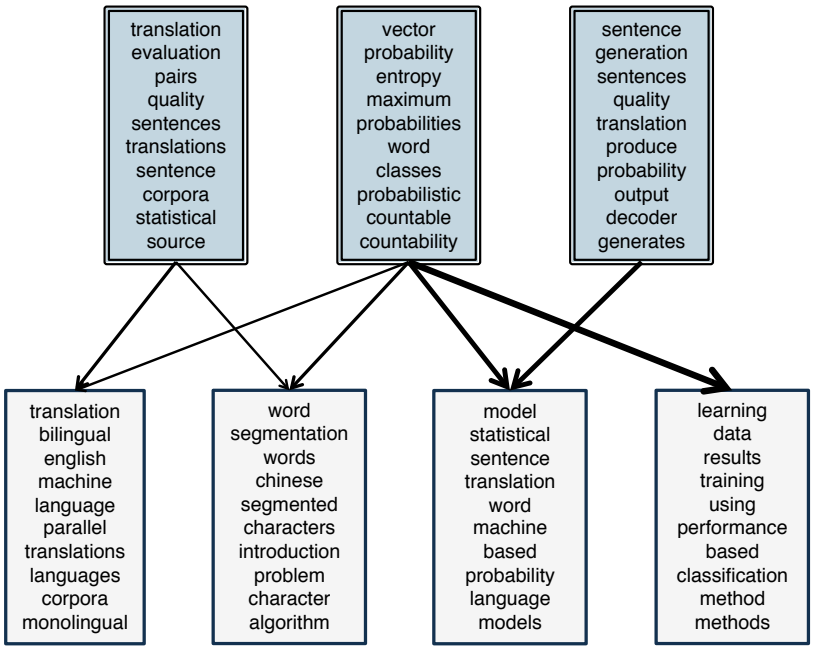


Figure 5.3: Examples of topic components and topics in SPRITE, visualized as the ten highest-weight words in the components and the ten most probable words in the topics. These components were learned on a dataset of computational linguistics abstracts, using the experimental setup described in Section 5.4.1 (using the DAG structure) with $T = 50$ topics and $C = 10$ components. For each example topic, an edge is present for every component such that the coefficient β_{tc} is at least one standard deviation above the mean value for that topic. The edge weights in the figure increase with the value of β_{tc}. This figure illustrates how high-level concepts encoded by components can influence topics. For example, all of the topics shown here draw from the component describing machine learning and probabilistic modeling (middle component), and the left two topics additionally draw from the component describing machine translation (left component).

topics to be defined in two ways. First, we can model that these topics will all have words in common. This is handled by the topic components—these three topics could all draw from a common "spoken language" topic component, with high-weight words such as *speech* and *spoken*, which informs the prior of all three topics. Second, we can model that these topics are likely to occur together in documents. For example, articles about dialog systems are likely to discuss automatic speech recognition as a subroutine. This is handled by

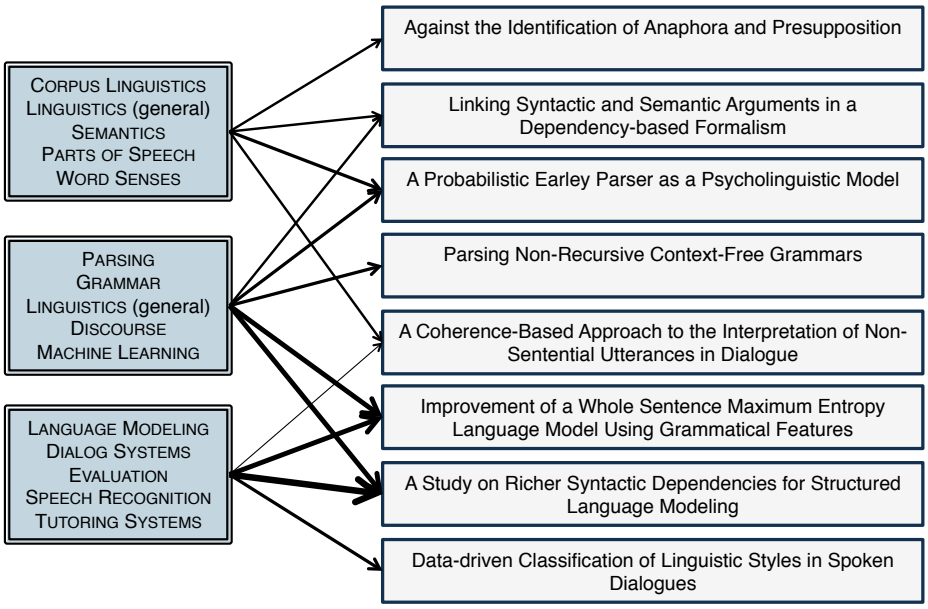
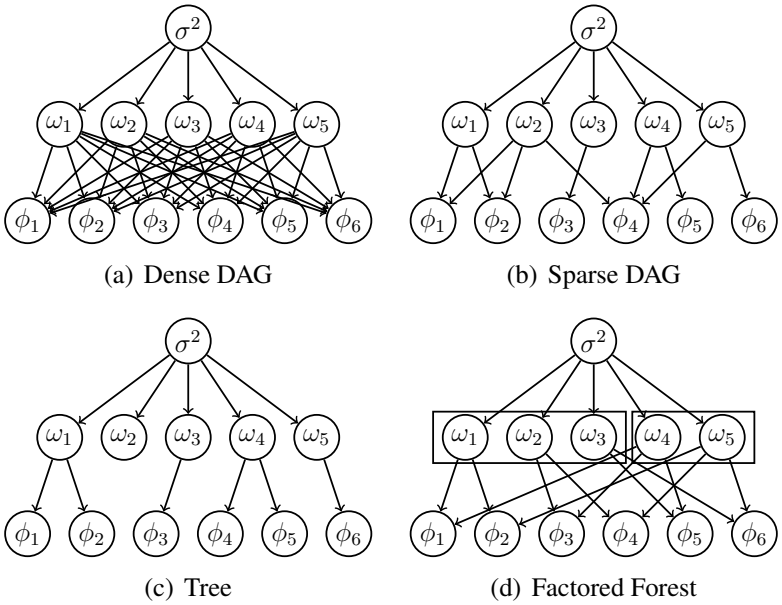


Figure 5.4: Examples of SPRITE document components and documents, visualized as the five highest-weight topics in the components and the titles of a sample of computational linguistics papers in the dataset, using the same experimental setup as in Figure 5.3. The smallcaps names of topics in the components are manually assigned upon inspection of the most probable words in the topics. For each example document, an edge is present for every component such that the coefficient α_{mc} is at least one standard deviation above the mean value for that topic. The edge weights in the figure increase with the value of α_{mc}. This figure illustrates how related topics are grouped into components, and how the components influence the choice of topics in documents. For example, the prior for the document "A Study on Richer Syntactic Dependencies for Structured Language Modeling" draws from a component with high weight for the LANGUAGE MODELING topic and a component with high weight for the PARSING and GRAMMAR topics.

the document components—there could be a "spoken language" document component that gives high weight to all three topics, so that if a document draw its prior from this component, then it is more likely to give probability to these topics together. More examples of these behaviors can be seen in Figures 5.3–5.4, with examples of topic components and document components learned from real data.

The next subsections will describe how particular priors over the coefficients can induce



(a) Dense DAG (b) Sparse DAG

(c) Tree (d) Factored Forest

Figure 5.5: Example graph structures describing possible relations between topic components (middle layer) and topics (bottom layer), as described in Section 5.1.1. Edges correspond to non-zero values for β (the component coefficients), as defined in step 2a of the generative story in Section 5.1. The root node is a shared prior over the component values, with other possibilities discussed in Section 5.1.4. The model structure for document components is similar, with δ instead of ω, α instead of β, and θ instead of ϕ.

various structures such as hierarchies and factorizations, and components and coefficients can also be provided as input to incorporate supervision and prior knowledge. The general prior structure used in SPRITE can be used to represent many types of topic models, outlined in Section 5.2.

5.1.1 Structuring the Components

By changing the the hyperparameters—the component coefficients ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) and the component values ($\boldsymbol{\delta}$ and $\boldsymbol{\omega}$)—we can obtain a diverse range of model structures and behav-

iors. This subsection discusses various graph structures that can describe the relationships between topic components and topics, and between document components and documents, illustrated in Figure 5.5.

5.1.1.1 Directed acyclic graph (DAG)

The general SPRITE model can be thought of as a dense directed acyclic graph (DAG), where every document or topic is connected to every component with some weight α or β. When many of the α or β coefficients are zero, the DAG becomes sparse. A sparse DAG has an intuitive interpretation: each document or topic depends on some subset of components.

By default, we will assume the prior over coefficients is a 0-mean normal distribution, which encourages the weights to be small, but not necessarily zero (*weakly* sparse). We note that to induce a sparse graph, one could use a 0-mean Laplace distribution as the prior over the coefficients, which prefers parameters such that some components are zero (*strongly* sparse), as used for the weights in SAGE (Eisenstein et al., 2011) (described in Section 2.3.3.2).

An alternative approach to sparsity is to let $\alpha, \beta \in \{0, 1\}$, which yields unweighted (binary) component selection. This is the structure used in the shared components topic model (Gormley et al., 2010) (described in Section 2.4.1.2). As in the sparse variant of FLDA in Section 3.1.2, we can relax the binary constraints to be real-valued in $(0, 1)$, using a "U-shaped" Beta(ρ<1) distribution as the prior to encourage sparsity.

5.1.1.2 Tree

When each document or topic has exactly one parent component—that is, one nonzero coefficient—we obtain a two-level tree structure, shown in Figure 5.5(c). This structure naturally arises in topic hierarchies, for example, where fine-grained topics are children of coarse-grained topics.

To create an unweighted tree, we require $\alpha_{mc} \in \{0, 1\}$ and $\sum_c \alpha_{mc} = 1$ for each document m. Similarly, $\beta_{tc} \in \{0, 1\}$ and $\sum_c \beta_{tc} = 1$ for each topic t. In this setting, α_m and β_t are indicator vectors which select a single component.

In this thesis, rather than strictly requiring α_m and β_t to be binary-valued indicator vectors, we create a relaxation that allows for easier parameter estimation. We let α_m and β_t be real-valued vectors in a simplex—that is, $\alpha_{mc} \in (0, 1)$, $\sum_c \alpha_{mc} = 1$ and $\beta_{tc} \in (0, 1)$, $\sum_c \beta_{tc} = 1$—but we place a prior over their values to encourage sparsity, favoring vectors with a single component near 1 and others near 0. This is achieved using a Dirichlet(ρ<1) distribution as the prior over each α_m and β_t, which has higher density near the boundaries of the simplex, as explained in Section 2.1.2. This generalizes the constraint relaxation technique used for sparse FLDA in Chapter 3, which approximated binary variables with real-valued variables with a sparse Beta prior. The Dirichlet distribution is the multivariate extension of the Beta distribution.

To create a weighted tree, the coefficients could be defined as a product of two variables: an "integer-like" indicator vector with sparse Dirichlet prior as suggested above, combined with a real-valued weight (e.g., with a normal prior). We take this approach in

our experiments (Section 5.4).

5.1.1.3 Factored forest

By using structured sparsity over the DAG, we can obtain a structure where components are grouped into K groups (or "factors", as in FLDA), and each document or topic has one parent from each group. Figure 5.5(d) illustrates this: the left three components belong to one group, the right two belong to another, and each bottom node has exactly one parent from each. This is a DAG that we call a "factored forest" because the subgraphs associated with each group in isolation are trees. This structure arises in "multi-dimensional" models (Section 2.4.2) like SAGE (Eisenstein et al., 2011) and factorial LDA, which allow tokens to be associated with multiple variables. We will make the connection to factorial LDA explicit in Section 5.2.1.

The "exactly one parent" indicator constraint is the same as in the tree structure but enforces a tree only within each group. This can therefore be (softly) modeled using a sparse Dirichlet prior as described above in the previous subsection. In this case, the subsets of components belonging to each factor have separate sparse Dirichlet priors. Using the example from Figure 5.5(d), the first three component indicators would come from one Dirichlet, while the latter two component indicators would come from a second.

5.1.2 Tying Topic and Document Components

A desirable property for many situations is for the topic and document components to correspond to each other, with the number of topic components equal to the number of document components ($C^{(\phi)} = C^{(\theta)}$). For example, if we think of the components as coarse-grained topics in a hierarchy, then the coefficients $\boldsymbol{\beta}$ enforce that topic-word distributions share a prior defined by their parent $\boldsymbol{\omega}$ components, while the coefficients $\boldsymbol{\alpha}$ represent a document's proportions of coarse-grained topics, which affects the document's prior over child topics through the $\boldsymbol{\delta}$ vectors. Consider the example with spoken language topics in Section 5.1: the three topics, signal processing, speech recognition, and dialog systems, are *a priori* likely both to share the same words and to occur together in documents. By tying these together, we ensure that the patterns are consistent across the two types of components, and the patterns from both can reinforce each other during inference. A concrete example of this will be seen in Section 6.1.

To do this, we could assume $\delta_{ct} = \beta_{tc}$. For example, we will show in Section 5.2.1 how factorial LDA is a case of SPRITE where the document components are the transpose of the topic components, $\delta_{ct} = \beta_{tc}$. A less restrictive assumption is that δ_{ct} and β_{tc} are correlated but not identical.[1] The approach we propose is to define each δ_c and β_t as a product of two variables (suggested in Section 5.1.1.2): a "binary" mask variable (with sparse Dirichlet prior), which we let be identical for both δ_c and β_t, and a real-valued positive weight which

[1] Perhaps an obvious option is to use a multivariate normal prior with positive covariance between these variables. However, a normal prior precludes the use of a sparse Dirichlet (which cannot model covariance) over $\boldsymbol{\delta}$ and $\boldsymbol{\beta}$, which is desired for most of the structures we have described.

is uncorrelated.

5.1.3 Positivity Constraints

It will often be desirable to constrain the coefficients ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) or components ($\boldsymbol{\delta}$ and $\boldsymbol{\omega}$) to positive values. One reason is that without constraints on the signs of the coefficients of components, the parameters of the priors are not identifiable,[2] since flipping the signs results in the same likelihood, i.e., $\alpha_m \cdot \delta_c = -\alpha_m \cdot -\delta_c$. Another reason is simply that positive values can be easier to interpret. For example, if we wish to interpret a topic's β_t coefficients as "selecting" a subset of components to draw from, as in the sparse DAG and tree structures described in the previous subsection, then the coefficients must all have the same sign. Similar reasons are used to motivate non-negativity constraints in non-negative matrix factorization (NMF) (Lee and Seung, 1997, 2001), a type of non-probabilistic topic model (Stevens et al., 2012).

If such a constraint is applied to a parameter x, we recommend re-parameterizing x as $f(\hat{x})$ in which $f(\hat{x})$ is a function whose domain is unconstrained and whose codomain is positive, such as $\hat{x}^2$ or $|\hat{x}|$. This leads to easier parameter estimation, allowing us to optimize $\hat{x} \in \mathbb{R}$ rather than $x \in \mathbb{R}_{>0}$. In this thesis, we let $x = \exp(\hat{x})$. This technique was also used in Section 4.2.3.2.

There are also cases where positivity constraints are not desired, however. For example,

[2] As mentioned in Footnote 6 of Chapter 2, topic models are already unidentifiable, so certainly models without identifiability can still be used. However, when parts of a model are not identifiable, then it is harder to compare different learned parameterizations (e.g., parameters from different Gibbs sampling runs).

in Chapter 6 we will see examples of SPRITE models that use components to create associations between topics and different author perspectives (e.g., political ideology or review sentiment), where "perspective" exists on a spectrum from positive to negative. In this case, we achieve better interpretability by allowing the parameters associated with positive perspective (e.g., positive sentiment in a review) to have the opposite sign of negative perspective. Note that these particular models do not lose identifiability (as suggested above) because α_m is fixed as input (as described in Section 5.2.2.4, on incorporating document attributes), so the sign is not arbitrary.

5.1.4 Deep Components

As for priors over the component values $\boldsymbol{\delta}$ and $\boldsymbol{\omega}$, we assume by default that they are generated from 0-mean normal distributions. While not experimented with in this thesis, it is also possible to allow the components themselves to have rich priors which are functions of higher-level components. Rather than assuming a mean of zero, the mean could be a weighted combination of higher-level weight vectors.

A simple instance of this idea was used in Chapter 4, in which the means of the normal priors over $\boldsymbol{\omega}$ in factorial LDA were set to values other than zero based on labeled data, used to guide the parameters to desired concepts (Section 4.1.3).

In general, SPRITE could be extended to have several levels of DAG-structured or tree-structured priors, similar to how the two-level form of Pachinko allocation (Li and McCallum, 2006) (Section 2.4.1.1) can be extended to deeper structures. At higher levels,

components would be used to generate means of normal priors (over the lower components) rather than generating Dirichlet parameters. Using components to generate normal means is also discussed in Section 5.1.7, when discussing the use of structured priors for logistic normal-based topic models rather than Dirichlet-based topic models.

5.1.5 Prior Variance

It is also useful to consider the effect of the variance of the Dirichlet priors. With lower variance (higher precision), the posterior will have higher density around the mean of the prior. As the posterior tends toward the mean of the prior, it is as if we had directly defined the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ as log-linear functions of the components, rather than taking a hierarchical approach of generating these parameters from priors defined by the log-linear functions. SPRITE's hierarchical Bayesian framework allows us to learn parameters that deviate from the prior, while the Dirichlet precision can be increased to create behavior that is closer to log-linear modeling. This tradeoff is discussed in comparison to product-of-experts models in Section 3.1.3.1.

5.1.6 Bias Components

Lastly, we note that it is useful to include a document component that is shared across all documents and a topic component that is shared across all topics. We refer to such components as *bias* or *background* vectors. This can be achieved by simply fixing the

coefficients α and β to 1 across all documents and topics for the bias components. Bias components have the effect that certain words are *a priori* more likely in all topics, and certain topics are *a priori* more likely in all documents.

Bias components thus act as "overall" values that influence the priors over topic and word distributions, which are then adjusted by the document-specific and topic-specific weights. This can improve the interpretability of the specific component values, which can be interpreted as deviations from the bias values. All experiments in this thesis include bias components.

5.1.7 Alternative Base Models

SPRITE assumes LDA as the model upon which the structured priors are built. However, the idea of structured priors—priors over parameters that are themselves functions of higher-level parameters—could be applied to other topic models as well. For instance, the correlated topic model (Blei and Lafferty, 2007) uses logistic normal priors (Section 2.3.2) rather than Dirichlet priors, in which the log of the topic proportions in each document have a normal prior. One could combine SPRITE-style functions with logistic normal priors, in which the means of the normal priors are linear functions of components, similar to how Dirichlet parameters can be functions of components. This approach is used in structural topic models (Roberts et al., 2013) (Section 2.3.3.3), in which the means of logistic normal priors are linear functions of additional parameters.

One could similarly use logistic normal priors for the word distributions in addition to

Model	Sec.	Document priors	Topic priors
LDA	5.2.2.1	Single component	Single component
SCTM	5.2.2.2	Single component	Sparse binary β
SAGE	5.2.2.2	Single component	Sparse ω
FLDA	5.2.1	Binary δ is transpose of β	Factored binary β
PAM	5.2.2.3	α are supertopic weights	Single component
DMR	5.2.2.4	α are feature values	Single component

Table 5.1: Topic models with Dirichlet priors that are generalized by SPRITE. The description of each model can be found in the noted section number. In some cases, the model is not equivalent to SPRITE, but captures similar behavior. PAM is not equivalent, but captures very similar behavior. The described component formulations of SCTM and SAGE are equivalent to SPRITE, but these differ from SPRITE in that the components directly define the parameters, rather than priors over the parameters.

the topic distributions, although this is not often done in topic modeling.

5.2 Special Cases and Related Models

We now describe several existing Dirichlet prior topic models and show how they are special cases of or similar to SPRITE. Table 5.1 summarizes these models and their relationship to SPRITE. In almost every case, we also describe how the SPRITE representation of the model offers improvements over the original model or can lead to novel extensions.

We will first focus on factorial LDA, describing how it is generalized by SPRITE and how the SPRITE formulation can address sparsity in a stronger way than the sparse variant proposed in Section 3.1.2 of Chapter 3. We will then describe how various other topic models are related to SPRITE, focusing on the structured topic models described early in this thesis, in Sections 2.3.3 and 2.4 of Chapter 2.

5.2.1 Factorial LDA as SPRITE

SPRITE is a generalization of factorial LDA (FLDA), introduced in Chapter 3. Recall that under FLDA, "topics" are actually tuples of K variables, such as aspect and sentiment in online reviews. Each document distribution θ_m is a distribution over tuples, and each tuple $\vec{t}$ has a word distribution $\phi_{\vec{t}}$. FLDA uses a similar log-linear parameterization of the Dirichlet priors as SPRITE, parameterized by $\tilde{\phi}_{\vec{t}}$ for each tuple and $\tilde{\theta}_m$ for each document.

We now show how to encode FLDA using SPRITE. A "topic" under SPRITE corresponds to a tuple under FLDA, and we will use the tuple notation in the equations below. The number of "topics" in SPRITE is then the number of possible tuples, $T = \prod_k T_k$, while the number of components is the total number of components from the K factors, $C = C^{(\phi)} = C^{(\theta)} = \sum_k T_k$. For now, we are ignoring the bias and background parameters under FLDA and focusing on the factor-dependent parameters.

The Dirichlet parameters for word distributions under FLDA are defined as:

$$\begin{aligned} \tilde{\phi}_{\vec{t}v} &= \exp(\sum_{k=1}^{K} \omega^{(k)}_{t_k v}) \\ &= \exp(\sum_{k=1}^{K} \sum_{c=1}^{T_k} \beta^{(k)}_{\vec{t}c} \omega^{(k)}_{cv}) \\ \text{where } \beta^{(k)}_{\vec{t}c} &= \begin{cases} 1 & \text{if } c = t_k \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{5.1}$$

We can see that the sum over one component from each of the K factors, as defined by FLDA, can be rewritten as a sum over all components in the model, multiplied by $\boldsymbol{\beta}$ coef-

ficients, as defined by SPRITE. With FLDA, the values of the $\boldsymbol{\beta}$ components are implicitly binary indicators, defined by the multi-dimensional structure. Within each kth factor, the t_kth component of $\beta_{\vec{t}c}^{(k)}$ will be 1 and all others will be 0. This corresponds to an unweighted factored forest structure, as described in Section 5.1.1.3, where the $\boldsymbol{\beta}$ values have only one nonzero value for the components within each factor.

In the notation above, we treat each $\beta_{\vec{t}}$ as a multi-dimensional array, using the superscripts (k) show which portion of the $\beta_{\vec{t}}$ coefficients correspond to which factor. This is just a notational difference between FLDA and SPRITE, but it is also possible to "flatten" the multi-dimensional array into a single vector to create notation that matches SPRITE.

The Dirichlet parameters for each document's tuple distribution can similarly be defined in terms of SPRITE:

$$\begin{aligned} \tilde{\theta}_{m\vec{t}} &= \exp(\sum_{k=1}^{K} \alpha_{mt_k}^{(k)}) \\ &= \exp(\sum_{k=1}^{K} \sum_{c=1}^{T_k} \alpha_{mt_k}^{(k)} \delta_{c\vec{t}}^{(k)}) \\ \text{where } \delta_{c\vec{t}}^{(k)} &= \begin{cases} 1 & \text{if } c = t_k \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{5.2}$$

It is clear from the definitions in Eqs. 5.1–5.2 that $\beta_{\vec{t}c}^{(k)} = \delta_{c\vec{t}}^{(k)}$. This is an instance where the topic and document components are tied together, as described in Section 5.1.2.

Note that the definition of FLDA (Section 3.1) includes bias and background terms for both $\tilde{\phi}$ and $\tilde{\theta}$. Each bias scalar and background vector could be reparameterized as one

vector (e.g., $\omega_{0v} = \omega^{(B)} + \omega_v^{(0)}$), which could then be incorporated as a bias component with SPRITE (Section 5.1.6).

5.2.1.1 Relaxing the factored structure

The SPRITE representation creates possibilities for improving the original FLDA model. FLDA assumes a one-to-one mapping between word distributions and tuples and assumes that the entire Cartesian product of the different factors is represented in the model (e.g., $\phi_{\vec{t}}$ parameters for every possible tuple). With SPRITE, rather than fixing the $\boldsymbol{\beta}$ indicators to encode the Cartesian product exactly, it is possible to *learn* values for $\boldsymbol{\beta}$ that are more flexible, while still respecting the factored forest constraints. If we do this, we allow for the possibility of having multiple word distributions correspond to the same tuple, as well as the possibility of leaving some tuples without corresponding word distributions. We will see an example of this approach in Section 5.4.

The ability to learn word distributions for only a subset of possible tuples is important. With SPRITE, it is possible to set the number of topics T to a value less than $\prod_k T_k$, avoiding the overparameterization that occurs when the Cartesian product is very large. Chapter 4 dealt with this issue by introducing a sparse FLDA variant that incorporated a sparsity pattern into the prior, assigning low prior probability to tuples, but the word distributions for those tuples still existed in the model. The FLDA Gibbs sampler still must compute probabilities for the full Cartesian product (since all tuples were still part of the model), so FLDA's sampling complexity does not scale to higher numbers of factors. SPRITE's ap-

proach creates sparsity in a stronger way. The number of word distributions can be smaller than the full Cartesian product, so the sampling complexity can be dramatically reduced.

5.2.2 Other Models

5.2.2.1 Latent Dirichlet allocation

In LDA (Blei et al., 2003b), all $\boldsymbol{\theta}$ vectors are drawn from the same prior, as are all $\boldsymbol{\phi}$ vectors. This is a basic instance of our model with only one component at the topic and document levels, $C^{(\theta)} = C^{(\phi)} = 1$, with coefficients $\alpha = \beta = 1$. In this setting, the hyperparameters $\tilde{\theta}$ and $\tilde{\phi}$ are formulated as $\exp(\alpha)$ and $\exp(\omega)$. Note that if the α and ω hyperparameters have normal priors (which is the default assumption in SPRITE), then this changes the LDA likelihood function to add an additional layer of priors.

5.2.2.2 Shared components and sparse additive models

SPRITE is closely related to shared components topic models (SCTM) (Gormley et al., 2010), described in Section 2.4.1.2. The tth topic's word distribution in SCTM is defined as: $\phi_{tv} \propto \prod_{c=1}^{C} \omega_{cv}^{\beta_{tc}}$ (Eq. 2.13), where the ω_c vectors are word distributions (rather than vectors in $\mathbb{R}^V$, as the SPRITE components are), and the $\beta_{tc} \in \{0, 1\}$ variables are indicators denoting whether component c is in topic t. This corresponds to the sparse DAG structure described in Section 5.1.1.1.

If the component structure of SPRITE was used to generate multinomial parameters

rather than Dirichlet parameters, then SPRITE would be equivalent to SCTM if $\boldsymbol{\beta}$ is constrained to binary values and SPRITE's $\boldsymbol{\omega}$ values are interpreted as the log of SCTM's component values. As discussed in Section 5.1.5, if the Dirichlet precision is very large, then the posterior over the parameters will be similar to the prior, in which case SPRITE is similar to SCTM. SPRITE's word distributions are also more general than SCTM in that we allow the $\boldsymbol{\beta}$ coefficients to be weighted. Moreover, the log-linear parameterization leads to much easier parameter estimation than in SCTM.

We also noted in Section 2.4.1.2 a connection between SCTM and sparse additive generative models (SAGE) (Eisenstein et al., 2011), which we described in Section 2.3.3.2. SAGE uses a log-linear parameterization to define word distributions, similar to our parameterization of the Dirichlet priors in SPRITE. As with SCTM, the major difference between SAGE and SPRITE is that SAGE directly defines distributions through log-linear functions rather than Dirichlet vectors. SAGE uses background weights, equivalent to including bias components in SPRITE (Section 5.1.6). SAGE also uses Laplace priors over $\boldsymbol{\omega}$ to induce sparsity, which we noted is a possibility for SPRITE in Section 5.1.1.1.

5.2.2.3 Topic hierarchies and correlations

While the two models in the previous subsection focused on word distributions, SPRITE's priors over topic distributions also have useful characteristics. The component-specific δ_c vectors can be interpreted as common patterns in the topic distributions, where each component is likely to give high weight to groups of topics that tend to occur together.

Each document's α_m coefficients encode the degree to which the different topic groups are present in that document.

Similar properties are captured by the Pachinko allocation model (PAM) (Li and McCallum, 2006). As explained in Section 2.4.1.1, under PAM, each document has a distribution over *supertopics*. Each supertopic is associated with a Dirichlet prior over *subtopic* distributions, where subtopics are the low-level topics that are associated with word parameters $\boldsymbol{\phi}$. Documents also have supertopic-specific distributions over subtopics (drawn from each supertopic-specific Dirichlet prior). Each word in a document is generated by first drawing a supertopic from the document's distribution over supertopics, then drawing a subtopic from that supertopic's document distribution over subtopics, and finally drawing a word from that subtopic's word distribution.

While not equivalent, this is quite similar to SPRITE where document components correspond to supertopics. Each document's α_m coefficients can be interpreted to be similar to a distribution over supertopics, and each δ_c vector is that supertopic's contribution to the prior over subtopics. The prior over the document's topic distribution is thus affected by the document's supertopic weights α_m. A difference is that SPRITE supertopic components are combined multiplicatively, due to the log-linear formulation, whereas in PAM, the combination is a linear mixture.

The SPRITE formulation naturally allows for interesting extensions to PAM. One possibility is to include topic components for the word distributions, in addition to document components, and to tie together δ_{ct} and β_{tc} as suggested in Section 5.1.2. This models

the intuitive characteristic that subtopics belonging to similar supertopics (encoded by $\boldsymbol{\delta}$) should come from similar priors over their word distributions (since they will have similar $\boldsymbol{\beta}$ values). That is, children of a supertopic are topically related—they are likely to share words. This is a richer alternative to the hierarchical variant of PAM proposed by Mimno et al. (2007), which modeled separate word distributions for supertopics and subtopics, but the subtopics were not dependent on the supertopic word distributions.

Another possible extension is to form a tree structure by applying indicator vector priors over $\boldsymbol{\delta}$ to form a tree structure—that is, each δ_c vector is encouraged to have only one nonzero value (Section 5.1.1.2). This would make each subtopic belong to exactly one supertopic, so the model could be interpreted as a true hierarchy. If components are used at both the topic and document levels, then all subtopics sharing a supertopic would come from exactly the same word distribution prior, and each δ_c vector would have non-zero entries only for its unique subtopics.

As discussed in Section 5.1.4, components themselves could depend on higher level components, which would allow this idea to be extended to deeper hierarchies as well.

5.2.2.4 Conditioning on document attributes

SPRITE also naturally provides the ability to condition document topic distributions on features of the document, such as a user rating in a review. To do this, let the number of document components be the number of features, and the value of α_{mc} is the mth document's value of the cth feature. The $\boldsymbol{\delta}$ vectors then influence the document's topic prior

based on the feature values. For example, increasing α_{mc} will increase the prior for the tth topic if δ_{ct} is positive and decrease the prior if δ_{ct} is negative. This is similar to the structure described in the previous subsection, but here the $\boldsymbol{\alpha}$ coefficients are fixed and provided as input, rather than learned and interpreted as supertopic weights.

This is identical to the Dirichlet-multinomial regression (DMR) topic model (Mimno and McCallum, 2008), described in Section 2.3.3.1. The DMR topic model defines each document's Dirichlet prior over topics as a log-linear function of the document's feature values and regression coefficients for each topic. The cth feature's regression coefficients correspond to the δ_c vector in SPRITE.

As with PAM, we can extend this model by including topic components in addition to document components. By tying the topic and document components together, we get the effect that topics with similar regression coefficients will have similar priors over their word distributions. In many settings this will be a desirable effect. For example, when modeling the ratings of reviews, one might assume that topics that are more likely to occur in documents with higher ratings should *a priori* prefer words such as "great" and "excellent", while words like "bad" are more likely in topics that are more likely in documents with low ratings.

5.3 Inference and Optimization

We now discuss how to infer the posterior over the latent variables $\mathbf{z}$ and parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, and to find MAP estimates of the hyperparameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\delta}$, and $\boldsymbol{\omega}$, given their hyperpriors. As with FLDA (Section 3.2), we take a Monte Carlo EM approach, using a collapsed Gibbs sampler to sample from the posterior of the topic assignments $\mathbf{z}$ conditioned on the hyperparameters, then optimizing the hyperparameters using gradient-based optimization conditioned on the samples. Our inference algorithm alternates between one Gibbs iteration and one iteration of gradient ascent, so that the parameters change gradually.

5.3.1 Latent Variable Sampling

As explained in Section 5.1, SPRITE is equivalent to LDA once conditioned on the structured priors. As with FLDA, this allows us to use the same collapsed Gibbs sampler as LDA (Section 2.5.1.1), where each token's topic value z_{mn} is sampled according to:

$$P(z_{mn} = t|\boldsymbol{z} - z_{mn}, \boldsymbol{w}, \tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\phi}}) \propto \left(n_t^m + \tilde{\theta}_{mt}\right) \left(\frac{n_{w_{mn}}^t + \tilde{\phi}_{tw_{mn}}}{\sum_v n_v^t + \tilde{\phi}_{tv}}\right) \tag{5.3}$$

where n_t^m is the number of tokens in the mth document currently assigned to the tth topic, and n_v^t is the number of tokens assigned to the tth topic and the vth vocabulary value, excluding the current assignment of z_{mn}.

This is identical to the LDA sampling distribution (Eq. 2.15), except that the global hyperparameters $\tilde{\theta}$ and $\tilde{\phi}$ have been replaced with document-specific priors $\tilde{\theta}_m$ and topic-

specific priors $\tilde{\phi}_t$, where $\tilde{\theta}_{mt} = \exp(\sum_{c=1}^{C^{(\theta)}} \alpha_{mc}\, \delta_{ct})$ and $\tilde{\phi}_{tv} = \exp(\sum_{c=1}^{C^{(\phi)}} \beta_{tc}\, \omega_{cv})$.

5.3.2 Optimizing the Structured Priors

The parameters of the structured priors are optimized using gradient ascent. The gradient of the log-likelihood has a similar form to that in Sections 2.5.2 and 3.2.

The partial derivative of the collapsed log-likelihood $\mathcal{L}$ of the corpus with respect to each hyperparameter β_{tc} is:

$$\frac{\partial \mathcal{L}}{\partial \beta_{tc}} = \frac{\partial P(\beta)}{\partial \beta_{tc}} + \sum_{v=1}^{V} \omega_{cv} \tilde{\phi}_{tv} \left(\Psi(n_v^t + \tilde{\phi}_{tv}) - \Psi(\tilde{\phi}_{tv}) + \Psi(\textstyle\sum_{v'} \tilde{\phi}_{tv'}) - \Psi(\textstyle\sum_{v'} n_{v'}^t + \tilde{\phi}_{tv'}) \right) \tag{5.4}$$

where n_v^t is the number of times word v is assigned to topic t (in the samples from the E-step), and Ψ is the digamma function, the derivative of the log of the gamma function.

$P(\beta)$ is the hyperprior. For a 0-mean normal prior with variance σ^2, $\frac{\partial P(\beta)}{\partial \beta_{tc}} = -\frac{\beta_{tc}}{\sigma^2}$. Under a Dirichlet($\rho$) prior, when we want each β_c to represent an indicator vector (as in Section 5.1.1.2), $\frac{\partial P(\beta)}{\partial \beta_{tc}} = \frac{\rho - 1}{\beta_{tc}}$.

For unconstrained parameters, we use the standard gradient ascent update rule: ${\beta_t}^{i+1} = {\beta_t}^{i} + \eta_i \nabla \mathcal{L}({\beta_t}^{i})$, with step size η_i at iteration i. For parameters constrained to the simplex (such as when β_t is a soft indicator vector), we use *exponentiated gradient ascent* (Kivinen and Warmuth, 1997) with the update rule: $\beta_{tc}^{i+1} \propto \beta_{tc}^{i} \exp(\eta_i \nabla \mathcal{L}({\beta_t}^{i}))$.

The partial derivatives for the other hyperparameters are similar:

$$\frac{\partial \mathcal{L}}{\partial \omega_{cv}} = \frac{\partial P(\alpha)}{\partial \omega_{cv}} + \sum_{t=1}^{T} \beta_{tc}\tilde{\phi}_{tv} \left(\Psi(n_v^t + \tilde{\phi}_{tv}) - \Psi(\tilde{\phi}_{tv}) + \Psi(\textstyle\sum_{v'} \tilde{\phi}_{tv'}) - \Psi(\textstyle\sum_{v'} n_{v'}^t + \tilde{\phi}_{tv'})\right) \tag{5.5}$$

$$\frac{\partial \mathcal{L}}{\partial \delta_{ct}} = \frac{\partial P(\delta)}{\partial \delta_{ct}} + \sum_{m=1}^{M} \alpha_{mc}\tilde{\theta}_{mt} \left(\Psi(n_t^m + \tilde{\theta}_{mt}) - \Psi(\tilde{\theta}_{mt}) + \Psi(\textstyle\sum_{t'} \tilde{\theta}_{mt'}) - \Psi(\textstyle\sum_{t'} n_{t'}^m + \tilde{\theta}_{mt'})\right) \tag{5.6}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_{mc}} = \frac{\partial P(\alpha)}{\partial \alpha_{mc}} + \sum_{t=1}^{T} \delta_{ct}\tilde{\theta}_{mt} \left(\Psi(n_t^m + \tilde{\theta}_{mt}) - \Psi(\tilde{\theta}_{mt}) + \Psi(\textstyle\sum_{t'} \tilde{\theta}_{mt'}) - \Psi(\textstyle\sum_{t'} n_{t'}^m + \tilde{\theta}_{mt'})\right) \tag{5.7}$$

where n_t^m is the number of tokens assigned to topic t in document m.

5.3.2.1 Tightening the sparsity preferences

For variables that we prefer to be binary but have softened to continuous variables using sparse Beta or Dirichlet priors, we can straightforwardly strengthen the preference to be binary by modifying the objective function to favor the prior more heavily. Specifically, under a Dirichlet(ρ<1) prior, we will introduce a scaling parameter $\tau_i \geq 1$ to the prior log-likelihood: $\tau_i \log P(\beta_t)$ with partial derivative $\tau_i \frac{\rho-1}{\beta_{tc}}$, which adds extra weight to the

sparse Dirichlet prior in the objective at iteration i. The algorithm used in our experiments begins with $\tau_1 = 1$ and optionally increases τ_i over time. This is a deterministic annealing approach, where τ_i corresponds to an inverse temperature (Ueda and Nakano, 1998; Smith and Eisner, 2004).

As τ approaches infinity, the prior-annealed MAP objective $\max_\beta P(\phi|\beta)P(\beta)^\tau$ approaches $\max_\beta P(\phi|\beta) \max_\beta P(\beta)$. Annealing only the prior $P(\beta)$ results in maximization of this term only, while the outer $\max$ chooses a good β under $P(\phi|\beta)$ as a tie-breaker among all β values that maximize the inner $\max$ (binary-valued β).[3]

We note that an alternative possibility is to optimize the inverse temperature τ rather than to deterministically increase it. By optimizing τ rather than annealing it, we can *learn* the optimal amount of sparsity in the model. One could also use different τ variables for different parameters, so that different parameter vectors can have different degrees of sparsity. This is a more flexible approach that would allow one to learn that some portions of the model are tree-structured while others are DAG-structured, for example.

5.3.2.2 Posterior inference of the priors

While we use maximum *a posteriori* (MAP) estimation of the Dirichlet parameters in this thesis, it is important to note that posterior inference of these parameters is also a possibility. Approximate posterior inference of the $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\delta}$, and $\boldsymbol{\omega}$ parameters could be done using the methods described in Section 2.5, including sampling and variational

[3]Other modifications could be made to the objective function to induce sparsity, such as entropy regularization (Balasubramanyan and Cohen, 2013).

inference.

Posterior inference may be particularly beneficial for the document-specific α_m parameters, whose MAP estimates may be unreliable since they are learned from a relatively small amount of data (the tokens within potentially short documents), in the same way that posterior inference is important for the document θ_m parameters (Blei et al., 2003b; Wallach et al., 2009a).

5.4 Experiments

In this section, we present a small set of experiments to show how SPRITE behaves under different structures. Specifically, we experiment with the basic SPRITE model for the three structures described in Section 5.1.1: a DAG, a tree, and a factored forest. For each structure, we also experiment with each type of component: document, topic, and both.

The purpose of these experiments is to understand a few variations of SPRITE in its most general form. We reserve a more exhaustive set of experiments, in which we compare to several baselines, for Chapter 6, where we consider more specific instantiations of SPRITE.

5.4.1 Experimental Details

We applied SPRITE to three corpora:

- *Debates:* A set of floor debates from the 109th–112th U.S. Congress, collected by Nguyen et al. (2013), who applied a hierarchical topic model (called supervised hi-

erarchical LDA) to this data. We took a sample of 5,000 documents from the House debates (850,374 tokens; 7,426 types), balanced across party affiliation.

- *Reviews:* A subset of the doctor reviews corpus described in Section 4.2. This corpus contains 20,000 documents (476,991 tokens; 10,158 types), balanced across positive/negative scores.

- *Abstracts:* A set of 957 abstracts from the ACL anthology (97,168 tokens; 8,246 types). This is a subset of the dataset analyzed with FLDA in Section 3.3.

We set T=50 topics and C=10 components for *Debates* and *Abstracts*, and K=20 and C=5 for *Reviews*. For the factored structures, we use two factors, with one factor having more components than the other: 3 and 7 components for *Debates*, and 2 and 3 components for *Reviews* (the total number of components across the two factors is therefore the same as for the DAG and tree experiments). These values were chosen as a qualitative preference, not optimized for predictive performance, but we experiment with different values in the next chapter.

We set $C = C^{(\phi)} = C^{(\theta)}$, tying together the components (for experiments that use both types) using the method described in Section 5.1.2. We also include "bias" components as described in Section 5.1.6, which are not counted toward the value of C.

We create weighted structures using the approach suggested in Section 5.1.1.2, where the coefficients are replaced with a product of two variables: a soft indicator and a real-valued weight, constrained to be positive (as suggested in Section 5.1.3). That is, we let

$\beta_t = \boldsymbol{s} \odot \hat{\beta}_t$ (where $\odot$ is the entry-wise product), such that $\boldsymbol{s}$ is an indicator vector with a sparse Dirichlet(ρ=0.01) prior. We apply weak regularization to all other hyperparameters via a $\mathcal{N}(0, 10^2)$ prior.

For the tree and factored structures, we use an annealing schedule (described in Section 5.3.2.1) of $\tau_i = 1.003^i$ to induce binary values. Section 6.2.2.4 shows experiments with different schedules, and we selected 1.003^i because it results in binary values. For the DAG structure, we simply set $\tau = 0$ which removes the sparsity-inducing prior from the objective altogether.

We set the gradient ascent step size η_i according to AdaGrad (Duchi et al., 2011), where the step size is the inverse of the sum of squared historical gradients. AdaGrad decayed too quickly for the $\boldsymbol{s}$ variables, such that their values would barely change from their initial value (uniform in the simplex) after thousands of iterations. For these, we used a variant suggested by Zeiler (2012) which uses an average of historical gradients rather than a sum.

We ran the inference algorithm for 5000 iterations, estimating the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ by averaging the final 100 iterations. The results are averaged across 10 randomly initialized samplers.

5.4.2 Quantitative Evaluation

We evaluated the different SPRITE structures using two measures of quality. The first is perplexity of held-out text using the "document completion" method described in Section 2.6.1. The second is the coherence metric of Mimno et al. (2011) described in Section

	Perplexity			Coherence		
	DAG	Tree	Factored	DAG	Tree	Factored
	Debates					
Document	1572.0 ± 0.9	1568.7 ± 2.0	1566.8 ± 2.0	-342.9 ± 1.2	-346.0 ± 0.9	-343.2 ± 1.0
Topic	1575.0 ± 1.5	1573.4 ± 1.8	1559.3 ± 1.5	-342.4 ± 0.6	-339.2 ± 1.7	**-333.9** ± 0.9
Combined	1566.7 ± 1.7	1559.9 ± 1.9	**1552.5** ± 1.9	-342.9 ± 1.3	-342.6 ± 1.2	-340.3 ± 1.0
	Reviews					
Document	1456.9 ± 3.8	**1446.4** ± 4.0	1450.4 ± 5.5	-512.2 ± 4.6	-527.9 ± 6.5	-535.4 ± 7.4
Topic	1508.5 ± 1.7	1517.9 ± 2.0	1502.0 ± 1.9	-500.1 ± 1.2	-499.0 ± 0.9	**-486.1** ± 1.5
Combined	1464.1 ± 3.3	1455.1 ± 5.6	1448.5 ± 8.5	-504.9 ± 1.4	-527.8 ± 6.1	-535.5 ± 8.2
	Abstracts					
Document	3107.7 ± 7.7	**3089.5** ± 9.1	3098.7 ± 10.2	-393.2 ± 0.8	-390.8 ± 0.9	-392.8 ± 1.5
Topic	3241.7 ± 2.1	3455.9 ± 10.2	3507.4 ± 9.7	-389.0 ± 0.8	-388.8 ± 0.7	**-332.2** ± 1.1
Combined	3200.8 ± 3.5	3307.2 ± 7.8	3364.9 ± 19.1	-373.1 ± 0.8	-360.6 ± 0.9	-342.3 ± 0.9

Table 5.2: Quantitative results for different structures (columns) and different components (rows) for two metrics (± std. error) across three datasets. The best (structure, component) pair for each dataset and metric is in bold.

2.6.2, defined as:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{l=2}^{20}\sum_{j=1}^{l-1}\log\frac{DF(v_{tl},v_{tj})+1}{DF(v_{tj})}$$

where $DF(v,w)$ is the document frequency of words v and w (the number of documents in which they both occur), $DF(v)$ is the document frequency of word v, and v_{ti} is the ith most probable word in topic t.

Note that the coherence metric only measures the quality of word clusters, ignoring the potential improvement in interpretability of organizing the topics into certain structures. However, it is still useful as an alternative measure of performance and utility, independent of the models' predictive abilities.

Table 5.2 shows the results of these two metrics for different structures and different component types.

Some trends are clear and consistent. Topic components always hurt perplexity (by very

significant margins on *Reviews* and *Abstracts*), while these components typically improve coherence. It has previously been observed that perplexity and topic quality are not correlated (Chang et al., 2009), and our finding is consistent with this. These results show that the choice of components depends on the task at hand. Combining the two components tends to produce results somewhere in between, suggesting that using both component types is a reasonable "default" setting.

Document components usually improve perplexity, likely due to the nature of the document completion setup, in which half of each document is held out. The document components capture correlations between topics, so by inferring the components that generated the first half of the document, the prior is adjusted to give more probability to topics that are likely to occur in the unseen second half. Another interesting trend is that the factored structure tends to perform well under both metrics, with the lowest perplexity and highest coherence in a majority of the nine comparisons (i.e., each row). Perhaps the models are capturing a natural factorization present in the data.

5.4.3 Qualitative Analysis

We investigate SPRITE qualitatively, focusing on the factored structure to compare to what was learned by FLDA. Figure 5.6 shows examples of components from each factor along with example topics that draw from all pairs of these components, learned on the *Abstracts* corpus. We find that the factor with the smaller number of components (left of the figure) seems to decompose into components representing the major themes or disciplines

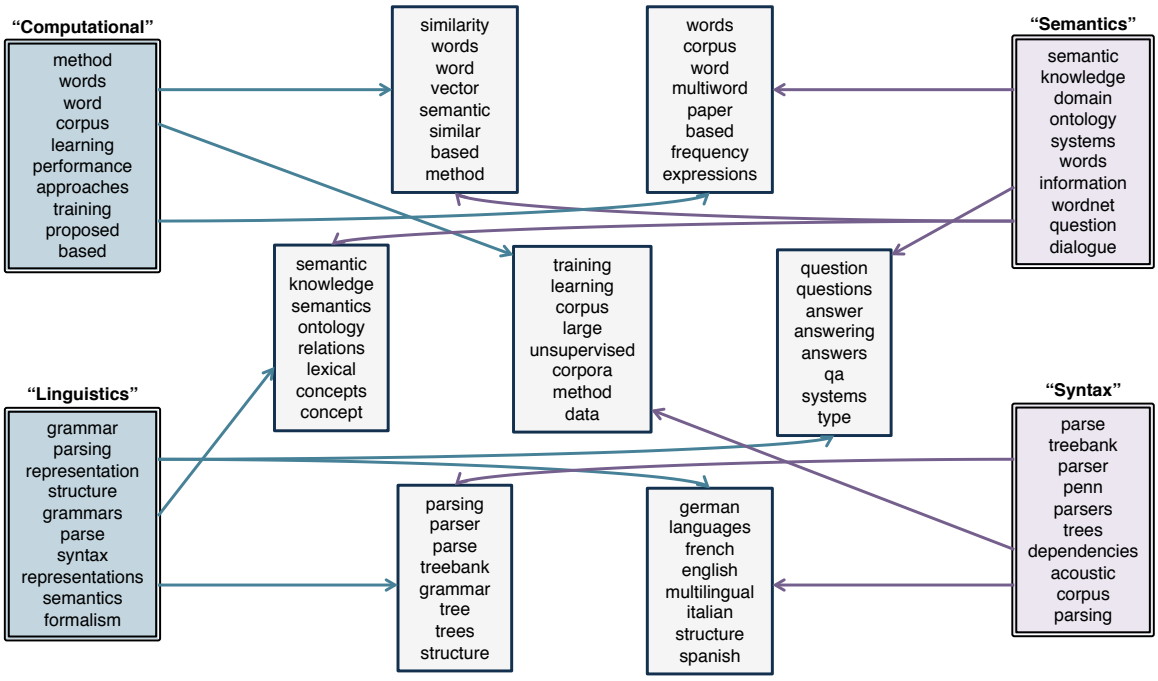


Figure 5.6: Examples of topics (gray boxes) and components (colored boxes) learned on the *Abstracts* corpus with 50 topics using a factored structure. The components have been grouped into two factors, one factor with 3 components (left) and one with 7 (right), with two examples shown from each. Each topic prior draws from exactly one component from each factor.

found in ACL abstracts, with one component expressing computational approaches (top) and the other expressing linguistic theory (bottom). The third component (not shown) has words associated with speech, including {*spoken*, *speech*, *recognition*}.

The factor shown on the right seems to decompose into different research topics: one component represents semantics (top) and another component represents syntax (bottom). Other components include morphology (top words including {*segmentation*, *chinese*, *morphology*}) and information retrieval (top words including {*documents*, *retrieval*, *ir*}).

Many of the topics intuitively follow from the components of these two factors. For

example, the two topics expressing vector space models and distributional semantics (top left and right) both draw from the COMPUTATIONAL and SEMANTICS components, while the topics expressing ontologies and question answering (middle left and right) draw from LINGUISTICS and SEMANTICS.

The factorization is similar to what was induced by FLDA. Figure 3.6 of Chapter 3 shows components that look similar to the computational methods and linguistic theory components here, and the factor with the largest number of components also decomposes by research topic. These results show that SPRITE is capable of recovering similar structures as FLDA, a more specialized model. SPRITE is also much more flexible than FLDA, as discussed in Section 5.2.1. While FLDA strictly models a one-to-one mapping of topics to each pair of components, SPRITE allows multiple topics to belong to the same pair (as in the semantics examples above), and conversely SPRITE does not require that all pairs have an associated topic.

5.5 Summary

This chapter presented SPRITE, a family of topic models that utilize structured priors to induce preferred topic structures. These structures were incorporated into the priors of both the word and topic distributions, unlike most previous work which considered one or the other.

Specific instantiations of SPRITE are similar or equivalent to several topic models. Our

framework has made clear advancements with respect to existing structured topic models. For example, SPRITE is more general and offers simpler inference than the shared components topic model (Gormley et al., 2010), and SPRITE allows for more flexible and scalable factored structures than FLDA, as described in Sections 5.2.2.2 and 5.2.1, respectively. Both of these models were motivated by their ability to learn interesting structures, rather than their performance at any predictive task. Similarly, our goal in this chapter was not to provide state-of-the-art results for a particular task, but to demonstrate a framework for learning structures that are richer than previous structured models. Therefore, our experiments focused on understanding how the different possible priors for SPRITE affect performance. Our results point to general rules of thumb: document components are useful for improving prediction, topic components improve coherence, and factored structures tend to perform well in general.

Ultimately, the model design choice depends on the application and the user needs. By unifying such a wide variety of topic models, SPRITE can serve as a common framework for enabling model exploration and bringing application-specific preferences and structure into topic models.

Chapter 6

Applications of SPRITE

In the previous chapter, we introduced SPRITE in its most general form. This chapter presents specific SPRITE-based models designed for the application of inferring perspective (i.e., opinion, ideology, or sentiment) from text. Not only do we provide additional evaluation of SPRITE through these applications, but the application-specific models are pedagogically useful, illustrating the purpose of the different structures and features described abstractly in Chapter 5.

This chapter mirrors Chapter 4 (applications of factorial LDA) with two main content sections, followed by related work and a concluding section. Both content sections focus on modeling perspective, with the second section introducing a more complex model. The first section (6.1) focuses specifically on health-related opinion in social media, and introduces methods for incorporating domain knowledge into the priors. The second section (6.2) is more general and its purpose is to provide additional evaluation and showcase a more

complex model structure. This chapter focuses on standard topic model evaluation, as described in Section 2.6, rather than application-specific evaluation, as in Chapter 4.

6.1 Predicting Public Opinion in Social Media

This section will present a SPRITE-based model that is used to infer *perspective* in text, where perspective exists on a real-valued spectrum that can be positive or negative (or neutral, with a value of 0). We focus on Twitter messages related to three public health and political issues: gun control, vaccination, and smoking. Our model aims to characterize the content of messages discussing polarizing issues such as gun control in the U.S., while learning topical associations between different perspective values on these issues. This section includes work from Benton et al. (2015).

This SPRITE model uses only a single component, $C^{(\theta)} = C^{(\phi)} = 1$ (in addition to a bias component), as will be explained in more detail in Section 6.1.2, which makes the model relatively simple and easy to understand. (A more complex SPRITE variant will be used in Section 6.2.) This single-component model is still more complex than LDA because the component is multiplied by coefficients that vary across topics and documents, reflecting the different perspective values of topics and documents, in contrast to LDA where the coefficients of components have an implicit value of 1. The model presented in this section also ties together the topic and document components (described in Section 5.1.2 of the previous chapter), thus demonstrating important structural properties of SPRITE

in the context of the task of perspective modeling.

As in Chapter 4, a challenge with using topic models for a specific application is the need to ground topics in meaningful concepts. We will show that prior knowledge can be incorporated into the priors similarly to incorporating knowledge into FLDA in Chapter 4. We will focus on methods of creating prior knowledge that are relatively inexpensive and do not require data annotation, based on Twitter hashtags and external survey data.

6.1.1 Task and Motivation

Social media has proved invaluable for research in a variety of social and health sciences, and in particular is a valuable resource for measuring public opinion. A large body of research has shown that public opinion can be inferred from Twitter data (O'Connor et al., 2010), including consumer sentiment (Bollen et al., 2011; Chamlertwat et al., 2012), political orientation (Tumasjan et al., 2010; Conover et al., 2011), and attitudes toward health issues (Salathe and Khandelwal, 2011; Myslin et al., 2013).

Much of this work on opinion inference relies on domain-specific models, sometimes requiring human-labeled data. As an alternative, we propose a topic modeling framework that is minimally supervised. In this section, we present a SPRITE-based model that can not only infer topics from social media messages, but learn to associate the topics with different perspective values (Section 6.1.2). Such a model can be used to summarize high-level opinions on different topics within a social media collection.

Since a fully unsupervised topic model is unlikely to learn coherent perspective associ-

Dataset	Size	Example Hashtags		U.S. State Data
Guns	100K	#gunsense, #guncontrolnow, #demandaplan	#gunrights, #noguncontrol, #gunfriendly	Firearm in household
Vaccines	20K	#protectusall, #vaccineswork, #sciencedenial	#vaccinedangers, #cdcfraud, #savekids	Flu shot in past year
Smoking	40K	#nowsmoking, #smokelover, #cigarfest	#quitsmoking, #notobacco, #lunghealth	Current smoker

Table 6.1: A summary of the three Twitter datasets, including example perspective hashtags and survey questions. "Size" refers to the number of tweets.

ations, we propose a method for incorporating domain knowledge into SPRITE's structured priors. A primary contribution of this study is to show how to guide the model toward learning the desired patterns while requiring only minimal human input (Section 6.1.3). We introduce a method of *lightweight* supervision via two types of information that can be obtained with relatively little effort: (i) hashtags associated with different stances on an issue (e.g., #GunControlNow vs #NoGunControl), and (ii) external survey data that is correlated with perspective (e.g., the percentage of gun owners in each U.S. state). This side information is incorporated into our model as a prior over the inferred perspective. Our experiments show that our model with lightweight supervision produces interpretable topics and perspectives and improves performance at predictive tasks (Section 6.1.4).

6.1.1.1 Datasets

We describe our three Twitter datasets to illustrate the type of social media data for which the SPRITE-based perspective model is designed. Each dataset is based on collecting keyword-filtered tweets discussing public health issues for which public opinion is divided across contrasting perspectives: gun control, vaccinations, and smoking. Tweets were collected from Dec. 2012 through Jan. 2015 using the Twitter search API with various

topic-specific keywords for the three respective datasets.

We augmented these datasets (summarized in Table 6.1) with additional information that we will use as supervision:

6.1.1.1.1 HASHTAGS

We identified a set of hashtags associated with two polar perspectives or stances associated with each of the three issues. For each dataset, we first looked through the 1000 most frequent hashtags and identified those clearly indicating a stance on the issue (i.e., for or against). To find less frequent but relevant hashtags, we computed the pointwise mutual information (PMI) between the initially identified hashtags and all others, and then looked through the top 1000 hashtags with highest PMI. In the end, we obtained 11 hashtags unambiguously for gun control and 11 against, 15 and 30 hashtags supporting and opposing smoking, and 18 and 20 hashtags supporting and opposing vaccination.

6.1.1.1.2 SURVEYS

We collected relevant survey data for each issue from the Behavioral Risk Factor Surveillance System,[1] an annual phone survey of hundreds of thousands of American adults (exact numbers vary by year), which was chosen for its very large and geographically widespread sample. We selected the following questions: the percentage of respondents in each U.S. state who have a firearm in their house (data from 2001, when the question was last asked), who have had a flu shot in the past year (2013), and who are current smokers (2013). While

[1] `http://apps.nccd.cdc.gov/gisbrfss/`

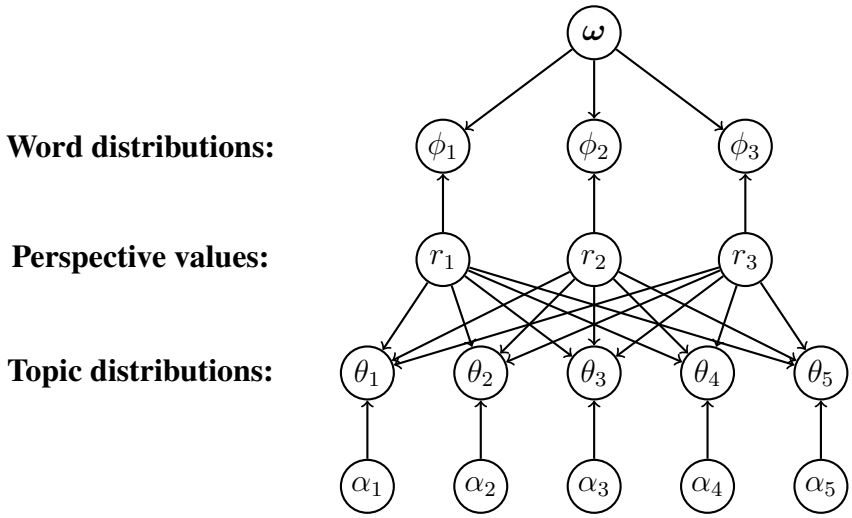


Figure 6.1: The graphical model for the parameters of the SPRITE-based perspective model in Section 6.1.2, using $T = 3$ topics and $M = 5$ documents as an example. The $\boldsymbol{\omega}$ component at the top is a vector of length V, while the r_t and α_m variables are scalars. It can be seen that the perspective $\boldsymbol{r}$ values influence both the word distributions in topics and the topic distributions in documents.

these survey questions are not directly about one's perspective on these issues—for example, someone might support the right to own firearms even if they do not own one, and someone might support vaccination even if they did not get a flu shot—our hypothesis is that these attributes will be correlated with different stances. For instance, tweeters from states with high gun ownership may be more likely to be against gun control. We resolved the U.S. state for approximately one third of tweets using *Carmen*, a tweet geolocation system (Dredze et al., 2013).

6.1.2 A Joint Model of Perspective and Topic

Recall that under SPRITE, each document's distribution over topics θ_m has its own prior, Dirichlet($\tilde{\theta}_m$), and each topic's distribution over words ϕ_t has its own prior, Dirichlet($\tilde{\phi}_t$), such that each prior is defined in terms of components that are shared across the different documents or topics. We now describe a variant of SPRITE that defines the priors in terms of variables about *perspective*.

In our model, each topic t is associated with a real-valued perspective variable r_t. We assume that "perspective", as modeled in this section, exists on a continuum from positive to negative. Similarly, each document m is associated with a perspective value, α_{m1}, and each word v in the vocabulary is associated with a perspective value, ω_{1v}.

These variables are used to influence the topic model priors in two ways. First, the prior over each document's topic distribution θ_m is a function of $\alpha_{m1} r_t$, which means that a topic t will have an increased prior probability in a document if the topic's perspective value r_t has the same sign as the document's perspective value α_{m1}, and have a decreased prior if the signs are opposite. This has the effect that, conditioned on the document's perspective value, certain topics will become more or less likely in a document. The perspective value r_t corresponds to the component value δ_{ct} using the SPRITE notation from Section 5.1, where $c = 1$.

Second, the prior over each topic's word distribution ϕ_t is a function of $\omega_{1v} r_t$, which means that a word v will have an increased prior probability in a topic if the topic's perspective value r_t has the same sign as the word's perspective value ω_{1v}. This has the effect

that topics with similar perspective values are *a priori* more likely to share words in common than those of opposite perspective values. Here, the perspective value r_t corresponds to the component coefficient β_{tc} using the SPRITE notation, where $c = 1$. This means that, implicitly, $\beta_{tc} = \delta_{ct}$.

The generative story is:

1. For each document m:
   (a) $\tilde{\theta}_{mt} = \exp(\alpha_{m0}\delta_{0t} + \alpha_{m1}\delta_{1t})$, where $\alpha_{m0} = 1$ and $\delta_{1t} = r_t$, for each topic t
   (b) $\theta_m \sim \text{Dirichlet}(\tilde{\theta}_m)$
2. For each topic t:
   (a) $\tilde{\phi}_{tv} = \exp(\beta_{t0}\omega_{0v} + \beta_{t1}\omega_{tv})$, where $\beta_{t0} = 1$ and $\beta_{t1} = r_t$, for each word v
   (b) $\phi_t \sim \text{Dirichlet}(\tilde{\phi}_t)$
3. For each token n in each document m:
   (a) Sample topic index $z_{mn} \sim \theta_m$
   (b) Sample word token $w_{mn} \sim \phi_{z_{mn}}$

The δ_0 and ω_0 vectors are bias components (Section 5.1.6) whose coefficients are always 1. The bias components act as overall weights for each topic t or word v, which are then adjusted by the second term. By showing that the scalar r_t appears in both the topic and document priors, it can be seen that we have tied the two together (Section 5.1.2), i.e.,

$r_t = \beta_{t1} = \delta_{1t}$, using the terminology of the model definition in Section 5.1. This model will be extended to more than one component in Section 6.2.

To summarize, the model includes a number of parameters that can be analyzed to provide additional insights into the relationship between topics and perspective:

- r_t indicates the tth topic's association with perspective, both in magnitude and direction. A positive value means that (i) the topic is more likely to occur in documents with a positive perspective, and (ii) the topic is more likely to have words in common with other topics with positive r. Topics that are not strongly associated with a perspective can have r values close to zero. These values are useful for interpretability because they indicate the degree to which each topic represents a perspective, and in what direction.

- ω_{1v} indicates the vth word's association with perspective. A positive value means that this word is more likely to have high probability in a topic with positive r. The words with highest and lowest ω values can therefore be viewed as having strong associations with positive or negative perspective, and provide an extra set of words that can be interpreted by humans, in addition to the topics themselves (the words with highest weight in each ϕ_t). These values are useful for interpretability because they can be used to understand which words are most associated with positive or negative perspective.

- α_{m1} indicates the mth document's association with perspective. A positive value

means that topics with positive r have higher prior probability in the document. These values can be interpreted as the perspective (both magnitude and direction) of each document, which can be useful for grouping tweets by perspective. These values are influenced by the prior in Eq. 6.1, but can adapt based on the topic content of the documents.

This model thus learns associations between perspective values and topics (through $\boldsymbol{r}$), documents (through $\boldsymbol{\alpha}$), and words (through $\boldsymbol{\omega}$), providing a richer view of text than an unstructured topic model.

6.1.3 Incorporating Lightweight Supervision

If the $\boldsymbol{r}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\omega}$ parameters are all estimated without supervision, the model is unlikely to learn patterns that actually correspond to perspective. We address this by using document-level labels to inform the values of $\boldsymbol{\alpha}$.

The mth document is associated with two variables used as supervision, s_{m0} and s_{m1}, indicating a value of perspective based on hashtag presence (s_{m0}) or state-level survey data (s_{m1}), as outlined in Section 6.1.1.1.

We set $s_{m0} = 1$ if a tweet m has a hashtag associated with a positive perspective, -1 for a negative perspective, and 0 if the tweet contains no perspective hashtags. We set s_{m1} to the survey percentage for the state that tweet m is resolved to, and we set $s_{m1} = 0$ for tweets that cannot be resolved to a U.S. state. We applied z-score normalization to the survey values so that the sign of the value corresponds to whether it is above or below the

average. Specifically, the mean of all survey values is subtracted from each survey value, and the result is divided by the standard deviation: the new value is called the *z-score* or *standard score*.

We choose the sign of the hashtag values to correspond to the survey values. For example, our assumption is that states with higher vaccination rates are more likely to be pro-vaccination, so we set pro-vaccination hashtags have positive s_{m0} values.

We incorporate these two variables into the model as a prior distribution over each document's perspective parameter α_{m1}. This prior encourages model parameters to be close to the supervised values, while allowing the parameters to deviate from the input, as the s_m values are not always accurate or available. Specifically, the prior is a normal distribution whose mean is a weighted combination of the variables:

$$\alpha_{m1} \sim \mathcal{N}(\lambda_0 s_{m0} + \lambda_1 s_{m1}, \sigma^2) \tag{6.1}$$

The $\boldsymbol{\lambda}$ parameters are estimated to find an optimal balance between the two types of information. The state survey data is less accurate but more commonly available, since only a small fraction of relevant tweets will contain a distinctive hashtag. If no supervision is available for a tweet, then the mean becomes 0 and the prior simply serves as a regularizer, rather than influencing the parameter in either direction. The variance σ^2 can be tuned to control the strength of this prior.

The inference algorithm for SPRITE involves alternating between one iteration of Gibbs

sampling and one iteration of gradient ascent for the hyperparameters $\boldsymbol{\alpha}$, $\boldsymbol{\omega}$, $\boldsymbol{r}$, and $\boldsymbol{\lambda}$. The first three hyperparameters are optimized as described in Section 5.3 (recall that $\boldsymbol{r}$ is equivalent to $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ in the notation of Chapter 5). The partial derivative of the log-likelihood $\mathcal{L}$ with respective to λ_0 is:

$$\frac{\partial \mathcal{L}}{\partial \lambda_0} = \sum_{m=1}^{M} s_{m0} \frac{\alpha_{m1} - (\lambda_0 s_{m0} + \lambda_1 s_{m1})}{\sigma^2} \tag{6.2}$$

The partial for λ_1 is nearly identical, but the leftmost s_{m0} term is replaced with s_{m1}.

6.1.3.1 Alternative methods of supervision

We will briefly discuss some alternative ideas for incorporating hashtag and survey supervision than the approach described above.

Instead of the linear interpolation of the two supervision variables in Eq. 6.1, one might consider using only one variable or the other. For example, the hashtag variable (which is the more accurate of the two supervision variables) could be used when known, and the survey variable could be used only when a hashtag is not present. Another improvement to the supervision variables would be to infer their values when they are unknown, rather than keeping the values fixed to 0.

Rather than using hashtag values to create supervision over the value of the document perspective α_{m1}, the hashtags could instead be used to create supervision for the word weights, $\boldsymbol{\omega}$, similar to the approaches used in Chapter 4, particularly the "seed word" ap-

proach in Section 4.2.4. Specifically, we can use a normal prior over the ω_{1v} values for hashtags such that the mean of the prior is positive or negative for hashtags representing positive or negative perspective. This would bias the model toward learning perspective values that agree with the sign of the hashtag perspectives.

6.1.4 Experiments

We now investigate the performance of this perspective model on the three Twitter datasets described in Section 6.1.1.1, investigating predictive performance and analyzing examples of learned topics.

6.1.4.1 Experimental details

Less than 1% of tweets in the collections contained the perspective hashtags used for supervision. In practice, one would typically up-sample the tweets with labels (hashtags) in order to learn a more accurate model, but this could also bias the topical content of the dataset. We therefore experiment with differing degrees of up-sampling, to measure the topic model performance when different amounts of supervision are included. We created two variants of the datasets, varying the way in which tweets are sampled to better balance the proportion that contain supervision:

- Tweets were sampled so that 50% contain perspective hashtags and 50% are resolved to U.S. states. (Approximately 25% of tweets in this version contain both types of

information, and another 25% contain neither.)

- Tweets were sampled so that 5% contain perspective hashtags and 33% have U.S. states. This is more (but not completely) realistic, with only a small percentage containing hashtags, and with an accurate proportion of tweets with states.

We note that there may be other approaches to handling hashtag sparsity than simply up-sampling data. For example, one might initially run a sampler on an hashtag-heavy dataset to learn good initial parameters, but then gradually include more tweets without hashtags, so that the full dataset can be represented. This idea is similar to the approach for combining labeled and unlabeled data suggested in Section 4.1.3.2.

Any token occurring less than ten times in a dataset was removed from tweets, as were stopwords, URLs, usernames, and common Twitter-specific tokens (e.g., rt). We ran 5000 Gibbs iterations, averaging the values sampled from the final 100. We used a low variance $\sigma^2 = 0.25$ for the prior over α_{m1} to encourage the parameters to stay near the supervised values. The various optimization parameters, such as the regularization variance and initial values, were minimally tuned during development to obtain interpretable output (see Section 6.4 for more discussion of tuning these parameters).

To improve inference time on these social media datasets, we used a multi-threaded SPRITE implementation following the "approximate distributed" sampling technique described in Section 2.5.3 of Chapter 2, in which samplers are independently run and combined across multiple partitions of the dataset.

	5% hashtag sample						50% hashtag sample					
Model	Guns		Vaccines		Smoking		Guns		Vaccines		Smoking	
bag-of-words	12.67	n/a	6.72	n/a	3.87	n/a	12.58	n/a	7.06	n/a	6.41	n/a
LDA	16.44	**2119**	**6.40**	2006	5.44	1802	12.34	2190	**6.14**	3055	6.06	2650
SPRITE	12.86	2501	8.21	2031	6.69	1790	12.36	2064	6.16	2990	5.42	2495
+Hashtags	**8.40**	2568	6.67	1995	5.45	1765	12.34	2068	6.95	**2730**	3.77	2304
+Survey	**8.40**	2530	6.72	2012	**3.74**	**1742**	**12.33**	2140	**6.14**	3137	**3.64**	2507
Full model	9.38	2597	7.74	**1968**	7.72	1769	12.36	**1959**	**6.14**	2786	4.49	**2252**

Table 6.2: Average test RMSE for the survey regression task (left number), along with lowest held-out perplexity (right number), for each dataset/model. The lowest (best) score for each dataset is in bold.

6.1.4.2 Quantitative evaluation

We evaluated our full SPRITE-based model against four topic model baselines: LDA with hyperparameter optimization, our model with no supervision (i.e., α_{m1} has a zero-mean normal prior rather than the informed mean in Eq. 6.1), and variants with only one of the two types of supervision (hashtags or survey data).

We performed two experiments to assess model quality. First, we computed held-out perplexity of each model using the "document completion" approach (Section 2.6.1), where every other token was used for training, with perplexity measured on the remaining tokens. We varied the number of topics $T \in \{10, 25, 50, 100\}$ topics and report the best result in Table 6.2.

Second, we evaluated the utility of topics as features for predicting the survey value for each U.S. state, reflecting how well topics capture themes relevant to the survey question. We inferred θ_m for each tweet and then averaged these topic vectors over all tweets originating from each state, to construct 50 feature vectors per model. We used these features

in a regularized linear regression model where each row corresponded to the average θ per state. That is, there are fifty instances (one per state), where each instance has T features (one per topic), where each feature is the mean value of θ_{mt} across all tweets for the state.

Average root mean-squared error (RMSE) was computed using five fold cross-validation: 80% of the 50 US states were used to train, 10% were used to tune the ℓ_2 regularization coefficient, and 10% were used for evaluation. For each cross-validation fold, the topic models only used supervision for tweets from the 40 states used for training data; for the held-out states, the supervision s_m variables had values of 0, so that the held-out survey values (which are being predicted) are not available to the topic model.

We also compare to a standard bag-of-words baseline, where features were normalized counts collapsed across state. Bag-of-words features were only used for this baseline, and were not combined with the topic features.

Results are shown in Table 6.2. We find that the full SPRITE model has the lowest perplexity in five of six datasets, demonstrating that it is learning a better representation of the data than LDA. SPRITE-based models also give the lowest survey prediction error in four of the datasets (and tying in another), demonstrating that including the survey data as supervision indeed yields representations of the data that are more predictive of the held-out values. The model that uses only survey data does better than the models that include hashtags, which is perhaps intuitive because this is the same type of data that is being regressed (though we note that the held-out survey values are not used by the topic model). More surprising is that the model using only hashtags typically outperforms the full

model combining both (under both metrics), which suggests there is room for improvement in our formulation for combining the two disparate types of side information into a joint model, as discussed in Section 6.1.3.1. Nevertheless, our results clearly show that including supervision improves the topics that are learned on these datasets.

6.1.4.3 Qualitative analysis

Tables 6.3–6.4 show examples of topics and perspectives learned by our model. These topics have associations with different perspective values on these issues (for or against gun control or vaccination). For instance, different topics about the flu vaccine have different perspective values: the "for" topic appears to encourage people to get a flu shot, while the two "against" topics discuss the recall of a flu vaccine after fears of death in Italy, and the effectiveness of the flu shot in the most recent season (which was lower than usual due to an unexpected influenza strain).

6.2 Jointly Learning Perspective and Topic Hierarchies

We now extend the joint topic and perspective model from Section 6.1 so that in addition to associating topics with perspective, topics are also organized into a hierarchy. This single SPRITE model encompasses nearly all of the structures and extensions described in

For gun control			**Against gun control**		
Highest component values			Lowest component values		
violence	$\omega_{1v} = .95$		hard	$\omega_{1v} = -.84$	
#newtown	$\omega_{1v} = .94$		like	$\omega_{1v} = -.83$	
deaths	$\omega_{1v} = .91$		oh	$\omega_{1v} = -.83$	
via	$\omega_{1v} = .91$		instead	$\omega_{1v} = -.82$	
campaign	$\omega_{1v} = .89$		lot	$\omega_{1v} = -.82$	
#demandaplan	$\omega_{1v} = .88$		better	$\omega_{1v} = -.81$	
annual	$\omega_{1v} = .86$		government	$\omega_{1v} = -.81$	
related	$\omega_{1v} = .86$		even	$\omega_{1v} = -.81$	
congress	$\omega_{1v} = .86$		think	$\omega_{1v} = -.80$	
free	$\omega_{1v} = .85$		cannot	$\omega_{1v} = -.80$	
Topic 27 $r_t = 0.34$	Topic 24 $r_t = 0.16$	Topic 4 $r_t = 0.11$	Topic 25 $r_t = -0.11$	Topic 22 $r_t = -0.29$	Topic 47 $r_t = -0.29$
control	violence	children	nra	teachers	never
bloomberg	culture	kids	war	school	someone
one	less	likely	even	armed	trigger
obama	sense	times	keep	schools	shoot
violence	problem	murdered	murder	carry	one
number	makes	giving	members	god	person
time	world	nothing	liberals	kids	pull
president	country	stand	fear	teacher	kids
lead	common	sorry	government	protect	point
agenda	america's	insane	call	security	anyone

Table 6.3: Examples of topics learned on the gun dataset (with 5% hashtags, 50 topics). The top row shows the words with highest positive/negative value of ω, indicating common word associations with each perspective. We show examples of three topics associated with each perspective, as defined by the topics' inferred r values. For increased lexical diversity, we excluded words common across many topics in the two datasets ("gun*") from this output.

For vaccination			**Against vaccination**		
Highest component values			Lowest component values		
#vaccinate		$\omega_{1v} = 1.63$	#cdcwhistleblower		$\omega_{1v} = -1.82$
kids		$\omega_{1v} = 1.62$	injury		$\omega_{1v} = -1.70$
#vaccineswork		$\omega_{1v} = 1.60$	#hearthiswell		$\omega_{1v} = -1.60$
anti-vaxxers		$\omega_{1v} = 1.46$	cdc		$\omega_{1v} = -1.56$
preventable		$\omega_{1v} = 1.46$	truth		$\omega_{1v} = -1.54$
children		$\omega_{1v} = 1.46$	#vaccine		$\omega_{1v} = -1.53$
home		$\omega_{1v} = 1.45$	autism		$\omega_{1v} = -1.52$
measles		$\omega_{1v} = 1.44$	#cdcfraud		$\omega_{1v} = -1.52$
important		$\omega_{1v} = 1.43$	cause		$\omega_{1v} = -1.48$
well		$\omega_{1v} = 1.38$	#autism		$\omega_{1v} = -1.48$
Topic 4	Topic 2	Topic 24	Topic 17	Topic 7	Topic 18
$r_t = 0.88$	$r_t = 0.77$	$r_t = 0.44$	$r_t = -1.05$	r_t=−0.41	r_t=−0.14
#vaccineswork	measles	flu	#cdcwhistleblower	flu	flu
#vaccines	kids	get	autism	italy	cdc
lives	disneyland	#flu	cdc	deaths	effective
help	#vaccinate	shot	#vaccines	court	may
#vaccination	cases	late	#autism	death	year's
work	people	#vaccine	mmr	child	virus
save	parents	season	#hearthiswell	13	less
great	outbreak	week	link	toll	work
kids	children	influenza	fraud	form	strain
need	unvaccinated	#flushot	#vaccine	rises	says

Table 6.4: Examples of topics learned on the vaccine dataset (50% hashtags, 25 topics). The top row shows the words with highest positive/negative value of ω, indicating common word associations with each perspective. We show examples of three topics associated with each perspective, as defined by the topics' inferred r values. For increased lexical diversity, we excluded words common across many topics in the two datasets ("vacc*") from this output.

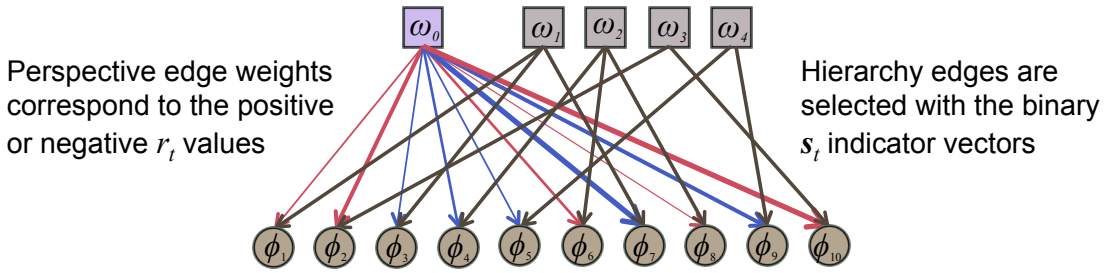


Figure 6.2: An illustration of the relationship between topic components and topic parameters in the joint perspective and topic hierarchy model of Section 6.2. The prior for each topic's word distribution ϕ_t has two parent components: the perspective component ω_0 and one hierarchy component $\omega_{t>0}$. Perspective coefficients can be positive or negative, while hierarchy coefficients are constrained to be positive.

Sections 5.1.1 and 5.2 in Chapter 5, allowing us to demonstrate and evaluate the different structures. Because this joint model includes many of the topic models described in Section 5.2, we can compare to many different baselines by ablating parts of the model to create simpler variants. Thus, the experiments in this section serve as a comprehensive comparison of different baseline models. We experiment with this model on a corpus of political debates (where perspective corresponds to ideology) and a corpus of online doctor reviews (where perspective corresponds to the review sentiment). This section includes additional material from Paul and Dredze (2015), which was also the basis for Chapter 5.

6.2.1 A Factored Hierarchical Model

Our primary extension to the model in Section 6.1.2 is to create a topic *hierarchy*, as described in Section 5.2.2.3. The hierarchy will use components for both topics and documents, where α_m is document m's supertopic proportions, δ_c is the cth supertopic's subtopic prior, ω_c is the cth supertopic's word prior, and β_t is the weight vector that selects

the tth topic's parent supertopic. We constrain each β_t to be a (soft) indicator vector to encode a tree structure (Section 5.1.1.2).

We use a weighted tree: while each β_t has only one nonzero element, the nonzero element can be a value other than 1. We do this by replacing the single coefficient β_{tc} with a product of two variables: $s_{tc}\hat{\beta}_{tc}$. Here, $\hat{\beta}_t$ is a real-valued weight vector, while s_t is a binary indicator vector which zeroes out all but one element of β_t. We do the same with the δ vectors, replacing δ_{ct} with $s_{tc}\hat{\delta}_{ct}$. The $\boldsymbol{s}$ variables are shared across both topic and document components, which is how we tie these together (Section 5.1.2). This is the same approach used in the experiments of Chapter 5 (Section 5.4). As before, we relax the binary requirement for $\boldsymbol{s}$ and instead allow a positive real-valued vector whose elements sum to 1, with a Dirichlet(ρ<1) prior to encourage sparsity. We will also experiment with annealing this prior, as proposed in Section 5.3.2.1.

To be more clearly interpreted as a hierarchy, we constrain the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ (and by extension, $\boldsymbol{\delta}$, since the parameters are tied) to be positive, as suggested in Section 5.1.3. Positive coefficients (rather than negative) give the effect that topics should be similar (not dissimilar) to their parent components, and topics should be more (not less) likely to appear in documents that favor the parent component, which is why we disallow negative coefficients.

Additionally, we *factorize* the hierarchy such that each topic depends not only on its supertopic, but also on a value indicating perspective. The prior for a topic will be a log-linear combination of both a supertopic (e.g., ENERGY) and a perspective (e.g., LIBERAL)

weight vector. As in the perspective model of the previous section, the scalar perspective value for the tth topic is denoted r_t. The $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ variables associated with the perspective component are denoted with superscript (r) rather than subscript c.

To learn meaningful perspective parameters, we include supervision in the form of *document attributes*, as described in Section 5.2.2.4. Each document includes a positive or negative score denoting the perspective, which is the variable $\alpha_m^{(\mathrm{r})}$ for document m. There is only a single perspective component, but it represents two ends of a spectrum with positive and negative weights; these parameters have no positivity constraints, unlike the supertopics.

Finally, we include "bias" component vectors denoted $b^{(\phi)}$ and $b^{(\delta)}$, which act as overall weights over the vocabulary and topics, so that the component-specific ω and δ weights can be interpreted as deviations from the global bias weights, as suggested in Section 5.1.6.

The model hyperparameters are summarized as follows:

- $s_t \sim \text{Dirichlet}(\rho < 1)$ (soft indicator)
- $\alpha_m^{(\mathrm{r})}$ is given as input (perspective value)
- $\tilde{\phi}_{tv} = \exp(b_v^{(\phi)} + r_t \omega_v^{(\mathrm{r})} + \sum_{c=1}^{C} s_{tc} \hat{\beta}_{tc} \omega_{cv})$
- $\tilde{\theta}_{mt} = \exp(b_t^{(\theta)} + \alpha_m^{(\mathrm{r})} r_t + \sum_{c=1}^{C} s_{tc} \alpha_{mc} \hat{\delta}_{ct})$

This includes most of the features described in Chapter 5 (trees, factored structures, tying topic and document components, and document attributes), so we can ablate model features to measure the effects of each.

6.2.2 Experiments

6.2.2.1 Experimental details

Our experiments use the *Debates* and *Reviews* corpora used in the previous chapter, described in Section 5.4. Both of these datasets include perspective values associated with each document, which we use as α_m. In contrast to the lightweight supervision approach used in the previous section (Section 6.1.3), each α_m is provided as input rather than inferred, because the perspective value is known. Specifically:

- *Debates:* Each document is a transcript of one Congressional speaker's turn in a debate, and each document includes the first dimension of the DW-NOMINATE score (Lewis and Poole, 2004), a real-valued score indicating how conservative (positive value) or liberal (negative value) the speaker is.
- *Reviews:* Each document is a review that contains ratings on a 1–5 scale for multiple aspects. We centered the ratings around the middle value 3, then took reviews that had the same sign for all aspects. The α_m value is set as the average score across the aspects.

The experimental setup is identical to that described in Section 5.4.1. Unless otherwise specified, T=50 topics and C=10 components (excluding the perspective and bias components) for *Debates*, and T=20 and C=5 for *Reviews*.

	Debates			*Reviews*		
Model	Perplexity	Prediction error	Coherence	Perplexity	Prediction error	Coherence
Full model	†**1555.5** ± 2.3	†0.615 ± 0.001	-342.8 ± 0.9	†1421.3 ± 8.4	†**0.787** ± 0.006	-512.7 ± 1.6
Hierarchy only	†1561.8 ± 1.4	0.620 ± 0.002	-342.6 ± 1.1	†1457.2 ± 6.9	†0.804 ± 0.007	-509.1 ± 1.9
Perspective only	†1567.3 ± 2.3	†**0.613** ± 0.002	-342.1 ± 1.2	†**1413.7** ± 2.2	†0.800 ± 0.002	-512.0 ± 1.7
SCTM-style	1572.5 ± 1.6	0.620 ± 0.002	†**-335.8** ± 1.1	1504.0 ± 1.9	†0.837 ± 0.002	†**-490.8** ± 0.9
PAM-style	†1567.4 ± 1.9	0.620 ± 0.002	-347.6 ± 1.4	†1440.4 ± 2.7	†0.835 ± 0.004	-542.9 ± 6.7
FLDA-style	†1559.5 ± 2.0	0.617 ± 0.002	-340.8 ± 1.4	†1451.1 ± 5.4	†0.809 ± 0.006	-505.3 ± 2.3
DMR	1578.0 ± 1.1	0.618 ± 0.002	-343.1 ± 1.0	†1416.4 ± 3.0	†0.799 ± 0.003	-511.6 ± 2.0
PAM	1578.9 ± 0.3	0.622 ± 0.003	†-336.0 ± 1.1	1514.8 ± 0.9	†0.835 ± 0.003	†-493.3 ± 1.2
FLDA	1574.1 ± 2.2	0.618 ± 0.002	-344.4 ± 1.3	1541.9 ± 2.3	0.856 ± 0.003	-502.2 ± 3.1
LDA (learned)	1579.6 ± 1.5	0.620 ± 0.001	-342.6 ± 0.6	1507.9 ± 2.4	0.846 ± 0.002	-501.4 ± 1.2
LDA (fixed)	1659.3 ± 0.9	0.622 ± 0.002	-349.5 ± 0.8	1517.2 ± 0.4	0.920 ± 0.003	-585.2 ± 0.9
bag-of-words	2521.6 ± 0.0	0.617 ± 0.000	†-196.2 ± 0.0	1633.5 ± 0.0	0.813 ± 0.000	†-408.1 ± 0.0
Naive baseline	7426.0 ± 0.0	0.677 ± 0.000	-852.9 ± 7.4	10158.0 ± 0.0	1.595 ± 0.000	-795.2 ± 13.0

Table 6.5: Perplexity of held-out tokens and mean absolute error for attribute prediction using various models (± std. error across 10 sampling trials). † indicates significant difference ($p < 0.05$) from optimized LDA under a two-sided t-test.

6.2.2.2 Quantitative evaluation

We measured perplexity and topic coherence as done in the previous chapter, described in Section 5.4.2. Additionally, we also evaluated how well the model can predict the attribute values (DW-NOMINATE score or user rating) of held-out documents. We trained a linear regression model using the document topic distributions θ_m as features. We held out half of the documents for testing and measured the mean absolute error. For the test documents, the topic models do not use the attribute values as priors, and instead α_m is set to 0.

Using these three metrics, we compared to several variants (denoted in bold) of the **full model** to understand how the different parts of the model affect performance:

- Variants that contain the hierarchy components but not the perspective component (**Hierarchy only**), and vice versa (**Perspective only**).
- The "hierarchy only" model using only document components $\boldsymbol{\delta}$ and no topic compo-

nents $\boldsymbol{\omega}$. This is a **PAM-style** model because it exhibits similar behavior to PAM (Section 5.2.2.3), with no structured priors for the word distributions of topics. We also compared to the original **PAM** model.

- The "hierarchy only" model using only topic components $\boldsymbol{\omega}$ and no document components $\boldsymbol{\delta}$. This is a **SCTM-style** model because it exhibits similar behavior to SCTM (Section 5.2.2.2), with no structured priors for the topic distributions of documents.
- The full model where $\alpha^{(\mathrm{r})}$ is learned (unsupervised) rather than given as input. This is a **FLDA-style** model that has similar behavior to FLDA (Section 5.2.1), since the model learns a factorization of topic and perspective. We also compared to the original **FLDA** model with 2 perspective components.
- The "perspective only" model but without the $\omega^{(\mathrm{r})}$ topic component, so the attribute value affects only the topic distributions and not the word distributions. This is identical to the **DMR** topic model (Section 5.2.2.4).
- A model with no components except for the bias vectors $b^{(\phi)}$ and $b^{(\theta)}$. This is equivalent to **LDA** with optimized hyperparameters **(learned)**. We also experimented with using **fixed** symmetric hyperparameters, using values suggested by Griffiths and Steyvers (2004): $50/T$ and 0.01 for topic and word distributions, respectively.

To put the results in context, we also compare to two types of baselines: (1) "bag-of-words" baselines, where we measure (a) the perplexity of add-one smoothed unigram language models, (b) the prediction error using bag-of-words features,[2] and (c) coherence of the un-

[2]Bag-of-words regression gave extremely poor performance without moderate regularization. For this

igram distribution; (2) naive baselines, where we measure (a) the perplexity of the uniform distribution over each dataset's vocabulary, (b) the prediction error when simply predicting each attribute as the mean value in the training set, and (c) the coherence of 20 randomly selected words (repeated for 10 trials).

Table 6.5 shows that the full SPRITE model substantially outperforms the LDA baseline at both predictive tasks. Generally, model variants with more structure perform better predictively than those with less.

The difference between **SCTM-style** and **PAM-style** is that the former uses only topic components (for word distributions) and the latter uses only document components (for the topic distributions). Results show that the structured priors are more important for topic than word distributions, since PAM-style has lower perplexity on both datasets. However, models with both topic and document components generally outperform either alone, including comparing the **Perspective only** and **DMR** models. The former includes both topic and document perspective components, while DMR has only a document level component.

PAM does not significantly outperform optimized LDA in most measures, likely because it updates the hyperparameters using a moment-based approximation, which is less accurate than our gradient-based optimization. **FLDA** perplexity is 2.3% higher than optimized LDA on *Reviews*, comparable to the 4% reported in Section 3.3.5 on a different corpus. The **FLDA-style** SPRITE variant significantly outperforms FLDA in most measures. This is likely because FLDA has a rigid structure that limits what can be learned:

baseline, we tuned the ℓ_2 regularization coefficient to minimize the resulting prediction error. With the topic model features, we used minimal, untuned regularization (a coefficient of 10^{-5}) simply to keep the weights from going to extreme values.

each topic corresponds to a unique (unweighted) pairing of components from the two factors; the SPRITE approach allows multiple topics to share the same components, and to combine them with different weights.

The results are quite different under the coherence metric. It seems that topic components (which influence the word distributions) improve coherence over LDA, while document components worsen coherence. This was also observed in Section 5.4.2. SCTM-style (which uses only topic components) does the best in both datasets, while PAM-style (which uses only documents) does the worst. PAM also significantly improves coherence over LDA, despite worse perplexity.

The **LDA (learned)** baseline substantially outperforms **LDA (fixed)** in all cases, highlighting the importance of optimizing hyperparameters, consistent with prior research (Wallach et al., 2009a).

Surprisingly, many SPRITE variants also outperform the bag-of-words regression baseline, even though the latter was tuned to optimize performance using heavy ℓ_2 regularization, which we applied only weakly (without tuning) to the topic model features. We also point out that the "bag-of-words" version of the coherence metric (the coherence of the top 20 words) is higher than the average topic coherence, which seems to be an artifact of how the metric is defined: the most probable words in the corpus also tend to co-occur together in most documents, so these words are considered to be highly coherent together.

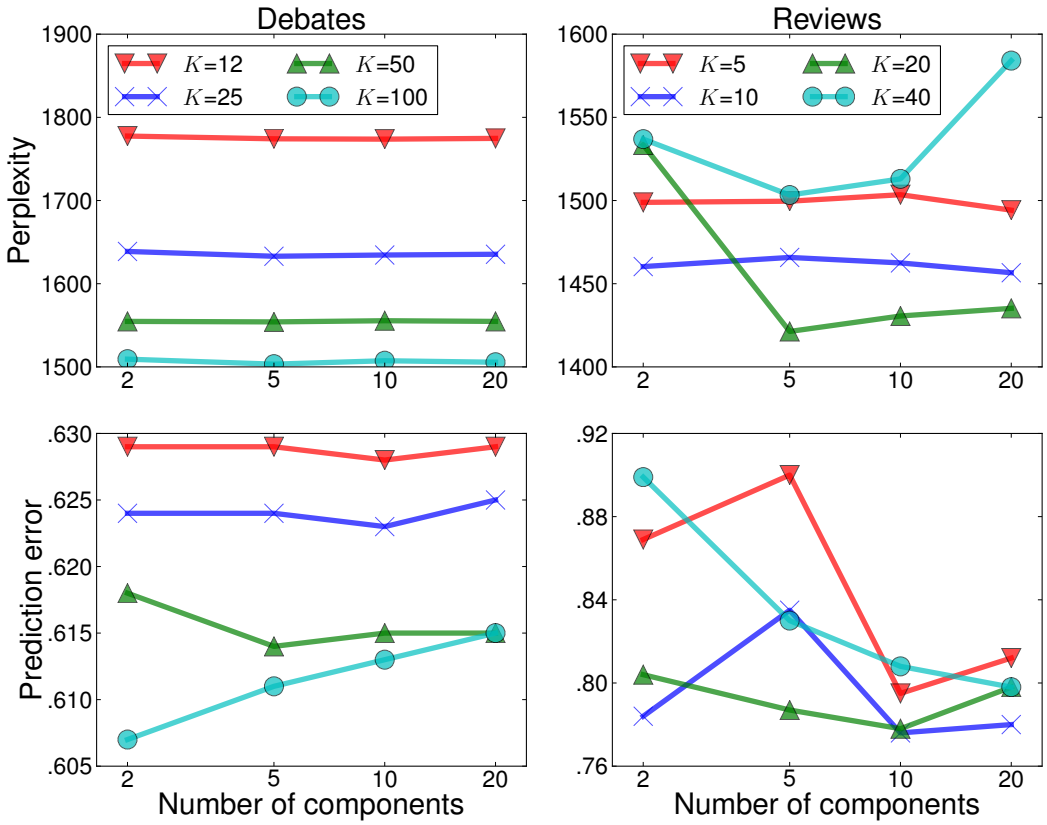


Figure 6.3: Predictive performance of the full model with different numbers of topics T across different numbers of components, represented on the x-axis (log scale).

6.2.2.3 Parameter sensitivity

We evaluated the full model at the two predictive tasks with varying numbers of topics, $T \in \{12, 25, 50, 100\}$ for *Debates* and $T \in \{5, 10, 20, 40\}$ for *Reviews*, and varying numbers of components, $C \in \{2, 5, 10, 20\}$. Figure 6.3 shows that performance is more sensitive to the number of topics than components, with generally less variance among the latter. More topics improve performance monotonically on *Debates*, while performance declines at 40 topics on *Reviews*. The middle range of components (5–10) tends to perform better than too few (2) or too many (20) components.

Regardless of quantitative differences, the choice of parameters may depend on the end application and the particular structures that the user has in mind, if interpretability is important. For example, if the topic model is used as a visualization tool, then 2 components would not likely result in an interesting hierarchy to the user, even if this setting produces low perplexity.

6.2.2.4 Structured sparsity

Recall that we used a relaxation of the binary $\boldsymbol{s}$ that induces a "soft" tree structure, since the sparse Dirichlet prior encourages but does not constrain the values of $\boldsymbol{s}$ to be binary indicator vectors. In Section 5.3.2.1, we suggested it is possible to induce "hard" tree structures by annealing only the prior term in the objective function, which strengthens the preference to be binary.

Table 6.6 shows the percentage of $\boldsymbol{s}$ values which are within $\epsilon = .001$ of 0 or 1 under various annealing schedules, increasing the inverse temperature τ by 0.1% after each iteration (i.e., $\tau_t = 1.001^i$) as well as 0.3% ($\tau_t = 1.003^i$) and no annealing at all ($\tau = 1$). At $\tau = 0$, we model a DAG because the model has no preference that $\boldsymbol{s}$ is sparse. Many of the values are binary in the DAG case, but the sparse prior substantially increases the number of binary values, obtaining fully binary structures with sufficient annealing.

We caution against an annealing schedule that binarizes the values too quickly, because a value that becomes zero can never become nonzero when using the exponentiated gradient updates (Section 5.3).

τ_i	*Debates*	*Reviews*
0.000 (Sparse DAG)	58.1%	42.4%
1.000 (Soft Tree)	93.2%	74.6%
1.001^i (Hard Tree)	99.8%	99.4%
1.003^i (Hard Tree)	100%	100%

Table 6.6: The percentage of indicator values that are sparse (near 0 or 1) when using different annealing schedules.

6.2.2.5 Qualitative analysis

Figure 6.4 shows examples of topics learned from the *Reviews* corpus. The figure includes the highest-probability words in various topics as well as the highest-weight words in the supertopic components and perspective component, which feed into the priors over the topic parameters. We see that one supertopic includes many words related to surgery, such as *procedure* and *performed*, and has multiple children, including a topic about dental work. Another supertopic includes words describing family members such as *kids* and *husband*. One topic has both supertopics as parents, which appears to describe surgeries that saved a family member's life, with top words including {*saved*, *life*, *husband*, *cancer*}. The figure also illustrates which topics are associated more with positive or negative reviews, as indicated by the value of r_t.

Interpretable parameters were also learned from the *Debates* corpus. Consider two topics about energy which have polar values of r_t. The conservative-leaning topic is about oil and gas, with top words including {*oil*, *gas*, *companies*, *prices*, *drilling*}. The liberal-leaning topic is about renewable energy, with top words including {*energy*, *new*, *technology*, *future*, *renewable*}. Both of these topics share a common parent of an industry-related

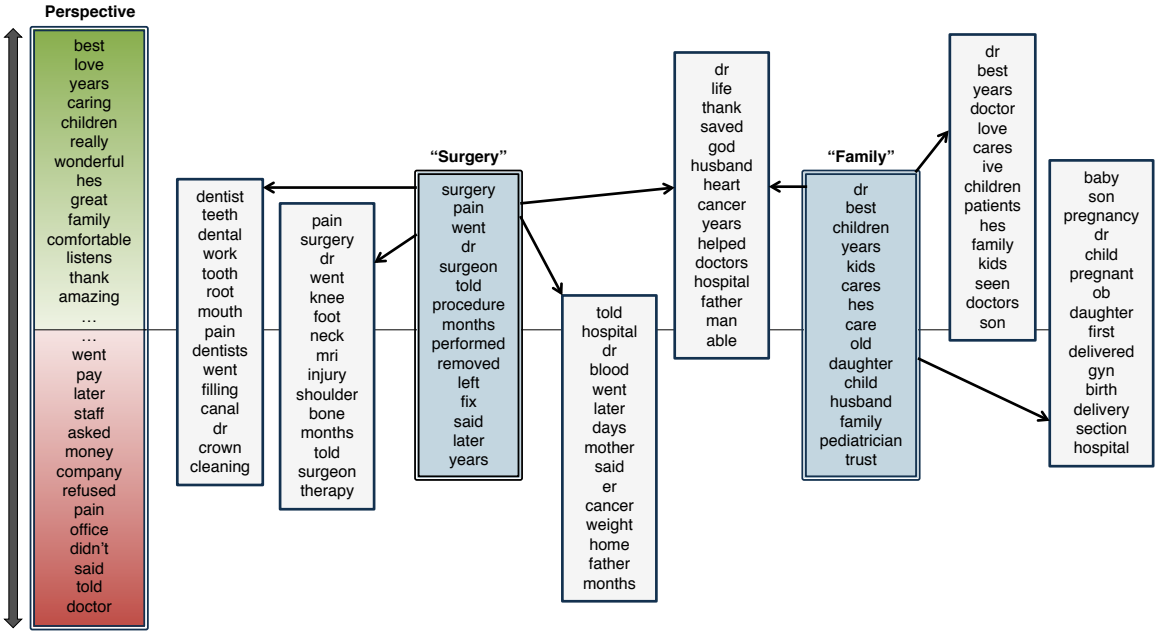


Figure 6.4: Examples of topics (gray boxes) and components (colored boxes) learned on the *Reviews* corpus with 20 topics and 5 components. Words with the highest and lowest values of $\omega^{(\mathrm{r})}$, the perspective component, are shown on the left, reflecting positive and negative sentiment words. The words with largest ω values in two supertopic components are also shown, with manually given labels. Arrows from components to topics indicate that the topic's word distribution draws from that component in its prior (with non-zero β value). There are also implicit arrows from the perspective component to all topics (omitted for clarity). The vertical positions of topics reflect the topic's perspective value r_t. Topics centered above the middle line are more likely to occur in reviews with positive scores, while topics below the middle line are more likely in negative reviews. Note that this is a "soft" hierarchy because the tree structure is not strictly enforced, so some topics have multiple parent components. Table 6.6 shows how strict trees can be learned by tuning the annealing parameter.

supertopic whose top words are {*industry*, *companies*, *market*, *price*}. A nonpartisan topic under this same supertopic has top words {*credit*, *financial*, *loan*, *mortgage*, *loans*}.

6.3 Related Work

Inverse regression topic models (Rabinovich and Blei, 2014) use document feature values (such as political ideology) to alter the parameters of the topic-specific word distributions, similar to what our perspective model does (Section 6.1). This is an alternative to the more common approach to regression-based topic modeling, where the variables typically affect the topic distributions rather than the word distributions. Our SPRITE-based model does both: the document features adjust the prior over topic distributions, but by tying together the document and topic components (by using a shared r variable that influences both $\tilde{\theta}$ and $\tilde{\phi}$) the document features also affect the prior over word distributions. To the best of our knowledge, this is the first topic model to condition both topic and word distributions on the same features.

Our hierarchical perspective model (Section 6.2) is related to supervised hierarchical LDA (SHLDA) (Nguyen et al., 2013), which learns a topic hierarchy while also learning regression parameters to associate topics with feature values such as political perspective. This model does not explicitly incorporate perspective-specific word priors into the topics, as in our factorized approach. The regression structure is also different. SHLDA is a "downstream" model, where the perspective value is a response variable conditioned on

the topics. In contrast, SPRITE is an "upstream" model, where the topics are conditioned on the perspective value. We argue that the latter is more accurate as a generative story (the emitted words depend on the author's perspective, not the other way around). Moreover, in our model the perspective influences both the word and topic distributions (through the topic and document components, respectively).

Encoding perspective along a continuous spectrum, as done in this chapter, is used in *ideal point models* (Lewis and Poole, 2004), which are commonly-used models in political science for measuring ideology of legislatures. A number of topic models have been proposed that combine topics with ideal points (Gerrish and Blei, 2012; Gu et al., 2014; Nguyen et al., 2015b), and the SPRITE models presented in this chapter could also be considered ideal point topic models.

The structural topic model (STM) (Roberts et al., 2013) (described in Section 2.3.3.3) is also related to SPRITE, but using logistic normal priors rather than Dirichlet priors (discussed in Section 5.1.7). STM has been used to summarize responses to surveys, which is related to our goal of survey measurement in this chapter (Roberts et al., 2014).

6.4 Using SPRITE in Practice

We will end this chapter by discussing some practical issues with running SPRITE, in particular by describing how the hyperparameters should be set in order to achieve good performance and desired behavior.

6.4.1 Role of the Hyperparameters

This subsection will describe some rules of thumb for inducing different types of behavior through different settings of the learning hyperparameters which control $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\delta}$, and $\boldsymbol{\omega}$. Most of this advice also applies to Factorial LDA, as a special case of SPRITE. These guidelines are informal, based on the "wisdom" of this author, having spent many months working with SPRITE and FLDA, while the next subsection will show the performance effect of different hyperparameter settings experimentally. We focus on two kinds of learning hyperparameters: the initialization of the parameter values and the regularization of the parameters.

6.4.1.1 Parameter initialization

The initial values of the Dirichlet prior parameters are important for learning good models. If the "pseudocounts" of the Dirichlet priors $\tilde{\theta}$ and $\tilde{\phi}$ are too large relative to the observed counts, then it will be difficult for the Gibbs sampler to learn anything, and it will remain in a state near its initial state. On the other hand, if they are too small, they may have no effect on sampling. A good practice is to examine the Dirichlet precision $\sum_v \tilde{\phi}_{tv}$ and $\sum_t \tilde{\theta}_{mt}$ for the topics and documents, and compare the initial precision to the observed counts, to be sure the initial precision is not so large as to hurt learning. A good rule of thumb is that the sum of the pseudocounts within a topic or document should be on the same order as the sum of the observed counts.

In the SPRITE implementation used for this thesis, components weights (δ, ω) are gen-

erally initialized to random values near 0, while coefficients (α, β) are initialized to 1 unless otherwise constrained.

For each topic's distributions over words, a typical symmetric prior is $\tilde{\phi}_{tv} = 0.01$ (Griffiths and Steyvers, 2004), and we have found this to be a good approximate value for initialization for each t, v. This means that each β_{tc} and ω_{cv} should be initialized such that $\exp(\sum_c \beta_{tc}\omega_{cv}) \approx 0.01$. This is typically done by setting a bias component (Section 5.1.6) with values $\omega_v \approx \log(0.01)$, while initializing all other component weights to be near 0 (with noise). When the β coefficients are constrained to the simplex, they are initialized to be uniform, $\beta_c = \frac{1}{C}$.

When informed asymmetric priors are used to incorporate prior knowledge, as in Chapter 4 when the ω weights are initialized to values inferred from labeled data, then it may be desirable to have a stronger prior, in which case ω values can be initialized such that the expected value of $\tilde{\phi}_{tv}$ is larger than 0.01. Another useful technique when using an informed asymmetric prior is to initialize the Gibbs sampler assignments according to the prior, either by setting each token's topic assignment to the word's most likely topic under the prior, or sampling the initial topic assignment according to the prior.

As for each document's distribution over topics, values are typically initialized such that $\tilde{\theta}_{mt} \approx 0.1$ or smaller in this thesis. While many researchers commonly use the heuristic of $\tilde{\theta}_t = \frac{50}{T}$ (Griffiths and Steyvers, 2004), this often produces pseudocounts that are too large for short documents. For example, a tweet may only have 5 tokens, so if there are 50 topics with $\tilde{\theta}_t = 1.0$, then the pseudocounts from the prior will sum to 50, washing out

$\sigma^2 = 0.01$		$\sigma^2 = 1$		$\sigma^2 = 100$	
Positive	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**
excellent	records	great	doctor	dr	doctor
wonderful	money	caring	rude	great	told
fantastic	worse	wonderful	refused	best	said
impressed	mistake	best	worst	highly	office
amazing	worst	helped	told	knowledgeable	poor
explains	horrible	knowledgeable	avoid	caring	refused
compassionate	dismissed	amazing	stay	excellent	money
exceptional	refused	kind	instead	wonderful	rude
expertise	mistakes	friendly	cold	friendly	drug
supportive	despite	excellent	worse	listens	prescribed
awesome	terrible	knowledgable	beware	love	unprofessional
attentive	tears	thorough	lost	thorough	later
fortunate	caused	dr	records	really	records
talented	failed	listens	allowed	kind	blood
knowledgable	tried	fantastic	horrible	compassionate	pain

Table 6.7: Examples of priors learned with different degrees of regularization on the doctor reviews corpus using the perspective and hierarchy model in Section 6.2.1. The words shown are the top 15 words with the highest and lowest weights in the perspective component vector $\omega^{(\mathrm{r})}$, corresponding to positive and negative sentiment in reviews, learned with three different values for the variance σ^2 of the normal prior over the ω weights.

the observed counts, and thus making learning difficult. Therefore, care should be given to initialize the document priors to sufficiently small values, particularly when working with short documents.

The document priors can be initialized analogously to the topic priors, using a bias component initialized with $\delta_t \approx \log(0.1)$ and other component weights randomly initialized near 0. When components are constrained to be positive, rather than initializing to near 0, they are initialized to be near a small value, such as $\exp(-2)$ in this implementation.

6.4.1.2 Parameter regularization

In addition to good initialization, it is important to have good regularization to control how the parameters will stray from their initial values during learning. Regularization is controlled by setting the variance of the normal priors over α, β, δ, and ω. Regularization serves two primary functions: maintaining stable values during learning, and learning values that are more generalizable and more interpretable.

Above, we explained that it is important to initialize the priors such that the Dirichlet precision is not so high that the "pseudocounts" overwhelm the observed counts. One should also be careful that the precision does not become too large during learning. There is a risk that the precision can become too large during learning: as the precision increases, the samples collected will have lower variance, which in turn may cause the gradient ascent optimizer to further increase the precision. This issue can be combatted by biasing the parameters to small values (that is, regularization).

While the default regularization priors are 0-mean normal distributions, it is not necessarily desirable to bias every parameter toward 0, as $\exp(0)$ is much larger than the values suggested above (0.01 and 0.1). An approach that we found to be successful was to use only weak regularization (high variance) for the weights of bias components, so that they are not heavily pushed toward 0 and can instead remain near their initial values, while applying heavier (low variance) regularization toward 0 for the other document-specific and topic-specific components. Another approach is to use the parameter's initial value as the mean of the normal prior rather than 0, to encourage the parameters to stay near their

initial setting, particularly for the bias components. This is likely to be more effective at keeping the Dirichlet precision at desired levels, though this approach was not used in the experiments in this thesis.

Biasing the component weights toward zero can also be important for interpretability, particularly for the ω values, which are an important part of SPRITE, as in Table 6.3 and Figure 6.4. For example, Table 6.7 shows examples of ω component weights corresponding to perspective in reviews learned with strong, moderate, and weak regularization, and the top words in the component are fairly different across the three versions. The component learned with strong regularization contains many more top words that are clearly associated with positive and negative sentiment, whereas the component with weak regularization contains many generic words that do not have strong sentiment associations, such as "really" and "told". However, the versions with moderate and weak regularization give high weight to important sentiment words such as "best" and "great" that do not appear at all in the top 15 words in the version with strong regularization.

In general, we have found that values of 1 or lower work well for the variance of the topic-specific ω values, while a variance of 100 or higher works well for the bias ω component. Unfortunately, it is difficult to tune this for interpretability without manual inspection with trial and error. While the interpretability of topic word distributions can be approximated with automated coherence metrics (Section 2.6.2), it is an open question whether similar metrics are useful for measuring interpretability of the priors.

	Perplexity			Coherence		
Hyperparameter	Stddev	KS p	Best	Stddev	KS p	Best
Initial δ bias	176.9	$<.001$	-1.0	4.5	.119	-2.0
Initial ω bias	75.3	.026	-2.0	26.6	$<.001$	-1.0
δ variance	33.9	.692	0.01	4.6	.032	0.01
ω variance	26.6	.786	0.01	3.6	.376	0.25

Table 6.8: Measuring the effect of different hyperparameter settings. For each hyperparameter, the table shows the standard deviation of the mean perplexity and coherence for sampling trials using five different settings of the hyperparameter and the p-value of a Kolmogorov-Smirnov (KS) test comparing the samples from the best hyperparameter setting to the worst, as well as the hyperparameter setting with the best result.

6.4.2 Hyperparameter Effect on Performance

We now experimentally compare different settings of the hyperparameters discussed in the previous subsection: the initialization and regularization of the component weights. We evaluated the full SPRITE model from Section 6.1 on the various Twitter datasets. We compared settings of the initial value of the bias components, for the possible values $\{-5.0, -3.0, -2.0, -1.0, 0.0\}$, as well as the variance of the normal prior for the δ and ω weights, for the possible values $\{.0001, .01, .25, 1.0, 100.0\}$. We ran Gibbs samplers with various combinations of these settings, with 294 sampling trials in total across the datasets.

For each sampling trial, we computed held-out perplexity and coherence using the same approach as in Section 6.1.4. Thus, each hyperparameter setting has multiple perplexity and coherence scores, from which we computed the mean score. Each hyperparameter thus has five mean scores (one for each of the five hyperparameter values) for each of the two metrics. To measure how sensitive the perplexity and coherence scores are to the setting of each hyperparameter, we measured the standard deviation of the scores across the five

mean scores, as well as a significance test to measure whether the highest mean score is significantly different from the lowest mean score for that hyperparameter.

To measure significance, we used a Kolmogorov-Smirnov (KS) test, which is a statistical test for whether two sets of samples come from the same distribution. In this case, the samples for a particular hyperparameter setting are the scores from the multiple Gibbs sampling trials with that setting, and we compare the scores of the setting that achieved the highest mean score with the scores from the setting with the lowest mean score. If the p-value of the KS test is .05, then we say with 95% confidence that the samples come from different distributions.

Table 6.8 shows the standard deviation across the five settings for each hyperparameter and metric along with the p-value of the KS test between the best and worst setting. Additionally, the table shows the hyperparameter setting that achieved the best mean score.

These results show that both perplexity and coherence are quite sensitive to the initialization of the bias components, while performance is less sensitive to the regularization of the components. The regularization of δ has a significant effect on coherence but not perplexity in these experiments. Interestingly, regularization of ω does not have much effect on coherence, even though ω influences the word distributions, which is what coherence measures. However, as shown in the previous section, the regularization of ω can in fact have an important effect on the interpretability of ω itself, which is not captured by the topic coherence metric.

We note that we also measured the effect of other important hyperparameters—the

number of topics and the number of components—earlier in this chapter, with results shown in Figure 6.3. We found that predictiveness of SPRITE seems to be more sensitive to the number of topics than the number of components. From the results, a general rule of thumb is that the number of components should be smaller than the number of topics.

There exist methods for automatically optimizing hyperparameters, finding hyperparameter values that give (locally) optimal results under some evaluation metric (e.g., perplexity or coherence as in Table 6.8) (Snoek et al., 2012; Maclaurin et al., 2015). While such approaches require running the Gibbs sampler many times with different hyperparameter settings, it can be considerably faster than a full grid search of hyperparameter values. However, how well this approach would work depends on the shape of the function (e.g., perplexity as a function of the hyperparameters). If the function is not smooth, then an optimizer may find a poor local optimum.

Another choice that a user must make is the selection of structure in SPRITE (e.g., DAG, tree, or factored). We compared the performance when using different structures in Section 5.4.2. As with the hyperparameters, the choice of structure could potentially be selected automatically during optimization. For example, Grosse et al. (2012) presented a generalization of many different model structures (as we do with SPRITE) along with a search algorithm for choosing among the possible structures.

6.5 Summary

This chapter served two primary purposes. One was to provide a more comprehensive evaluation of SPRITE than the small of experiments provided in Chapter 5. The other was to illustrate how specific SPRITE instantiations can be created for specific purposes.

In Section 6.1, we introduced a joint model of topic and perspective based on the SPRITE framework, incorporating both direct and distant supervision to learn meaningful topics. Our predictive experiments demonstrate improvements in model quality when incorporating lightweight supervision, and we gave examples of intuitive topics. The model includes many interpretable parameters that allow one to examine associations between perspective and topics, perspective and words, and perspective and documents. This provides a richer representation of perspective than unstructured topic models, making the model well-suited for exploratory health and social science analysis.

In Section 6.2, we demonstrated the utility and versatility of SPRITE by constructing a single model with many different characteristics, including a topic hierarchy, a factorization of topic and perspective, and supervision in the form of document attributes. Our experiments explored how each of these various model features affect performance, and our results showed that models with structured priors perform better than LDA.

We additionally presented some practical advice on using SPRITE in Section 6.4, including experimental comparison of different hyperparameter settings. We found that both initialization and regularization can have effects on performance and interpretability, with initialization being particularly important for quantitative performance.

Chapter 7

Conclusion

We conclude by summarizing the contributions of each chapter and proposing future research directions.

Chapter 2 provided a broad array of background material on probabilistic topic modeling, including frequentist and Bayesian topic modeling, structured topic modeling, learning and inference including scalability, and evaluation. This chapter could serve as a standalone guide to topic modeling, while also laying the foundation for the remainder of the thesis, focusing on the algorithms and structures that are used later. An additional contribution of this chapter was to make observations on similarities between Gibbs sampling, especially as used in practice, and stochastic optimization. This connection would be interesting to examine more in future work. In particular, it is difficult to disentangle the different learning goals—inferring posterior means versus searching for MAP estimates—and the algorithm used for that goal, and this would be good to explore more experimentally.

Chapter 3 introduced factorial LDA (FLDA), a novel topic model that organized word distributions into a multi-dimensional structure. This chapter made multiple technical contributions. One contribution was to introduce the concept of and the term "multi-dimensional topic modeling". While Section 2.4.2 surveyed other multi-dimensional topic models that were introduced before FLDA, they were largely tailored to specific applications and limited to two dimensions. Factorial LDA generalized these concepts.

An important contribution was the use of structured priors to let the word distributions reflect the multi-dimensional structure. Previous multi-dimensional topic models like the topic aspect model (TAM) (Paul and Girju, 2010b) had no mechanism to ensure the word distributions reflected the different components. By modeling this structure through priors, we obtain word distributions that are influenced by the different components, unlike in TAM, but with additional parameters beyond the independent combination of the components, unlike in shared components topic models (Gormley et al., 2010). The structured Dirichlet prior is powerful because the Dirichlet precision can adjust the degree to which the prior behaves like either of these two extremes.

A third contribution is the inclusion of a sparsity pattern that can learn which subset of the Cartesian product of tuples is actually supported by data. We incorporated this sparsity pattern into the structured priors using a formulation that allows for efficient inference. We approximated binary constraints by using sparse Beta priors over real-valued variables.

A limitation of FLDA is that all word distributions are associated with all K factors. It could be beneficial to have some word distributions that are associated with a smaller

number of factors. That is, in addition to K-tuples, word distributions could be associated with 1-, 2-, ..., $(K-1)$-tuples. This could be modeled with SPRITE (Chapter 5), by multiplying the $\boldsymbol{\omega}$ components with binary coefficients to select a subset of factors for each word distribution.

Chapter 4 demonstrated the use of FLDA for two health science applications: summarizing information about drugs from web forums and measuring healthcare quality from patient reviews. The purpose of this chapter was to showcase real problems that FLDA can be applied to and to provide further evaluation of the model. The primary technical contribution of this chapter was to introduce a framework for incorporating domain knowledge into FLDA as priors over the component parameters. This was done by training a simplified supervised model on labeled data, then using those parameters as normal priors for the FLDA parameters. A good future research direction is to consider alternatives to this pipeline approach of using the parameters of one model as priors for another. A more satisfying solution would jointly model the labeled domain data with the unlabeled corpus.

Chapter 5 introduced SPRITE, a family of **s**tructured-**pri**or **t**opic mod**e**ls. SPRITE extends the idea of structured priors used for FLDA, in which priors over topic and word distributions are log-linear functions of underlying components. This results in a shared structure across topics and documents that can be used to model interesting relations between topics and to provide more interpretable output to practitioners.

While FLDA combined components in a particular way for a particular purpose, this chapter showed that components can be combined in other interesting ways. We showed

how different types of structures—a factorization structure as in FLDA, or a tree structure for organizing topics into a hierarchy—can be modeled by placing various constraints on the ways in which components can be combined. We offered a number of suggestions for possible approaches within the SPRITE framework.

We also showed that SPRITE is closely related to many of the structured topic models described in Chapter 2 (Sections and 2.3.3 and 2.4): shared components topic models (Gormley et al., 2010), sparse additive generative models (SAGE) (Eisenstein et al., 2011), Pachinko allocation (Li and McCallum, 2006), and Dirichlet-multinomial regression (Mimno and McCallum, 2008). Not only can SPRITE model similar characteristics as these existing models within a single framework, we also showed that it can extend these models in interesting ways. We focused especially on how FLDA is a special case of SPRITE and how the SPRITE framework can be used to extend FLDA to less rigid structures.

A technical challenge is that adding constraints to the component parameters creates difficulties for parameter optimization. We addressed this by generalizing the technique of using sparsity-inducing Beta priors for the sparse FLDA variant in Chapter 3. To learn binary indicator vectors, we constrained the vectors to the simplex (which is an easier constraint to learn under than requiring values to be binary) and applied sparse Dirichlet priors to prefer values that are close to binary. We also showed that these soft preferences can be turned into hard constraints in a computationally easy way using annealing techniques, which is an improvement over the approach in Chapter 3.

A future research direction is to consider bringing SPRITE's log-linear formulation of

combining components to the generation of multinomial parameters directly rather than Dirichlet parameters, which is how SAGE is formulated (Eisenstein et al., 2011). While we explained in Section 5.1.5 that SPRITE's approach using priors provides more flexibility, these different approaches have not been compared empirically.

Chapter 6 showed how two SPRITE-based models can be used to model perspective in text. The first model used a single component to reflect information about perspective. The second model included additional components to include a topic hierarchy as well as a factorization of topic and perspective, demonstrating multiple structures described in Chapter 5. We applied the models to a diverse set of opinionated corpora, including political debate transcripts, online reviews, and social media messages. Like with Chapter 4, the primary purpose of this chapter was to showcase specific ways that SPRITE can be used and to provide additional evaluation. Another contribution of this chapter was to demonstrate that even lightweight forms of supervision—Twitter hashtags and survey statistics, rather than data annotation—can serve as useful prior knowledge in topic models.

We have shown throughout this thesis that topic models can be powerful aids for analyzing large collections of text. Yet people often have expectations about topics in a given corpus and how they should be structured for a particular task. Research has shown that it is crucial for the user experience that topics meet these expectations (Mimno et al., 2011; Talley et al., 2011). In this thesis, we worked toward addressing this problem by introducing models that provide richer structures than standard topic models and allow for the incorpo-

ration of prior knowledge. Our models provide more control over the types of topics that are learned, and our structured-prior approach offers a unified and powerful framework for a variety of structured topic modeling tasks.

Moving forward, a key challenge will be to make models like SPRITE easier to tune, evaluate, and understand; and in particular, to make them more accessible to practitioners in various scientific domains. This could be addressed in part by using SPRITE as the underlying model for a user interface that visualizes text corpora (Chaney and Blei, 2012; Snyder et al., 2013). A user interface could allow practitioners to explore topics, documents, and the relationships between them (as defined by components) in more depth. Such an interface could also be combined with *interactive* learning of the model (Hu et al., 2011), in which users can repeatedly specify preferences and constraints during the learning process. In standard topic models, interactivity is useful for specifying which words should be grouped together. SPRITE includes additional structure that would warrant more interactivity. For example, users might interactively specify that certain topics should share parent components. Incorporating prior knowledge, such as with survey statistics (Section 6.1.1.1.2) or seed words (Section 4.2.4.2.3), could also be done interactively through an interface, and users could see how different types of supervision affect the topics that are learned. Given that many of the modeling choices in this thesis were motivated by human interpretability, creating a tool to allow users to fully visualize and explore these models is an important next step in bringing this research into practice.

Bibliography

Ahmed, A. and Xing, E. (2010). Staying informed: supervised and semi-supervised multi-view topical analysis of ideological perspective. In *EMNLP*.

Allan, J., Papka, R., and Lavrenko, V. (1998). On-line new event detection and tracking. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Andrzejewski, D., Zhu, X., and Craven, M. (2009). Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *ICML*.

Asuncion, A., Welling, M., Smyth, P., and Teh, Y. W. (2009). On smoothing and inference for topic models. In *Uncertainty in Artificial Intelligence (UAI)*.

Balasubramanyan, R. and Cohen, W. (2013). Regularization of latent variable models to obtain sparsity. In *SIAM Conference on Data Mining*.

Benton, A., Paul, M. J., Hancock, B., and Dredze, M. (2015). A structured model of topic and perspective in social media. In preparation.

BIBLIOGRAPHY

Bilmes, J. A. and Kirchhoff, K. (2003). Factored language models and generalized parallel backoff. In *NAACL*.

Blei, D. (2013). Topic modeling and digital humanities. *Journal of Digital Humanities*, 2(1).

Blei, D., Griffiths, T., Jordan, M., and Tenenbaum, J. (2003a). Hierarchical topic models and the nested Chinese restaurant process. In *NIPS*.

Blei, D. and Lafferty, J. (2007). A correlated topic model of Science. *Annals of Applied Statistics*, 1(1):17–35.

Blei, D., Ng, A., and Jordan, M. (2003b). Latent Dirichlet allocation. *Journal of Machine Learning Research (JMLR)*.

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

Boyd-Graber, J. and Blei, D. (2008). Syntactic topic models. In *NIPS*.

Brody, S. and Elhadad, N. (2010). Detecting salient aspects in online reviews of health providers. In *AMIA*.

Bruneau, J., Roy, É., Arruda, N., Zang, G., and Jutras-Aswad, D. (2012). The rising prevalence of prescription opioid injection and its association with hepatitis c incidence among street-drug users. *Addiction*.

BIBLIOGRAPHY

Canini, K. R., Shi, L., and Griffiths, T. L. (2009). Online inference of topics with latent Dirichlet allocation. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 65–72.

Chamlertwat, W., Bhattarakosol, P., Rungkasiri, T., and Haruechaiyasak, C. (2012). Discovering consumer insight from Twitter via sentiment analysis. *J. UCS*, 18(8):973–992.

Chaney, A. and Blei, D. (2012). Visualizing topic models. In *ICWSM*.

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., and Blei, D. (2009). Reading tea leaves: How humans interpret topic models. In *NIPS*.

Chuang, J., Roberts, M., Stewart, B., Weiss, R., Tingley, D., Grimmer, J., and Heer, J. (2015). TopicCheck: Interactive alignment for assessing topic model stability. In *NAACL-HLT*.

Cohen, R., Aviram, I., Elhadad, M., and Elhadad, N. (2014). Redundancy-aware topic modeling for patient record notes. *PLoS ONE*, 9(2):e87555.

Collins, M. (2002). Discriminative training methods for hidden Markov models: Theory and experiments with Perceptron algorithms. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Conover, M. D., Gonçalves, B., Ratkiewicz, J., Flammini, A., and Menczer, F. (2011). Predicting the political alignment of Twitter users. In *IEEE Third Inernational Conference on Social Computing (SocialCom)*, pages 192–199.

BIBLIOGRAPHY

Corazza, O., Schifano, F., Farre, M., Deluca, P., Davey, Z., Drummond, C., Torrens, M., Demetrovics, Z., Di Furia, L., Flesland, L., et al. (2011). Designer drugs on the Internet: a phenomenon out-of-control? The emergence of hallucinogenic drug Bromo-Dragonfly. *Current Clinical Pharmacology*, 6(2):125–129.

Corazza, O., Schifano, F., Simonato, P., Fergus, S., Assi, S., Stair, J., Corkery, J., Trincas, G., Deluca, P., Davey, Z., Blaszko, U., Demetrovics, Z., Moskalewicz, J., Enea, A., di Melchiorre, G., Mervo, B., di Furia, L., Farre, M., Flesland, L., Pasinetti, M., Pezzolesi, C., Pisarska, A., Shapiro, H., Siemann, H., Skutle, A., Enea, A., di Melchiorre, G., Sferrazza, E., Torrens, M., van der Kreeft, P., Zummo, D., and Scherbaum, N. (2012). Phenomenon of new drugs on the Internet: the case of ketamine derivative methoxetamine. *Human Psychopharmacology: Clinical and Experimental*, 27(2):145–149.

Darling, W. M., Paul, M. J., and Song, F. (2012). Unsupervised part-of-speech tagging in noisy and esoteric domains with a syntactic-semantic Bayesian HMM. In *Proceedings of the Workshop on Semantic Analysis in Social Media*.

Dasgupta, S. and Ng, V. (2010). Mining clustering dimensions. In *ICML*.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from in-

complete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Doucet, A., De Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo Methods in Practice*.

Dredze, M., Paul, M., Bergsma, S., and Tran, H. (2013). Carmen: A Twitter geolocation system with applications to public health. In *AAAI HIAI Workshop*.

Dredze, M., Wallach, H. M., Puller, D., and Pereira, F. (2008). Generating summary keywords for emails using topics. In *13th International Conference on Intelligent User Interfaces (IUI)*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159.

Dunn, M., Bruno, R., Burns, L., and Roxburgh, A. (2011). Effectiveness of and challenges faced by surveillance systems. *Drug Testing and Analysis*, 3(9):635–641.

Eisenstein, J., Ahmed, A., and Xing, E. P. (2011). Sparse additive generative models of text. In *ICML*.

Eisenstein, J., Chau, D. H. P., Kittur, A., and Xing, E. P. (2012). Topicviz: Semantic navigation of document collections. In *CHI Work-in-Progress Paper*.

Eisenstein, J., O'Connor, B., Smith, N. A., and Xing, E. P. (2010). A latent variable model for geographic lexical variation. In *EMNLP*.

BIBLIOGRAPHY

Ellimoottil, C., Hart, A., Greco, K., Quek, M. L., and Farooq, A. (2012). Online reviews of 500 urologists. *The Journal of Urology*.

EMCDDA (2012). 2012 annual report on the state of the drugs problem in Europe. *European Monitoring Centre for Drugs and Drug Addiction, Lisbon*.

Emmert, M., Sander, U., and Pisch, F. (2013). Eight questions about physician-rating websites: A systematic review. *Journal of Medical Internet Research*, 15(2):e24.

Fenton, J. J., Jerant, A. F., Bertakis, K. D., and Franks, P. (2012). The cost of satisfaction: a national study of patient satisfaction, health care utilization, expenditures, and mortality. *Arch. Intern. Med.*, 172(5):405–411.

Fox, S. and Duggan, M. (2013). Health online 2013. Technical report, Pew Internet and American Life Project.

Freund, Y. and Schapire, R. E. (1999). Large margin classification using the Perceptron algorithm. *Machine Learning*, 37(3):277–296.

Galizzi, M. M., Miraldo, M., Stavropoulou, C., Desai, M., Jayatunga, W., Joshi, M., and Parikh, S. (2012). Who is more likely to use doctor-rating websites, and why? A cross-sectional study in London. *BMJ open*, 2(6).

Gallagher, C. T., Assi, S., Stair, J. L., Fergus, S., Corazza, O., Corkery, J. M., and Schifano, F. (2012). 5,6-methylenedioxy-2-aminoindane: from laboratory curiosity to 'legal high'. *Human Psychopharmacology: Clinical and Experimental*, 27(2):106–112.

BIBLIOGRAPHY

Ganchev, K., Graça, J. a., Gillenwater, J., and Taskar, B. (2010). Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.*, 11:2001–2049.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741.

Gerrish, S. and Blei, D. (2012). How they vote: Issue-adjusted models of legislative behavior. In *Advances in Neural Information Processing Systems (NIPS)*.

Ghahramani, Z. and Jordan, M. I. (1997). Factorial hidden Markov models. *Machine Learning*, 29(2-3):245–273.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1995). *Markov Chain Monte Carlo in Practice*. CRC Press.

Goldstein, J., Mittal, V., Carbonell, J., and Kantrowitz, M. (2000). Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization*, pages 40–48.

Goodman, D. C., Fisher, E. S., and Chang, C.-H. (2011). After hospitalization: A Dartmouth Atlas report on post-acute care for medicare beneficiaries.

Gormley, M., Dredze, M., Van Durme, B., and Eisner, J. (2010). Shared components topic models. In *NAACL*.

Greaves, F., Ramirez-Cano, D., Millett, C., Darzi, A., and Donaldson, L. (2013). Harness-

ing the cloud of patient experience: using social media to detect poor quality healthcare. *BMJ Quality & Safety*, 22(3):251–255.

Griffiths, T. and Ghahramani, Z. (2006). Infinite latent feature models and the Indian buffet process. In *NIPS*.

Griffiths, T. and Steyvers, M. (2004). Finding scientific topics. In *Proceedings of the National Academy of Sciences of the United States of America*.

Griffiths, T. L., Steyvers, M., Blei, D. M., and Tenenbaum, J. B. (2005). Integrating topics and syntax. In *Advances in Neural Information Processing Systems (NIPS)*, pages 537–544.

Grosse, R. B., Salakhutdinov, R., Freeman, W. T., and Tenenbaum, J. B. (2012). Exploiting compositionality to explore a large space of model structures. In *Uncertainty in Artificial Intelligence*.

Gu, Y., Sun, Y., Jiang, N., Wang, B., and Chen, T. (2014). Topic-factorized ideal point estimation model for legislative voting network. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 183–192.

Guimaraes, P. and Lindrooth, R. (2005). Dirichlet-multinomial regression. *Economics Working Paper Archive at WUSTL, Econometrics*, (0509001).

Haghighi, A. and Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *NAACL '09: Proceedings of Human Language Technologies: The*

2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 362–370.

Hall, D., Jurafsky, D., and Manning, C. D. (2008). Studying the history of ideas using topic models. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Hastings, W. (1970). Monte Carlo samping methods using Markov chains and their applications. *Biometrika*, 57:97–109.

Hill, S. L. and Thomas, S. H. L. (2011). Clinical toxicology of newer recreational drugs. *Clinical Toxicology*, 49(8):705–719.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14:1771–1800.

Hoffman, M., Blei, D., and Bach, F. (2010). Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *UAI*.

Hout, M. C. V. and Bingham, T. (2012). Costly turn on: Patterns of use and perceived consequences of mephedrone based head shop products amongst Irish injectors. *International Journal of Drug Policy*.

Hu, N., Zhang, J., and Pavlou, P. A. (2009). Overcoming the J-shaped distribution of product reviews. *Commun. ACM*, 52(10):144–147.

BIBLIOGRAPHY

Hu, Y., Boyd-Graber, J., and Satinoff, B. (2011). Interactive topic modeling. In *Association for Computational Linguistics*.

Huang, J., Zhang, T., and Metaxas, D. (2009). Learning with structured sparsity. In *International Conference on Machine Learning (ICML)*, pages 417–424.

Jagarlamudi, J., III, H. D., and Udupa, R. (2012). Incorporating lexical priors into topic models. In *EACL*.

Jain, S. and Neal, R. (2000). A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182.

Jockers, M. L. and Mimno, D. (2013). Signicant themes in 19th-century literature. *Poetics*, Dec.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kivinen, J. and Warmuth, M. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132:1–63.

Lau, H. J., Newman, D., and Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539.

BIBLIOGRAPHY

Lee, D. D. and Seung, H. S. (1997). Unsupervised learning by convex and conic coding. In *Advances in Neural Information Processing Systems (NIPS)*.

Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*.

Lewis, J. and Poole, K. (2004). Measuring bias and uncertainty in ideal point estimates via the parametric bootstrap. *Political Analysis*, 12(2):105–127.

Li, A. Q., Ahmed, A., Ravi, S., and Smola, A. J. (2014). Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 891–900.

Li, W. and McCallum, A. (2006). Pachinko allocation: DAG-structured mixture models of topic correlations. In *International Conference on Machine Learning*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In Marie-Francine Moens, S. S., editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain.

López, A., Detz, A., Ratanawongsa, N., and Sarkar, U. (2012). What patients say about their doctors online: A qualitative content analysis. *Journal of General Internal Medicine*, pages 1–8.

Lu, Y., Zhai, C., and Sundaresan, N. (2009). Rated aspect summarization of short comments. In *WWW*.

BIBLIOGRAPHY

Maclaurin, D., Duvenaud, D., and Adams, R. P. (2015). Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning*.

Martins, A. F. T., Smith, N. A., Aguiar, P. M. Q., and Figueiredo, M. A. T. (2011). Structured sparsity in structured prediction. In *EMNLP*.

May, C., Clemmer, A., and Van Durme, B. (2014). Particle filter rejuvenation and latent Dirichlet allocation. In *ACL*.

McCallum, A. K. (1999). Multi-label text classification with a mixture model trained by EM. In *AAAI Workshop on Text Learning*.

Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C. (2007). Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW*.

Meier, L., van de Geer, S., and Bühlmann, P. (2008). The group lasso for logistic regression. *Journal Of The Royal Statistical Society Series B*, 70(1):53–71.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.

Mimno, D. (2012a). Computational historiography: Data mining in a century of classics journals. *ACM Journal of Computing in Cultural Heritage*, 5(3).

Mimno, D. (2012b). *Topic Regression*. PhD thesis, University of Massachusetts Amherst.

BIBLIOGRAPHY

Mimno, D., Li, W., and McCallum, A. (2007). Mixtures of hierarchical topics with pachinko allocation. In *International Conference on Machine Learning (ICML)*, pages 633–640.

Mimno, D. and McCallum, A. (2008). Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *UAI*.

Mimno, D., Wallach, H., Talley, E., Leenders, M., and McCallum, A. (2011). Optimizing semantic coherence in topic models. In *EMNLP*.

Minka, T. (2003). Estimating a Dirichlet distribution.

Mitchell, T. M. (1980). The need for biases in learning generalizations. In Shavlik, J. W. and Dietterich, T. G., editors, *Readings in Machine Learning*, pages 184–191. Morgan Kauffman.

Morgan, E. M., Snelson, C., and Elison-Bowers, P. (2010). Image and video disclosure of substance use on social media websites. *Computers in Human Behavior*, 26(6):1405–1411. Online Interactivity: Role of Technology in Behavior Change.

Myslin, M., Zhu, S. H., Chapman, W., and Conway, M. (2013). Using Twitter to examine smoking behavior and perceptions of emerging tobacco products. *J. Med. Internet Res.*, 15(8):e174.

Newman, D., Asuncion, A., Smyth, P., and Welling, M. (2009). Distributed algorithms for topic models. *The Journal of Machine Learning Research*, 10:1801–1828.

BIBLIOGRAPHY

Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). 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.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*.

Nguyen, V., Boyd-Graber, J., and Resnik, P. (2013). Lexical and hierarchical topic regression. In *Neural Information Processing Systems*.

Nguyen, V.-A., Boyd-Graber, J., Resnik, P., and Miler, K. (2015a). Tea party in the house: a hierarchical ideal point topic model and its application to Republican legislators in the 112th Congress. In *Association for Computational Linguistics (ACL)*.

Nguyen, V.-A., Boyd-Graber, J., Resnik, P., and Miler, K. (2015b). Tea party in the house: A hierarchical ideal point topic model and its application to Republican legislators in the 112th Congress. In *Association for Computational Linguistics*.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Mach. Learn.*, 39(2-3):103–134.

O'Connor, B., Balasubramanyan, R., Routledge, B. R., and Smith, N. A. (2010). From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*.

BIBLIOGRAPHY

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*.

Paul, M. (2009). Cross-collection topic models: Automatically comparing and contrasting text. Undergraduate thesis, University of Illinois at Urbana-Champaign.

Paul, M. and Dredze, M. (2012a). Factorial LDA: Sparse multi-dimensional text models. In *Neural Information Processing Systems (NIPS)*.

Paul, M. and Dredze, M. (2013). Drug extraction from the web: Summarizing drug experiences with multi-dimensional topic models. In *NAACL*.

Paul, M. and Girju, R. (2009a). Cross-cultural analysis of blogs and forums with mixed-collection topic models. In *EMNLP*, pages 1408–1417.

Paul, M. and Girju, R. (2009b). Topic modeling of research fields: An interdisciplinary perspective. In *International Conference on Recent Advances in Natural Language Processing (RANLP)*.

Paul, M. and Girju, R. (2010a). Summarizing contrastive viewpoints in opinionated text. In *Empirical Methods in Natural Language Processing*.

Paul, M. and Girju, R. (2010b). A two-dimensional topic-aspect model for discovering multi-faceted topics. In *AAAI*.

BIBLIOGRAPHY

Paul, M., Wallace, B., and Dredze, M. (2013). What affects patient (dis)satisfaction? Analyzing online doctor ratings with a joint topic-sentiment model. In *AAAI Workshop on Expanding the Boundaries of Health Informatics Using AI*.

Paul, M. J. (2012). Mixed membership Markov models for unsupervised conversation modeling. In *EMNLP-CoNLL*.

Paul, M. J. and Dredze, M. (2011). You are what you Tweet: Analyzing Twitter for public health. In *5th International AAAI Conference on Weblogs and Social Media (ICWSM)*.

Paul, M. J. and Dredze, M. (2012b). Experimenting with drugs (and topic models): Multidimensional exploration of recreational drug discussions. In *AAAI 2012 Fall Symposium on Information Retrieval and Knowledge Discovery in Biomedical Text*.

Paul, M. J. and Dredze, M. (2014). Discovering health topics in social media using topic models. *PLOS ONE*, 9(8):e103408.

Paul, M. J. and Dredze, M. (2015). SPRITE: Generalizing topic models with structured priors. *Transactions of the Association for Computational Linguistics (TACL)*, 3:43–57.

Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30:377–397.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M. (2008). Fast collapsed gibbs sampling for latent Dirichlet allocation. In *Proceedings of the 14th ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, pages 569–577.

Prier, K. W., Smith, M. S., Giraud-Carrier, C., and Hanson, C. L. (2011). Identifying health-related topics on Twitter: An exploration of tobacco-related tweets as a test topic. In *Proceedings of the 4th International Conference on Social Computing, Behavioral-cultural Modeling and Prediction*, SBP'11, pages 18–25. Springer-Verlag.

(Psychonaut), P. W. R. G. (2009). Bromo-Dragonfly, MDPV, Spice, Mephodrone, and Salvia Divinorum reports. http://www.psychonautproject.eu/technical.php. Institute of Psychiatry, King's College London.

Rabinovich, M. and Blei, D. (2014). The inverse regression topic model. In *International Conference on Machine Learning*.

Raiffa, H. and Schlaifer, R. (1961). *Applied Statistical Decision Theory*. Studies in managerial economics. Division of Research, Graduate School of Business Adminitration, Harvard University.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. (2009). Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*.

Rao, D., Paul, M., Fink, C., Yarowsky, D., Oates, T., and Coppersmith, G. (2011). Hierarchical Bayesian models for latent attribute detection in social media. In *International Conference on Weblogs and Social Media (ICWSM)*.

BIBLIOGRAPHY

Resnik, P., Armstrong, W., Claudino, L., Nguyen, T., Nguyen, V.-A., and Boyd-Graber, J. (2015). Beyond LDA: Exploring supervised topic modeling for depression-related language in Twitter. In *ACL Workshop on Cognitive Modeling and Computational Linguistics*.

Reyes, J., Negrón, J., Colón, H., Padilla, A., Millán, M., Matos, T., and Robles, R. (2012). The emerging of xylazine as a new drug of abuse and its health consequences among drug users in Puerto Rico. *Journal of Urban Health*, pages 1–8.

Robbins, H. (1956). An empirical Bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 157–163. University of California Press.

Roberts, M. E., Stewart, B. M., Tingley, D., and Airoldi, E. M. (2013). The structural topic model and applied social science.

Roberts, M. E., Stewart, B. M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S., Albertson, B., and Rand, D. (2014). Structural topic models for open-ended survey responses. *American Journal of Political Science*, 58(4):1064–1082.

Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *International Conference on Web Search and Data Mining (WSDM)*.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., and Smyth, P. (2004). The author-topic model for authors and documents. In *UAI*.

BIBLIOGRAPHY

Ruppert, D. (1991). Stochastic approximation. In Ghosh, K. and Sen, P. K., editors, *Handbook of Sequential Analysis*. Marcel Dekker, New York.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes Twitter users: real-time event detection by social sensors. In *International Conference on World Wide Web (WWW)*, New York, NY, USA.

Salathe, M. and Khandelwal, S. (2011). Assessing vaccination sentiments with online social media: Implications for infectious disease dynamics and control. *PLoS Comput Biol*, 7(10):e1002199.

SAMHSA (2012). The DAWN report. http://www.samhsa.gov/data/2k12/DAWN105/SR105-synthetic-marijuana.pdf.

Schifano, F., Deluca, P., Baldacchino, A., Peltoniemi, T., Scherbaum, N., and et al. (2006). Drugs on the web: the Psychonaut 2002 EU project. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, 30(4):640 – 646.

Schneider, E. C., Zaslavsky, A. M., Landon, B. E., Lied, T. R., Sheingold, S., and Cleary, P. D. (2001). National quality monitoring of Medicare health plans: the relationship between enrollees' reports and the quality of clinical care. *Med Care*, 39(12):1313–1325.

Segal, J., Sacopulos, M., Sheets, V., Thurston, I., Brooks, K., and Puccia, R. (2012). Online

doctor reviews: Do they track surgeon volume, a proxy for quality of care? *J Med Internet Res*, 14(2).

Sequist, T. D., Schneider, E. C., Anastario, M., Odigie, E. G., Marshall, R., Rogers, W. H., and Safran, D. G. (2008). Quality monitoring of physicians: linking patients' experiences of care to clinical quality and outcomes. *J Gen Intern Med*, 23(11):1784–1790.

Smith, N. A. and Eisner, J. (2004). Annealing techniques for unsupervised statistical language learning. In *Association for Computational Linguistics (ACL)*.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NIPS)*.

Snyder, J., Knowles, R., Dredze, M., Gormley, M. R., and Wolfe, T. (2013). Topic models and metadata for visualizing text corpora. In *North American Chapter of the Association for Computational Linguistics (NAACL) (Demo Paper)*.

Sofaer, S. and Firminger, K. (2005). Patient perceptions of the quality of health services. *Annu Rev Public Health*, 26:513–559.

Stevens, K., Kegelmeyer, P., Andrzejewski, D., and Buttler, D. (2012). Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

BIBLIOGRAPHY

Taddy, M. A. (2012). On estimation and selection for topic models. In *AISTATS*.

Talley, E., Newman, D., II, B. H., Wallach, H., Burns, G., Leenders, M., and McCallum, A. (2011). A database of National Institutes of Health (NIH) research using machine learned categories and graphically clustered grant awards. *Nature Methods*.

Titov, I. and McDonald, R. (2008). A joint model of text and aspect ratings for sentiment summarization. In *ACL*.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., and Welpe, I. M. (2010). Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10:178–185.

Ueda, N. and Nakano, R. (1998). Deterministic annealing EM algorithm. *Neural Networks*, 11(2):271–282.

Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57.

Walker, A. J. (1977). An efficient method for generating discrete random variables with general distributions. *ACM TOMS*, 3(3):253–256.

Wallace, B., Paul, M., Sarkar, U., Trikalinos, T., and Dredze, M. (2014). A large-scale quantitative analysis of latent factors and sentiment in online doctor reviews. *Journal of the American Medical Informatics Association*, 21(6):1098–1103.

Wallach, H. (2006). Topic modeling: beyond bag-of-words. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 977–984.

BIBLIOGRAPHY

Wallach, H., Mimno, D., and McCallum, A. (2009a). Rethinking LDA: Why priors matter. In *NIPS*.

Wallach, H., Murray, I., Salakhutdinov, R., and Mimno, D. (2009b). Evaluation methods for topic models. In *ICML*.

Wallach, H. M. (2008). *Structured Topic Models for Language*. PhD thesis, University of Cambridge.

Wang, C. and Blei, D. (2009). Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process. In *NIPS*.

Wang, W. Y., Mayfield, E., Naidu, S., and Dittmar, J. (2012). Historical analysis of legal opinions with a sparse mixed-effects latent variable model. In *ACL*, pages 740–749.

Wax, P. (2002). Just a click away: Recreational drug web sites on the Internet. *Pediatrics*, 109(6).

Williamson, S., Wang, C., Heller, K., and Blei, D. (2010). The IBP-compound Dirichlet process and its application to focused topic modeling. In *ICML*.

Yao, L., Mimno, D., and McCallum, A. (2009). 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.

Zeiler, M. (2012). ADADELTA: An adaptive learning rate method. *CoRR*, abs/1212.5701.

BIBLIOGRAPHY

Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214.

Zhai, C., Velivelli, A., and Yu, B. (2004). A cross-collection mixture model for comparative text mining. In *KDD*.

Zhai, K. and Boyd-Graber, J. (2013). Online topic models with infinite vocabulary. In *International Conference on Machine Learning*.

Zhang, D., Zhai, C., Han, J., Srivastava, A., and Oza, N. (2009). Topic modeling for OLAP on multidimensional text databases: topic cube and its applications. *Statistical Analysis and Data Mining*, 2.

Vita

Michael J. Paul received the B.S. degree in Computer Science from the University of Illinois at Urbana-Champaign in 2009. He enrolled in the Computer Science Ph.D. program at Johns Hopkins University in 2010, earning the M.S. Eng. degree in 2012. He was awarded the National Science Foundation Graduate Research Fellowship as well as the Johns Hopkins Whiting School of Engineering Dean's Fellowship in 2010, and he was awarded the Microsoft Research PhD Fellowship in 2013. His research has focused on natural language processing and machine learning with applications to social science and public health.

Beginning in August 2015, Michael will be an Assistant Professor of Information Science and Computer Science at the University of Colorado, Boulder, as a founding member of a new department focused on data and society.