INCORPORATING DOMAIN KNOWLEDGE IN LATENT TOPIC MODELS

by

David Michael Andrzejewski

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2010

For my parents and my Cho.

ACKNOWLEDGMENTS

Obviously none of this would have been possible without the diligent advising of Mark Craven and Xiaojin (Jerry) Zhu. Taking a bioinformatics class from Mark as an undergraduate initially got me excited about the power of statistical machine learning to extract insights from seemingly impenetrable datasets. Jerry's enthusiasm for research and relentless pursuit of excellence were a great inspiration for me. On countless occasions, Mark and Jerry have selflessly donated their time and effort to help me develop better research skills and to improve the quality of my work. I would have been lucky to have even a single advisor as excellent as either Mark or Jerry; I have been extremely fortunate to have them both as co-advisors.

My committee members have also been indispensable. Jude Shavlik has always brought an emphasis on clear communication and solid experimental technique which will hopefully stay with me for the rest of my career. Michael Newton helped me understand the modeling issues in this research from a statistical perspective. Working with prelim committee member Ben Liblit gave me the exciting opportunity to apply machine learning to a very challenging problem in the debugging work presented in Chapter 4. I also learned a lot about how other computer scientists think from meetings with Ben. Optimization guru Ben Recht provided great ideas and insights about issues of scalability in practical applications of machine learning.

The Computation and Informatics in Biology and Medicine (CIBM) program at the University of Wisconsin–Madison provided invaluable support, both financial and otherwise, for the majority of my graduate career. CIBM gave me the opportunity to get involved in the exchange of ideas at the crossroads of computational and biological sciences in a variety of venues.

The machine learning community at Wisconsin was a constant source of advice, expertise, and camaraderie. In particular it was great to interact with everyone who participated in the AI and

other various reading groups. My graduate school experience was also greatly enriched by interactions and collaborations with past and present members of the Zhu and Craven research groups. In alphabetical orders, members of the Craven lab during my studies were Debbie Chasman, Deborah Muganda, Keith Noto, Irene Ong, Yue Pan, Soumya Ray, Burr Settles, Adam Smith, and Andreas Vlachos, while members of the Zhu group included Bryan Gibson, Nate Fillmore, Andrew Goldberg, Lijie Heng, Kwang-Sung Jun, Ming Li, Junming Sui, Jurgen Van Gael, and Zhiting Xu.

Biological collaborators Brandi Gancarz (Ahlquist lab) and Ron Stewart (Thomson lab) helped keep this research focused on real problems. The biological annotations in Chapter 5 were provided by Brandi, and Ron is the biological expert collaborator for the text mining application in Chapter 7.

The LogicLDA work presented in Chapter 7 benefited from helpful discussions with Markov Logic experts Daniel Lowd, Sriraam Natarajan, and Sebastian Riedel.

My summer internship working with Alice Zheng at Microsoft Research was a great experience where I learned a lot about how research gets applied to real-world problems.

Working with David Stork on the computer analysis of Mondrian paintings was very interesting, and definitely taught me to broaden my horizons with respect to what constitutes a machine learning problem.

My parents and family have always been supportive of my plan to stay in school (nearly) forever, and for that I am incredibly grateful.

Finally, Cho has been with me through the entire process: from graduate school applications right up through the final defense. Her patience and love have meant everything to me and made this all possible.

DISCARD THIS PAGE

TABLE OF CONTENTS

Page

Page

DISCARD THIS PAGE

LIST OF TABLES

Table Page

DISCARD THIS PAGE

LIST OF FIGURES

Figure Page

ABSTRACT

Latent topic models can be used to automatically decompose a collection of text documents into their constituent topics. This representation is useful for both exploratory browsing and other tasks such as informational retrieval. However, learned topics may not necessarily be meaningful to the user or well aligned with modeling goals. In this thesis we develop novel methods for enabling topic models to take advantage of side information, domain knowledge, and user guidance and feedback. These methods are used to enhance topic model analyses across a variety of datasets, including non-text domains.

Table 0.1: Symbols used in this thesis (part one).

Concept	Symbol	Meaning
Data	W	The number of words in the vocabulary
	$\mathbf{w}$	Vector of words
	w_i	The vocabulary word of the i^{th}word in the corpus
	D	The number of documents in the corpus
	$\mathbf{d}$	Vector of document assignments
	d_i	The document associated with the i^{th}word in the corpus
	N_d	Number of words in document d
LDA	β	Dirichlet hyperparameter for topic-word multinomials
	α	Dirichlet hyperparameter for document-topic multinomials
	$\phi_j(v)$	Multinomial topic-word probability $P(w = v\|z = j)$
	$\theta_u(j)$	Multinomial document-topic probability $P(z = j\|d = u)$
	$\mathbf{z}$	Vector of topic assignments
	z_i	The latent topic associated with the i^{th}word in the corpus
	T	Number of latent topics
Collapsed Gibbs	$\mathbf{z}_{-i}$	The vector of latent topic assignments *excluding* z_i
	$n_v^{(d)}$	For a given $\mathbf{z}$, number of times topic v appears in document d
	$n_{-i,v}^{(d)}$	Same as the count $n_v^{(d)}$, but *excluding* index i
	$n_v^{(w)}$	For a given $\mathbf{z}$, number of times word w assigned topic v
	$n_{-i,v}^{(w)}$	Same as the count $n_v^{(w)}$, but *excluding* index i

Table 0.2: Symbols used in this thesis (part two).

Concept	Symbol	Meaning
ΔLDA	o	Observed document label (program failure or success)
	T_u	"usage" topics representing normal program behavior
	T_b	"buggy" topics representing buggy program behavior
	$\alpha^{(s)}$	Dirichlet hyperparameter for *successful* documents (e.g., $[1\,1\,1\,0\,0]$)
	$\alpha^{(f)}$	Dirichlet hyperparameter for *failing* documents (e.g., $[1\,1\,1\,1\,1]$)
Topic-in-set	$C^{(i)}$	Set of compatible topics for w_i
	η	Constraint strength ($\eta = 1$ hard, $\eta = 0$ unconstrained)
	q_{iv}	Standard LDA Gibbs sampling probability of $z_i = v$
	$\delta(v \in C^{(i)})$	Compatibility indicator function, 1 if $v \in C^{(i)}$ and 0 otherwise

Table 0.3: Symbols used in this thesis (part three).

Concept	Symbol	Meaning
	Must-Link (A, B)	Words A and B are Must-Linked
	Cannot-Link (A, B)	Words A and B are Cannot-Linked
	η	Constraint strength ($\eta \to \infty$ hard, $\eta = 1$ unconstrained)
	$\gamma^{(k)}$	Dirichlet tree edge weight *into* node k
	$C(k)$	Children of node k
	L_j	Set of leaves for topic j Dirichlet Tree
	$L(s)$	Set of leaves descended from node s
	I_j	Set of internal nodes for topic j Dirichlet Tree
	$\Delta(s)$	Incoming minus outgoing edge weights at node s
	R	Number of Cannot-Link graph connected components
	$M_{r1} \dots M_{rQ^{(r)}}$	Maximal cliques of connected component r's complement
	$\mathbf{q}$	Maximal clique selection vector for a single topic
Dirichlet Forest	$q_j^{(r)}$	Maximal clique for topic j, connected component r
	$I_v(\uparrow i)$	Ancestors of leaf w_i in topic v Dirichlet Tree
	$C_v(s{\downarrow}i)$	Unique node that is immediate child of internal node s and an ancestor of w_i (may be w_i itself)
	$I_{j,r=q'}$	Set of internal nodes below the r-th branch of tree selected by $\mathbf{q}_j$ when clique $M_{rq'}$ is selected
	$t'(q_u)$	Internal nodes in the subtree selected by q_u
	$a(w_i)$	Set of ancestors of the leaf word w_i
	$s(j)$	Immediate descendants of internal node j
	$a(w_i, j)$	The unique node in $a(w_i) \cap s(j)$ (possibly w_i itself)
	$n_j^{(k)}$	Number of words under node k assigned to topic j
	$\gamma_j^{post(k)}$	Posterior edge weight into k for topic j

Table 0.4: Symbols used in this thesis (part four).

Concept	Symbol	Meaning
LogicLDA	$\mathtt{W}(i, v)$	Logical predicate, *true* iff $w_i = v$
	$\mathtt{D}(i, j)$	Logical predicate, *true* iff $d_i = j$
	$\mathtt{Z}(i, t)$	Logical predicate, *true* iff $z_i = t$
	$\mathbf{o}$	Variable representing all other side information (e.g., sentences)
	KB	Weighted first-order knowledge base $\{(\lambda_1, \psi_1), \ldots, (\lambda_L, \psi_L)\}$
	ψ_i	The i^{th}first-order rule
	λ_i	The weight of the i^{th}rule
	$G(\psi_i)$	The set of groundings for rule ψ_i
	$\lvert \cup_k G(\psi_k) \rvert$	Number of non-trivial groundings for a given knowledge base
	$\mathbb{1}_g$	Indicator function for ground formula g, 1 if g *true* and 0 otherwise
LogicLDA inference	p	MaxWalkSAT probability of random (versus greedy) local step
	Δ	MaxWalkSAT global objective function change
	$\mathbf{z}_{KB}$	Set of all z_i involved in *non-trivial* ground formulas
	z_{it}	Relaxed topic assignments ($z_{it} \in [0, 1]$ and $\sum_t z_{it} = 1$)
	f	Alternating Optimization with Mirror Descent term
		(can be either an LDA term or logic-polynomial term)
	∇f	Alternating Optimization with Mirror Descent gradient w.r.t. f
	η	Alternating Optimization with Mirror Descent mirror descent step size

Chapter 1

Introduction

The goal of this thesis is to make topic models more useful by giving the user tools for integrating domain knowledge into the model. Latent topic models [Hofmann, 1999, Blei et al., 2003a] assume that grouped count data (e.g., words in documents) are generated by group-specific mixtures of hidden components. When these models are applied to a corpus of natural language documents (e.g., newswire articles), the statistical regularities captured by the hidden components often correspond to meaningful semantic themes, earning them the name "topics". These models are useful for a wide variety of tasks, both in natural language processing and beyond.

However, it is widely acknowledged that the discovered topics may not always correspond to what the user had in mind. The mechanisms developed in this work allow the user to influence the learned topics, while still retaining the statistical pattern discovery abilities which make topic modeling such a powerful tool. While a suitably sophisticated researcher may be capable of formulating arbitrary models for any occasion, the goal of this work is to develop simple yet flexible methods for expressing domain knowledge in topic models, enabling non-experts to adapt topic modeling to their application needs.

1.1 A motivating example

Say you are given a text corpus consisting of people's wishes for the New Year [Goldberg et al., 2009] and told to describe the common themes present in these wishes. Manual inspection of individual wishes is one approach, but it is not practical for large numbers of wishes.

A more scalable method would be to examine the word frequencies throughout the corpus (after filtering uninformative "stopwords" [Manning and Schütze, 1999]). The top ten most frequent words in this corpus are shown in the first two columns of Table 1.2, and the "word cloud" visualization[1] in Figure 1.1a displays words with size proportional to their frequency in the corpus. Notice that combining all wishes predominantly yields words which are shared across many different types of wishes ("wish", "happy", "love"), yielding little insight about different wish themes.

A slightly more sophisticated approach would be to *cluster* [Duda et al., 2000] the wishes themselves by their word content, hoping that the clusters correspond to common wish themes. However, notice that wishes such as "To get into grad school and find love of my life" do not fit neatly into a *single* cluster. This wish appears to exhibit two very distinct themes, *love* and *career*, which will probably correspond to two distinct clusters.

Latent topic models [Hofmann, 1999] solve this problem by assuming that a corpus of documents contains a collection of themes, called *topics*. Each topic z consists of a multinomial distribution $P(w|z)$ over vocabulary words w. Each document then consists of a *mixture* of these topics, allowing multiple topics to be present in the same document. Both the topics and the document-topic mixtures are estimated simultaneously from data.

Taking each individual wish to be a document, the remaining columns of Table 1.2 show the estimated word probabilities $P(w|z)$ for three of the twelve learned topics. Figures 1.1b,1.1c, and 1.1d show the corresponding word clouds, where words with high probability $P(w|z)$ within a given topic appear larger. We can see that these topics give us a much richer understanding of common themes than the corpus-wide word frequencies alone.

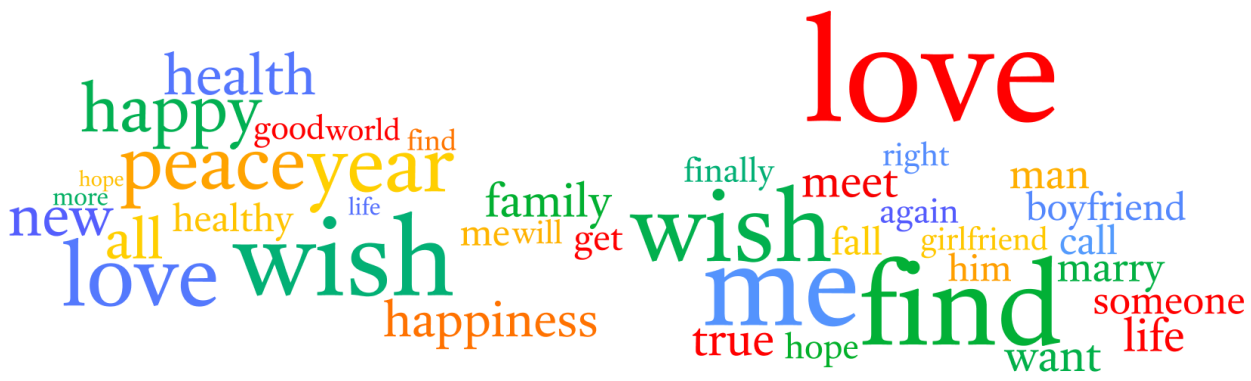
1.2 Latent topic modeling

Latent topic models such as Probabilistic Latent Semantic Analysis (pLSA) [Hofmann, 1999] and Latent Dirichlet Allocation (LDA) [Blei et al., 2003a] model observed data in groups (e.g., words in documents) as being associated with mixtures of unobserved components (the topics). In this thesis we use the LDA model, a more complete generative model [Blei et al., 2003a]. In

[1] http://www.wordle.net

Table 1.1: Example wishes.

Lose weight and get married	To Stop Drinking Liquor and Find a Nice Girlfriend
god bless us all and peace to the world	Vote for Ron Paul!
Peace on Earth	for my cousins cancer to be cured
To get into grad school and find love of my life	God bless us
Get a Job	bring my boyfriend home safe from iraq
Be closer to friends & family and WIN THE LOTTERY!	bush to get impeached
To have all of our dreams realized	More Cowbell



(a) Corpus-wide frequencies.

(b) Learned *love* topic.

(c) Learned *troops* topic.

(d) Learned *religion* topic.

Figure 1.1: Word cloud representations of corpus-wide frequencies and learned topics. More frequent or more probable words appear larger. Note that the "labels" (*love*, *troops*, and *religion*) are manually assigned, not learned automatically.

Table 1.2: Corpus-wide word frequencies and learned topic word probabilities.

Corpus freq		*love* topic		*troops* topic		*religion* topic	
wish	13666	love	0.119	all	0.065	love	0.086
love	11036	me	0.093	god	0.064	forever	0.023
year	10066	find	0.091	home	0.059	jesus	0.022
peace	9647	wish	0.075	come	0.047	know	0.017
happy	8457	true	0.022	may	0.041	loves	0.016
new	7313	life	0.020	safe	0.036	together	0.015
all	7055	meet	0.020	us	0.030	u	0.014
health	7019	want	0.020	bless	0.026	always	0.013
happiness	6173	man	0.017	troops	0.025	2	0.013

LDA, topics are *shared* across all documents, but words in each document are modeled as being drawn from a *document-specific* mixture of these topics. For each word in a document d, a random topic z is sampled from the document-topic mixture $P(z|d)$, and a random word w is then sampled from the corresponding topic-word multinomial $P(w|z)$. Figure 1.2b shows hypothetical topics for a corpus of Presidential State of the Union Addresses, and Figure 1.2a shows the latent topic z associated with each word w in a given sentence.

w	Most	Americans	think	their	taxes	are	high	enough.
z	1	2	4	-	3	-	1	1

(a) Excerpt from the 2008 State of the Union Address delivered by George W. Bush, along with latent topic assignments.

Topic 1 (*amounts*)	Topic 2 (*America*)	Topic 3 (*laws*)	Topic 4 (*opinion*)
many	Americans	exemption	prefer
few	country	forms	consider
most	people	regulation	about
low	citizen	law	think
amount	nation	rebate	agree
high	public	taxes	issue
enough	voters	refund	debate
...	...	...	...

(b) High-probability words for several latent topics.

Figure 1.2: A hypothetical example of topic modeling applied to Presidential State of the Union Addresses.

Besides purely exploratory analysis (as in the initial motivating example), topic models have also been applied to a wide variety of tasks in natural language processing [Boyd-Graber et al., 2007, Newman et al., 2007, Rosen-Zvi et al., 2004], vision [Cao and Fei-Fei, 2007, Wang and Grimson, 2008], and social network analysis [McCallum et al., 2005, Bhattacharya and Getoor, 2006]. In many of these studies, the patterns discovered by latent topic modeling have been exploited to improve performance in other prediction tasks while simultaneously providing human-interpretable "explanations" of the results in terms of the learned topics.

1.3 Challenges in topic modeling

The standard LDA model is *unsupervised*, decomposing the observed data into latent topics according to a purely data-driven objective function (e.g., maximum likelihood). In this sense, latent topic modeling has much in common with unsupervised clustering techniques which attempt to optimize a data-driven objective function. However, this also means that topic models inherit some of the inherent disadvantages of unsupervised learning. For example, the clustering task itself is ill-defined [Caruana et al., 2006], as there may be multiple candidate partitions of the dataset which capture different aspects of the underlying structure. In topic modeling, recovering topics which appear reasonable but are "orthogonal" to user goals is a common failure mode. For example, a user trying to learn *sentiment* topics from a corpus of positive and negative movie reviews [Pang and Lee, 2004] may recover topics related to movie *genre* instead.

Purely unsupervised topic modeling can also recover topics which represent strong statistical patterns but do not correspond to user expectations of semantically meaningful topics. The first row of Table 1.3 shows high-probability words for a topic learned from news articles [Newman et al., 2009]. In this topic the two rather distinct geographical regions of *Korea* and *Carolina* have been merged into a single topic due to their associations with the words "north" and "south". The next two rows contain high-probability words for a pair of topics learned from a corpus of MEDLINE abstracts, and represent scientific measurements and citations, respectively. All of these topics represent real

patterns in the data, but may not align with the user idea of a "topic", potentially making them less useful for exploratory browsing or other tasks.

Table 1.3: Example learned topics which may not be meaningful or useful to the user.

Issue	High probability words
North/south confusion	north south carolina korea korean
Units of measurement	microm km values vmax nmol constant
Citation abbreviations	et al natl proc acad 1992

The simple LDA generative model may fail to capture important structure or extra information. Researchers have therefore developed a variety of extensions to the base LDA model. For example, the topics themselves may be correlated [Blei and Lafferty, 2006a], or we may be modeling both text and image data [Blei and Jordan, 2003]. By exploiting additional assumptions or sources of information, these topic model variants can be more effective in uncovering meaningful topics, and will be discussed further in Chapter 2.

1.4 Adding domain knowledge

The goal of this work is to enhance the effectiveness of latent topic modeling by developing general methods for the incorporation of domain knowledge. In the presence of multiple candidate topic decompositions for a given corpus, domain knowledge can steer the model towards topics which are best aligned with user modeling goals. We also show how a general mechanism for encoding additional modeling assumptions and side information can lessen the need for "custom" topic model variants.

In order to resolve the ambiguities of unsupervised learning, we can turn to recent clustering research for inspiration. Clustering researchers have developed a variety of methods which allow the user to assist the learner in recovering the "correct" clustering by supplying additional domain knowledge. For example, the user could supply a known clustering they do *not* want the learner to return [Gondek and Hofmann, 2004], or pairwise labels for items indicating whether or not they

belong in the same cluster [Wagstaff et al., 2001, Basu et al., 2006, Basu et al., 2008]. These methods combine user guidance with statistical learning in order to improve *quantitative* performance (i.e., accuracy) with respect to the true target clustering. While no clearly analogous notion of accuracy exists for topic modeling, this thesis will show that the inclusion of domain knowledge can help steer topic models towards the discovery of topics which are relevant to user modeling goals.

The more complex topic model variants discussed above can also be viewed as bringing additional domain knowledge to the topic modeling task. However, the creation of "custom" topic models from scratch can be difficult and error-prone. In this thesis we develop general mechanisms for incorporating domain knowledge into topic models, facilitating the use of different types of assumptions and domain knowledge.

1.5 Overview

This thesis begins by formally defining the LDA model and explaining how topics are actually learned from data in Chapter 2. This chapter also discusses some of the general issues related to topic modeling. In Chapter 3 we survey the wide variety of extensions and modifications researchers have made to the base LDA model, as well as some of the recent research on partially supervised clustering techniques.

Chapters 4, 5, 6, and 7 constitute the main body of work in the thesis. Each chapter describes a different mechanism for the incorporation of domain knowledge into latent topic modeling. The types of domain knowledge used by these models form a natural progression from relatively simple forms of guidance to richer and more general types of domain knowledge.

Chapter 4 describes ΔLDA, an extension which allows the user to define "restricted" topics which can only be used in specially labeled documents. This model is applied to statistical debugging, where the learned topics allow us to both locate bugs in code as well as cluster failing program runs by root cause of failure.

Chapter 5 covers Topic-in-Set knowledge, an extremely simple and effective way to include domain knowledge about individual topic assignments, which can be used to construct topics around a few strategically chosen "seed" words.

The Dirichlet Forest prior is introduced in Chapter 4. This prior allows the user to encode prior knowledge about *pairs* of words via Must-Link and Cannot-Link constraints, where Must-Linked words are encouraged to have similar probabilities across all topics while Cannot-Linked words are prevented from both having high probability in the same topic.

Chapter 7 describes LogicLDA, which allows the user to express general domain knowledge in first-order logic (FOL). LogicLDA generalizes several existing topic model extensions, and this work also introduces a new scalable inference technique of possible interest to the Markov Logic Network (MLN) research community. In addition to several example applications on well-known datasets, Chapter 7 contains an application case-study undertaken in collaboration with a biological domain expert. Here we use LogicLDA to learn sets of words related to biological concepts as well as associations between these concepts and genes of interest.

Chapter 8 concludes the thesis, summarizing the contributions and describing directions for further research building on the foundations established in this work.

Without general mechanisms for incorporating domain knowledge, the user is required to to formulate a specialized variant of LDA if they wish to encode application-specific constraints or domain knowledge. Specifying a custom model, deriving the associated inference scheme, and implementing it reliably and efficiently in software requires a non-trivial amount of user expertise and effort, severely limiting the usefulness of topic modeling. By allowing the user to easily incorporate general domain knowledge, the models presented in this thesis (especially LogicLDA) can serve as generic platforms for adaptation of topic modeling to a wide variety of specific applications.

KEY IDEAS

◇ Topic models generate documents with *document-specific* mixtures of *shared* topics.

◇ Topic models are applicable to a variety of tasks, including exploratory analysis.

◇ Domain knowledge can help yield more meaningful and useful topics.

Chapter 2

Background

2.1 Overview

This chapter introduces fundamental topic modeling concepts and notation. We begin with a formal definition of the LDA model and then move on to how one goes about applying it to a given corpus of text documents. We briefly discuss practical topic modeling issues including approximate inference, hyperparameters, and selecting the number of topics.

2.2 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) [Blei et al., 2003a] is a generative probabilistic model [Ng and Jordan, 2001] which can be applied to a corpus of text documents in bag-of-words (word count) form [Manning and Schütze, 1999]. That is, LDA *assumes* a hypothetical generative process is responsible for creating our observed set of documents. This hypothetical procedure involves the generation of the unobserved (i.e., latent) topics themselves. Applying LDA to an observed corpus consists of doing inference to "invert" the generative procedure and recover the latent topics from the observed words.

2.2.1 Model definition

The observed variables in LDA are the words $\mathbf{w}$ and documents $\mathbf{d}$. Let $\mathbf{w} = w_1 \dots w_N$ represent a corpus of length N (i.e., the concatenation of all documents contains N words total) where each word w_i is a discrete random variable belonging to a vocabulary of size W: $\{1, 2, \dots, W\}$. The vector $\mathbf{d} = d_1 \dots d_N$ associates each word with a document index $d_i \in \{1, 2, \dots, D\}$.

LDA assumes that each word $\mathbf{w} = w_1 \dots w_N$ is associated with a latent topic $\mathbf{z} = z_1 \dots z_N$. Each of these topics $t = 1 \dots T$ is associated with a multinomial ϕ_t over the W-word vocabulary, and each ϕ is drawn from a Dirichlet prior with parameter β. Likewise, each document $j = 1 \dots D$ is associated with a multinomial θ_j over topics, drawn from a Dirichlet prior with parameter α. The first section of Table 0.1 restates these definitions for quick reference. The full generative procedure is then

1. For each topic $t = 1, 2, \dots, T$

   Sample topic-word multinomial $\phi_t \sim Dirichlet(\beta)$

2. For each document $j = 1, 2, \dots, D$

   Sample document-topic multinomial $\theta_j \sim Dirichlet(\alpha)$

   For each word $\{w_i | d_i = j\}$

   Sample topic $z_i \sim Multinomial(\theta_j)$

   Sample word $w_i \sim Multinomial(\phi_{z_i})$

where the individual document lengths $N_j = \sum_i \{i | d_i = j\}$ and the total corpus length $N = \sum_{j=1}^{D} N_j$ are assumed to be given.

This procedure implies a joint probability distribution over the random variables $(\mathbf{w}, \mathbf{z}, \phi, \theta)$, which is given by

$$P(\mathbf{w}, \mathbf{z}, \phi, \theta \mid \alpha, \beta, \mathbf{d}) \propto \left(\prod_t^T p(\phi_t | \beta) \right) \left(\prod_j^D p(\theta_j | \alpha) \right) \left(\prod_i^N \phi_{z_i}(w_i) \theta_{d_i}(z_i) \right), \tag{2.1}$$

where $\phi_{z_i}(w_i)$ is the w_i-th element in vector ϕ_{z_i}, and $\theta_{d_i}(z_i)$ is the z_i-th element in vector θ_{d_i}. The conditional dependencies implied by this distribution can be represented by the directed graphical model [Bishop, 2006] shown in Figure 2.1.

A step-by-step example of the generative process is shown in Figure 2.2. Here we have a very simple vocabulary (Figure 2.2a) and $T = 3$ topics (Figure 2.2b). Per the generative procedure

described above, we first sample a topic-word multinomial ϕ for each topic. Figure 2.2c shows the numerical values and Figure 2.2e shows these ϕ vectors plotted graphically on a *simplex*, where the nearness of each ϕ_i to each corner is proportional to the probability that ϕ_i places on the corresponding word. For example ϕ_1 places high probability on w_3 and is therefore very near the top corner which corresponds to w_3. Next, for each document we sample a document-topic multinomial θ as shown in Figure 2.2d and Figure 2.2f. Finally, Table 2.1 shows a document d_1 generated by repeatedly sampling a topic z_i from the document-topic multinomial θ_1 and then sampling the corresponding word w_i from the appropriate topic-word multinomial ϕ_{z_i}.

Table 2.1: Example generated document d_1, with $N_1 = 6$

w_i	Dog	Run	Cat	Run	Dog	Run
z_i	3	3	1	3	2	3

It is important to emphasize that only the words $\mathbf{w}$ and their documents $\mathbf{d}$ are actually observed. The hyperparameters α and β can be either user-supplied or estimated from data [Blei et al., 2003a, Wallach, 2008]. The latent topic assignments $\mathbf{z}$, topic-word multinomials ϕ, and document-topic multinomials θ are all *unobserved*. We now turn to a critical step in topic modeling: the recovery of these latent variables via posterior inference given a corpus $(\mathbf{w}, \mathbf{d})$.

2.2.2 Inference

Estimation of (ϕ, θ) requires knowledge of the latent topic assignments $\mathbf{z}$. Unfortunately the posterior $P(\mathbf{z}|\mathbf{w}, \mathbf{d}, \alpha, \beta)$ over $\mathbf{z}$ given the observed corpus $(\mathbf{w}, \mathbf{d})$ and model hyperparameters (α, β) is intractable due to the coupling between ϕ and θ in the exponentially large summation over all possible $\mathbf{z}$ [Blei et al., 2003a, Sontag and Roy, 2009]. Researchers have therefore developed various schemes for doing approximate inference, such as Variational Bayes (VB) [Blei et al., 2003a], Expectation-propagation (EP) [Minka and Lafferty, 2002], Collapsed Gibbs Sampling (CGS) [Griffiths and Steyvers, 2004], and Collapsed Variational Bayes (CVB) [Teh et al., 2006b]. A brief note about differences between these approaches is postponed until the

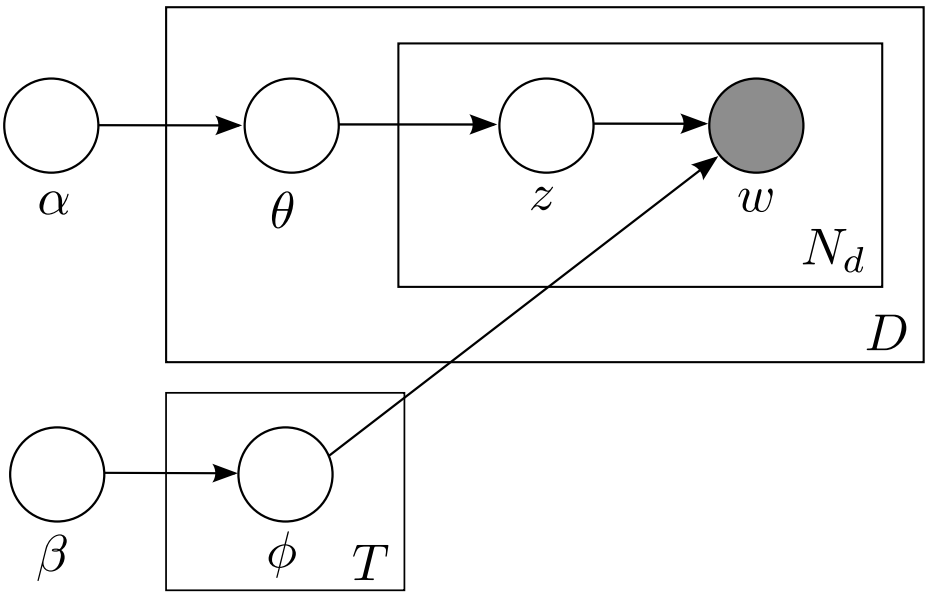


Figure 2.1: The directed graphical model representation of Latent Dirichlet Allocation (LDA). Each node represents a random variable or model hyperparameter, and the directed edges indicate conditional dependencies. For example, each word w depends on both the latent topic z and the topic-word multinomial ϕ. The "plates" indicate repeating structures: the T different ϕ drawn from $Dirichlet(\beta)$, the D documents, and the N_d words in each document d.

	1	Dog
w	2	Run
	3	Cat

(a) Example vocabulary.

T	3
α	0.5
β	0.1

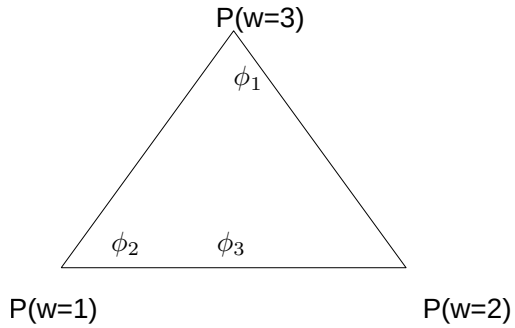
(b) Example parameters.

ϕ		w		
		1	2	3
	1	.1	.1	.8
z	2	.8	.1	.1
	3	.5	.4	.1

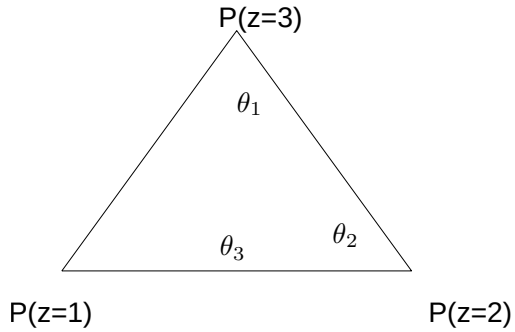
(c) $\phi_z \sim Dirichlet(\beta)$

θ		z		
		1	2	3
	1	.15	.15	.7
d	2	.15	.7	.15
	3	.5	.4	.1

(d) $\theta_d \sim Dirichlet(\alpha)$



(e) Simplex representation of ϕ.



(f) Simplex representation of θ.

Figure 2.2: Example of the LDA generative process.

discussion of hyperparameters in Section 2.2.2.3. We now review CGS in more detail, as much of the work in this thesis builds on this approach.

2.2.2.1 Collapsed Gibbs Sampling (CGS)

Assume the hyperparameters (α, β) are given and fixed. CGS then consists of integrating out (ϕ, θ) and sampling the topic assignments $\mathbf{z}$ from the posterior $P(\mathbf{z}|\mathbf{w}, \mathbf{d}, \alpha, \beta)$. For each iteration of the sampling procedure, we resample z_i for every position i in the corpus according to the following formula:

$$P(z_i = v|\mathbf{w}, \mathbf{d}, \mathbf{z}_{-i}, \alpha, \beta) \propto \left(\frac{n_{-i,v}^{(d)} + \alpha}{\sum_u^T (n_{-i,u}^{(d)} + \alpha)}\right)\left(\frac{n_{-i,v}^{(w_i)} + \beta}{\sum_{w'}^W (n_{-i,v}^{(w')} + \beta)}\right) \tag{2.2}$$

where $n_{-i,v}^{(d)}$ is the number of times $\mathbf{z}$ uses topic v in document d excluding position i. Likewise, $n_{-i,v}^{(w_i)}$ is the number of times $\mathbf{z}$ uses topic v to emit word w_i, again excluding position i. This scheme is an instance of Markov Chain Monte Carlo (MCMC) inference; it can be shown that samples drawn in this way constitute states of a Markov chain whose stationary distribution is the true posterior of $\mathbf{z}$ [MacKay, 2003]. A significant advantage of CGS is ease of implementation in software: the only data structures required are the count matrices $n_t^{(w)}$ (dimensionality $T \times W$) and $n_t^{(d)}$ (dimensionality $T \times D$). Furthermore, the derivation of new sampling equations for LDA variants (see Chapter 3) is usually relatively straightforward.

After running our Markov chain for sufficiently many samples [Gelman et al., 2004], one can use the counts from the *final* sample of the chain to estimate the topics ϕ and document mixing weights θ as the means of their posteriors

$$\hat{\phi}_t(w) = \frac{n_t^{(w)} + \beta}{n_t^{(*)} + \sum_{w'}^W \beta} \qquad\qquad \hat{\theta}_j(t) = \frac{n_t^{(j)} + \alpha_t}{n_*^{(j)} + \sum_{t'}^T \alpha_{t'}} \tag{2.3}$$

where $\phi_t(w)$ is the probability of word w under topic t, $P(w|z = t)$, and $\theta_j(t)$ is the probability of seeing topic t in document j, $P(z = t|d = j)$. The count $n_t^{(*)}$ is the total number of all words assigned to topic t, and likewise the count $n_*^{(j)}$ is the total number of words in document j.

It may seem unusual to estimate $(\hat{\phi}, \hat{\theta})$ from a single sample. However, it is not valid to pool or average across multiple samples because the probability distribution is invariant to permutations of the topic indices. Informally, Topic 2 may not "mean" the same thing from one sample to another. Quantities that are insensitive to the topic indices (e.g., how often are w_i and $w_{i'}$ are assigned to the same topic) can safely be estimated across multiple samples, however.

2.2.2.2 Initialization

Theoretically, the initial state $\mathbf{z}$ of the Gibbs sample should not matter, since the Markov chain will (eventually) converge to the true distribution after many samples. However, in practice it may speed convergence to initialize using a reasonable heuristic. In this thesis, we do "online-like Gibbs" initialization. In this scheme we begin with an empty $\mathbf{z}$ and incrementally sample each z_i according to Equation 2.2, using counts from the previously assigned z_i. That is, the very first z_i will be assigned based on the hyperparameters only, while the final z_i will be a true collapsed Gibbs sample conditioned on all other positions $\mathbf{z}_{-i}$.

2.2.2.3 Hyperparameters and scalability

Recall the variety of inference techniques mentioned earlier in this chapter. Given all of these inference methods, how could a prospective topic modeler decide which to use? Researchers have typically studied the performance of these approaches along two dimensions: computational speed and topic "quality".

The quality of the resulting topics is usually defined in terms of data fit, as calculated by the likelihood they assign to a set of held-aside documents [Wallach et al., 2009b]. However it must be noted that, while it is crucial that the learned topic model capture true regularities present in the data, ultimate notions of topic quality are inescapably dependent on modeling goals (i.e., how will these topics be actually be used?). Surprisingly, a user study conducted with Amazon Mechanical Turk [Chang et al., 2009] found topic data fit to be *inversely* proportional to topic interpretability in some cases.

Data fit still provides a useful means of evaluating different approximate inference schemes in isolation from specific topic modeling applications. Interestingly, recent empirical work [Asuncion et al., 2009] has found that performance differences between inference approaches may be an artifact of subtle variation in the effects of the smoothing hyperparameters (α, β) for each approach. The implication of these findings is that, as long as the hyperparameters (α, β) are tuned to the particular inference technique and dataset, the choice of inference method need not have a strong influence on learned topic quality. The model hyperparameters can be learned using a variety of techniques [Minka, 2000, Blei et al., 2003a, Wallach, 2008], for example within a Gibbs EM procedure when coupled with CGS. Other research also confirms that learning the hyperparameters from data (particularly α) can have a strong impact on the quality of the learned topics [Wallach et al., 2009a].

In this thesis we use *fixed* values for the hyperparameters (α, β) for simplicity, with specific values indicated for each experiment. For fixed values, a reasonable starting point [Griffiths and Steyvers, 2004] is to set $\beta = 0.1$ and $\alpha = 50/T$ where T is the number of topics.

We briefly return to the other critical dimension of computation time — how long does it take to learn topics from 1,000 documents? 10,000? 10 million? Speed differences among the inference schemes exist [Asuncion et al., 2009], with CVB-style schemes generally being the fastest due to being both deterministic and collapsed. However, performance differences between schemes may be swamped by the scalability gains to be had from modifications which parallelize inference [Asuncion et al., 2008, Newman et al., 2008] across multiple cores or machines. For applications where scalable inference is critical, distributed inference should definitely be considered.

2.2.3 The number of topics

A perennial question in topic modeling is how to set the number of topics T. Several approaches exist, but ultimately, the appropriate number of topics must depend on both the corpus itself and user modeling goals.

- **Set manually via "trial and error"**

  If there is a human in the loop, it may be simplest to try multiple values of T within some reasonable range (e.g., $T \in \{5, 10, 25, 50, 100, 200\}$). The user can then quickly scan the learned topics associated with each value of T and select the value which seems most appropriate to the corpus.

- **Use domain knowledge**

  If the topics are meant to correspond to known entities in the world (e.g., in a vision task each topic may be a type of object), then we can simply set T equal to the true number of entities we are trying to model. For example, in Chapter 4 we assume that topics correspond to bugs in software; knowing the number of bugs present, we set T accordingly.

- **Optimize with respect to held-aside likelihood**

  Given a means of evaluating the probability $P(\mathbf{w}'|T)$ of a validation set of held-aside documents [Wallach et al., 2009b], we can learn topics for different values of T and choose the value which maximizes the likelihood of the held-aside validation set [Griffiths and Steyvers, 2004]. Plotting these values, we can typically see the familiar pattern of $P(\mathbf{w}'|T)$ increasing with larger T up to a point, beyond which the model *overfits* [Mitchell, 1997] the data and $P(\mathbf{w}'|T)$ on the held-aside documents begins to fall.

- **Optimize performance on secondary task**

  If the learned topics are to be used as input to another task, then it follows that T should be chosen according to performance on the ultimate task of interest. For example, if the topics are going to be used for document classification, T could be chosen to optimize classifier performance on a held-aside validation set.

- **Infer the number of topics automatically**

  Researchers have recently applied ideas from *nonparametric Bayesian statistics* to sidestep the issue of setting T altogether [Teh et al., 2006a]. The Dirichlet Process (DP) [Neal, 1998] is a distribution over multinomial distributions with potentially infinitely many components.

Loosely speaking, we can therefore encode uncertainty about the number of topics T by replacing a Dirichlet prior with a Dirichlet Process prior. Inference under these models then "automatically" sets the number of topics T based on the observed data and given hyperparameters. However, note that hyperparameter choice *indirectly* influences the number of topics T the model will use, and these hyperparameters must be chosen somehow (either manually selected or learned from data).

For the work presented in this thesis, we set T manually, either by from domain knowledge or by trial and error.

KEY IDEAS

- ◇ LDA assumes a specific *generative* procedure was responsible for the observed corpus.
- ◇ Applying LDA consists of *inferring* the hidden topics given the observed corpus.
- ◇ This inference problem is *intractable*, but various approximation algorithms exist.

Chapter 3

Related work

This chapter covers existing research related to the use of domain knowledge in topic models. The fact that LDA is a well defined probabilistic model has enabled researchers to craft a wide variety of customizations and extensions to the base model. We discuss a variety of extensions to the standard unsupervised LDA topic model which exploit additional information or structure to learn richer, more informative models. We also introduce a rough categorization of these topic model variants to help organize this body of work. The work presented in this thesis complements and extends the existing research on topic modeling by providing general mechanisms for the inclusion of user provided domain knowledge.

We also review models and algorithms which augment clustering with additional constraints or side information. Because of the relationship between clustering and topic modeling discussed in Chapter 1, these methods can provide useful ideas and perspectives which we can bring to bear on topic modeling problems. In particular, the Dirichlet Forest prior (Chapter 6) adapts and applies ideas from constrained clustering to the topic modeling context.

3.1 LDA variants

This chapter discusses extensions to the base LDA model. We loosely categorize these approaches as modeling additional types of information (LDA+X), or modifying the word generation (ϕ-side) or topic generation (θ-side) aspects of the base LDA model. Note that these categories are intended purely as a rough guide to organizing our understanding and not a "hard" partitioning. For

several of these models, one could make valid arguments that they belong in a different category than the one in which they are presented here.

3.1.1 LDA+X

These models extend LDA by modeling additional observed data beyond the text documents. This additional data could be document labels, images associated with the documents, or links between documents. We loosely categorize these models as "LDA+X" variants. Intuitively, the latent topics are forced to "explain" this additional data as well as the document text. Correlations between document text and the additional data can then shape the recovered topics. These models also support additional applications, such as labeling new documents, inferring image annotation terms, or predicting unseen edges between documents.

Supervised models: Supervised LDA [Blei and McAuliffe, 2008] can be applied to labeled documents, augmenting each document d with a label variable y_d, which can be either categorical (for classification) or continuous (for regression). Each y_d value is modeled by a Generalized Linear Model (GLM) in the vector of mean topic counts $\bar{z} = \frac{1}{N_d}\sum_{n=1}^{N_d} z_n$ for that document. This approach can therefore make label predictions by calculating the posterior topic assignments for a test document to obtain a $\bar{z}$ value. Furthermore, jointly training the model in this way tends to produce topics which are able to "explain" the label value $\mathbf{y}$ for the training set. In this way the label information indirectly influences the particular nature of the topical decomposition discovered by the model. This adapts LDA to the standard supervised learning setting, but restricts user guidance to providing labels or response values $\mathbf{y}$ to be explained. MedLDA [Zhu et al., 2009] also does supervised topic modeling, attempting to learn a *max-margin* classifier using the latent topics.

Text+image: Correspondence LDA [Blei and Jordan, 2003] is a joint model of images and associated text (e.g., captions). Each latent topic is associated with a multivariate Gaussian distribution for generating image patches *and* a multinomial distribution for generating caption words. This coupling captures the idea that the captions of similar images should contain similar text topics, and allows interesting applications such as the automatic annotation of new images. Dual-Wing Harmoniums [Xing et al., 2005] are a similar model based on the undirected harmonium graphical

model. Multi-class LDA with annotations [Wang et al., 2009] combines ideas from Supervised LDA and Correspondence LDA to jointly model an image, the supervised label of an image, and terms used to annotate the image.

Note that applications of topic-style models directly to vision tasks constitute a significant research literature of their own [Fei-Fei and Perona, 2005, Cao and Fei-Fei, 2007, Wang and Grimson, 2008, Wang et al., 2007], which we will not discuss further in this thesis.

Authors and networks: The Author-Topic Model [Rosen-Zvi et al., 2004] adds observed author information for each document. Within a document, topics are generated by first sampling an *author*, and then sampling a topic from an *author-topic* mixture. The associations between individuals and topics can then be used for applications such as assigning reviewers to scientific papers [Mimno and McCallum, 2007] or assigning developers to software bugs [Linstead et al., 2007].

Even more interestingly, for communication texts such as e-mail we may have both author and *recipient* information [McCallum et al., 2005], which can be used to further influence the topics. This is an example of including *network* structure in the topic model. Many types of documents are intrinsically networked, such as scientific articles and citations, or websites and links. Topic-link LDA [Liu et al., 2009], Relational Topic Models [Chang and Blei, 2009], and the Citation Influence Model [Dietz et al., 2007] represent this additional structure.

3.1.2 ϕ-side

Recall that words in LDA are generated by simple topic multinomials ϕ, which are in turn drawn from a Dirichlet prior with hyperparameter β. The models presented in this section modify the word generating aspect of LDA in order to capture richer linguistic structure or side information. We therefore loosely group these models together as ϕ-side.

Syntax and sequence-aware models: These topic models dispense with the "bag of words" assumption and explicitly attempt to capture the sequential structure of language. Hidden Topic Markov Models [Gruber et al., 2007] enforce that all words in a sentence share the same topic, and at each sentence boundary, flip a coin to decide whether to draw a new topic from the document mixture θ or simply use the same topic as the previous sentence. This nicely captures the

notion of the topic coherence within sentences. The Bigram Topic Model [Wallach, 2006] and Topical $n-$gram Model [Wang and Mccallum, 2005] extend the word generation model to allow conditioning on previous words, enabling the representation of meaningful bigrams such as "neural networks". HMM-LDA [Griffiths et al., 2004] embeds a topic model within a Hidden Markov Model, allowing most HMM states to model *syntactic* words while a special *semantic* HMM state emits words using the LDA model, capturing the document-level themes. Syntactic Topic Models [Boyd-Graber and Blei, 2008] explicitly model the dependency parse trees of sentences, forcing topics to explain both observed words and hidden topics of child nodes in the tree. This allows the model to learn syntactically relevant topics capturing grammatical regularities in language.

Side information: The Concept-Topic Model [Chemudugunta et al., 2008] extends LDA with special topics, known as "concepts." A concept is defined by some subset of the vocabulary c, and is restricted to place non-zero probability only on words $w \in c$. That is, if z is a concept then $P(w|z) = 0$ for all $w \notin c_z$. These subsets of the vocabulary can be supplied by some external knowledge source, such as a concept hierarchy. Words assigned to concepts can then be understood to correspond to a *known and labeled* concept, allowing interesting applications such as tagging individual words, labeling documents, or summarizing corpora. The Concept-Topic Model has the advantage of offering a straightforward way to link known concepts to the data, but cannot generalize the provided concepts and is somewhat limited in expressiveness.

LDA with WordNet (LDAWN) [Boyd-Graber et al., 2007] uses an existing linguistic resource in order explicitly model word sense (e.g., "tree bank" versus "bank bailout"). WordNet [Miller, 1995] organizes words into synonym sets (synsets), and these synsets themselves can be organized into a hyponym ("is-a") graph. LDAWN modifies the LDA generative procedure by having topics emit words via a random walk over this graph. Since different senses of a given word reside in different synsets, the latent assignment of a token to a synset should reveal the associated word sense. This model has achieved strong results on word sense disambiguation tasks.

3.1.3 θ-side

In standard LDA, the document-topic mixture θ is independently sampled from a Dirichlet prior α for each document. This simple approach may ignore correlations which may exist among the topics, or information we may have about the documents. θ-side variants are primarily concerned with richer representations of the document-topic associations.

Topic structure: The standard Dirichlet prior on θ assumes no dependencies between the individual topics in a given document (aside from the normalization requirement that $\sum_t \theta_d(t) = 1$). However, the presence of a *neural networks* topic (e.g., containing "backpropagation", "sigmoid") in a scientific article may lead us to expect to see an *experimental methods* topic (e.g., "cross-fold", "significant") as well.

Hierarchical LDA (hLDA) [Blei et al., 2003b], Correlated Topic Models (CTM) [Blei and Lafferty, 2006a], and Pachinko Allocation Machines (PAM) [Li and McCallum, 2006, Mimno et al., 2007] all modify the topic sampling procedure in different ways in order to capture dependencies between topics. CTM replaces the Dirichlet prior on θ with a logistic normal distribution, allowing pairwise correlations between topics. PAM generates topics by root-to-leaf traversal of a directed acyclic graph (DAG), encoding topic correlations via the outgoing edge weights of the internal nodes. hLDA models topics with a tree-structured hierarchy over topics where topics get more specific as one moves from root to leaf. In addition to potentially modeling the data more accurately, the connections between topics may yield additional useful insights.

Document information: It is not uncommon for us to have additional information associated with the documents themselves. If we believe there to be a relationship between this information and the content of the document, it makes sense to try to incorporate it into the topic model. The Dirichlet-multinomial regression (DMR) topic model [Mimno and McCallum, 2008] is a flexible model which conditions θ on arbitrary document metadata (e.g., authors, citations, or timestamps). This is accomplished by modeling the Dirichlet hyperparameter α as being generated by a weighted function of the document metadata vector.

Labeled LDA [Ramage et al., 2009] assumes that each document is associated with a $K-$dimensional binary vector Λ of labels. For example, a document may be tagged with the

tags *sports* and *business*. Each of these labels is then associated with its own special topic which can only be used in documents which have that label. This allows the model to learn special topics that are strongly associated with their corresponding labels. Returning to our example, Labeled LDA would have a special topic which only appears in documents having the *sports* tag, which is therefore likely to place high probability on words associated with sports.

Markov Random Topic Fields (MRTF) [Daumé, 2009] assume the existence of a weighted graph over documents (e.g., shared authors or publication venues). Intuitively, we believe that documents connected by a high-weight edge should contain similar topics. For each edge (d_1, d_2), the model adds a potential function penalizing the difference between θ_{d_1} and θ_{d_2}.

Gaussian Markov Random Field (GMRF) topic models [Mimno et al., 2008] incorporate a similarity graph over document *collections*. Each document draws a topic proportion θ from a logistic normal distribution, as in CTM. However, this model introduces additional coupling between the CTM parameters *within* a collection. Furthermore, the GMRF introduces parameter coupling *across* document collections depending on the user-supplied graph structure.

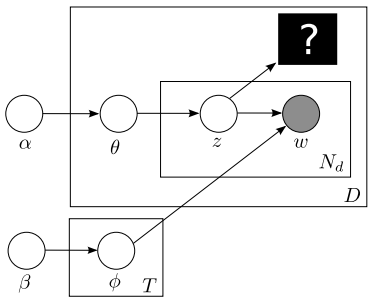
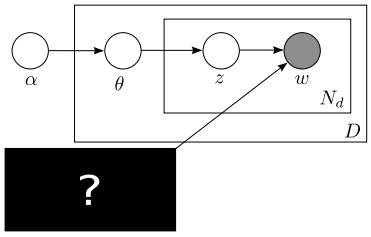
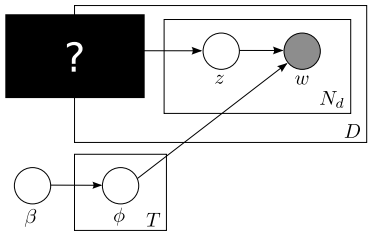
Topics over time: Often, documents come with some sort of *timestamp* (e.g., year of publication for scientific articles). When doing topic modeling, it is therefore natural to ask if we can exploit this temporal information in order to learn something about the the evolution of topics and trends in their usage.

In dynamic topic models (DTM) [Blei and Lafferty, 2006b], it is assumed that we can partition the corpus into disjoint time slices. Using logistic normal distributions (again as in CTM), both the document-topic mixtures and topic-word multinomials evolve via multivariate Gaussian dynamics (i.e., at time step s natural parameter ν_s is Gaussian distributed with mean ν_{s-1}). The requirement of discretized timestamps is an obvious limitation of this approach. The continuous time dynamic topic model (cDTM) [Wang et al., 2008] is similar to CTM, but replaces discrete Gaussian evolution with its continuous limit, Brownian motion. Topics over time (TOT) [Wang and McCallum, 2006] takes a different approach, modeling timestamps as being *generated* by the model itself. These models are all able to learn interesting trends in topics. For example, TOT shows interesting results when applied to a corpus of NIPS publications; the *neural network*

topic becomes less prevalent relative to the *support vector machines* topic over the time period from 1987 to 2003 [Wang and McCallum, 2006].

3.1.4 Summary and relation to this thesis

Table 3.1: Overview of LDA variant families discussed in this chapter.

Variant family	Diagram	Examples
LDA+X		Images, labels
ϕ-side		Concepts, WordNet
θ-side		Topic correlations

This chapter has demonstrated the diversity of potential extensions to the basic topic model. Table 3.1 summarizes the different families of model variants. In general, however, these models tend to assume particular types of additional structure or side information, as well as how these additions influence topic recovery. The goal of this thesis is to create *general* mechanisms allowing

the user to inject different types of domain knowledge into topic modeling. The models we develop provide new and general mechanisms for incorporating domain knowledge. In particular, the LogicLDA model (Chapter 7) is quite flexible, and can be used to "encode" many of the existing variants presented in this chapter.

3.2 Augmented clustering

Recall that standard LDA is a fundamentally *unsupervised* model, and is often used in an exploratory setting to learn more about a given dataset. In both of these senses, topic modeling is related to the classic problem of *clustering*, where we wish to find a "good" partition of a given set of instances [Duda et al., 2000]. As discussed earlier, topic modeling and clustering also share certain challenges such as the difficulty of evaluating solutions and the existence of multiple meaningful solutions. In recent years there has been exciting research on augmenting traditional clustering approaches with various types of partial supervision or user guidance. We now briefly review this line of work, as it has helped to shape some of the directions investigated in this thesis.

3.2.1 Constrained clustering

Often when clustering, we have some idea what our desired clustering should look like. One particular type of knowledge is a *pairwise label*: given a pair of instances, should they end up in the same cluster, or not? In constrained clustering [Wagstaff et al., 2001, Basu et al., 2006, Basu et al., 2008], this information is supplied in the form of Must-Link and Cannot-Link constraints, respectively. Candidate clustering solutions which violate these constraints are then penalized. An alternative approach [Xing et al., 2002] uses these types of pairwise labels to learn a *distance metric*. The learned metric should place Must-Linked instances near one another, and Cannot-Linked instances far apart. Clustering is then done according to the learned distance function.

Inverting the previous setting, we may know what our clustering should *not* look like. That is, we may already know a high-quality clustering of the data, and we do not want the algorithm to tell us what we already know. Conditional Information Bottleneck (CIB) [Gondek and Hofmann, 2004]

optimizes an information-theoretic objective function [Tishby et al., 1999] *conditioned* on a previously known clustering. Informally, this approach tries to discover a clustering that "tells us something new", relative to the clustering we already know about. A related approach, Information Bottleneck with Side Information [Chechik and Tishby, 2002], allows the user to supply additional side information, labeled as either relevant or irrelevant. Again, an information-theoretic approach attempts to maximize mutual information with respect to the relevant side information while minimizing mutual information with respect to the irrelevant side information.

3.2.2 Interactive clustering

Other recent work is explicitly targets *interactive* clustering. One approach to the "multiple valid clusterings" problem [Dasgupta and Ng, 2009, Dasgupta and Ng, 2010] is built upon *spectral clustering* [Shi and Malik, 2000] as the base clustering method. Spectral clustering operates on a similarity graph between items, and the resulting solutions are given in terms of *eigenvectors* of the associated *graph Laplacian*. A binary clustering based on the second eigenvector[1] is the optimizer of a purely data-driven objective function. The eigenvectors themselves are orthogonal, and therefore represent significantly different clusterings. By taking separate clusterings for each of the top N eigenvectors and presenting them to the user, an appropriate clustering can be discovered. An alternative approach [Bekkerman et al., 2007] uses "seed" words which are thought to be highly informative with respect to the target clustering (but are not themselves "labeled"). The algorithm then proceeds via information-theoretic biclustering, an approach which works by simultaneously clustering the words (based on their appearance in the document clusters) and the documents (based on which word clusters they contain). If the mutual information between the seed words and the target document clustering is high, initializing this procedure from the seed words should be more likely to result in the recovery of the target document clustering.

[1] The first eigenvector is constant over all items and therefore not useful for clustering.

KEY IDEAS

◇ LDA is an extensible base model, yielding many application- and data-specific variants.

▷ **LDA+X** variants jointly model text documents and *related data* (e.g., labels or images).

▷ ϕ**-side** variants modify the generation of *words* (e.g., word order).

▷ θ**-side** variants modify the generation of *topics* (e.g., topic correlations).

◇ The goal of this work is to develop general methods for including domain knowledge.

◇ Recent clustering work incorporates constraints and user interaction - these ideas are also applicable to topic modeling.

Chapter 4

DeltaLDA

The work described in this chapter exploits prior knowledge at the document level. The ΔLDA model [Andrzejewski et al., 2007] assumes that a corpus is generated using two sets of topics: a *shared* set of topics used to model all documents, and an *exclusive* set of topics which can only appear in a special subset of documents. Intuitively, the exclusive topics are forced to "explain" the patterns present in the special subset of documents but absent from the corpus as a whole (this difference is the "delta" in the model name). In the topic modeling taxonomy from Chapter 3, ΔLDA is a θ-side modification, including domain knowledge about documents into the generation of the document-topic multinomial θ.

While the ΔLDA model is quite general, it was originally motivated by a statistical debugging application. We therefore begin with a brief introduction to the statistical debugging dataset and problem setting.

4.1 Cooperative Bug Isolation (CBI)

The Cooperative Bug Isolation Project (CBI) [Liblit, 2007] aims to improve understanding of software failures (bugs) by recording data about program executions (runs) via special instrumentation code woven into the original program code. A simplified example of this instrumentation is a branch counter (Figure 4.1) which records which direction was actually taken at a given conditional expression in the original program.

For practical reasons, these observations are only recorded probabilistically (subsampled) and no ordering information is preserved. The instrumentation software deterministically adds a finite

```
int x = my_func()
if (x > 5) {
    branch_42_true++
    ...
}
else {
    branch_42_false++
    ...
}
```

Figure 4.1: An example predicate used by CBI.

number of these recorders to the original program code, and for each execution (run) these predicates are each associated with an event count telling us how many times that event was observed during execution.

We use latent topic modeling to better understand this dataset. Although program behavior may seem quite different from natural language text, both domains feature discrete count data (word counts or predicate counts) with a natural grouping structure (document or program run). Much as LDA can yield insights by showing us the latent topic patterns behind a text corpus, we hope to discover the latent program behavior patterns behind a corpus of program runs. Table 4.1 shows a mapping from our previous text modeling domain to this statistical debugging domain.

Table 4.1: Schema mapping between text and statistical debugging.

Text Data	CBI Data
Document	Program run
Vocabulary	Predicates
Word counts	Predicate counts

4.2 DeltaLDA

Suppose we have a buggy implementation of the UNIX utility `grep`, and we are interested in using the CBI framework to understand the buggy behavior. We record the predicate counts for many runs of our instrumented copy of `grep`, some of which crash or give incorrect output (fail) and some of which run normally (succeed). Given this corpus of program runs, we can apply latent topic modeling in order to discover underlying patterns of predicate activation across these runs.

One potential problem with this approach is that much of the statistical structure present in the predicate counts may be associated with *normal* (i.e., non-buggy) program operation. This problem is illustrated by the "buggy bars" synthetic dataset shown in Figure 4.2. Here the "vocabulary" is a 5×5 grid of 25 pixels, and the "word counts" for each document are represented by normalized pixel intensity, with white being the highest and black being zero. Our synthetic documents will be partitioned into two sets: *successful* and *failing*. In Figure 4.2a we see the ground truth topics used to generate the synthetic data. Each of these topics assigns uniform probability to a subset of the pixel vocabulary (a horizontal or vertical bar), and zero probability to the rest of the vocabulary. The eight horizontal and vertical bar topics represent the normal usage patterns, and are associated with hyperparameter $\alpha = 1$ in *all* documents, while the three "x"-shaped topics represent weaker bug patterns and are associated with hyperparameter $\alpha = 0.1$ in *failing* documents only. A handful of successful (left) and failing (right) documents generated by this synthetic data model are shown in Figure 4.2b. To be more concrete, the α hyperparameters used to generate the document-topic mixture weights θ are:

$$\alpha = \begin{bmatrix} \alpha^{(s)} \\ \alpha^{(f)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.1 & 0.1 & 0.1 \end{bmatrix}. \tag{4.1}$$

For a successful (non-failing) document, we generate θ by sampling from a Dirichlet using the first row of values $\alpha^{(s)}$. The 0 entries for the last 3 columns ensure that the corresponding θ will place zero probability on those topics. Likewise for a failing document we use the second row $\alpha^{(f)}$, which has non-zero entries for the final 3 columns, allowing non-zero corresponding θ values.

Restating the procedure, successful documents are generated by a mixture of the usage topics *only*, while failing documents are generated by a mixture of *both* the usage topics and buggy topics.

We then try to recover the true topics with standard LDA, using the correct number of ground truth topics ($T = 11$), and modeling failed runs only. Unfortunately, the more prominent usage patterns overwhelm the buggy ones (Figure 4.2c), resulting in duplications of the usage topics.

(a) Ground truth usage and buggy topics used to generate the synthetic dataset.

(b) Example successful (left) and failing (right) runs generated by the synthetic data model.

(c) Topics recovered by standard LDA, which does not recover "weak" buggy topics.

(d) Topics recovered by ΔLDA, which exploits outcome labels to recover buggy topics.

Figure 4.2: The "buggy bars" synthetic dataset.

Note that we have side-information in the form of program failure or success for each individual run. The ΔLDA model uses this side-information to enhance detection of buggy behavior patterns by partitioning the set of T topics into a set of usage topics T_u and a set of buggy topics T_b. Similar to the synthetic example, the usage topics are *shared* by all runs, and are meant to capture behavior patterns which are related to normal program execution. The buggy topics are *restricted* to only appear in the failing runs, and are therefore meant to capture *failure-specific* behavior patterns.

In order to capture our notions of shared and restricted topics, it is necessary to extend the base LDA model. We achieve this restriction in ΔLDA by the use of an additional observed outcome variable $o \in \{s, f\}$, which selects between separate document-topic Dirichlet priors $\alpha^{(s)}$

(success) and $\alpha^{(f)}$ (failure) for each individual run. For topics t in the restricted set of buggy topics T_b, we enforce that $\alpha_t^{(s)} = 0$ and $\alpha_t^{(f)} > 0$. This corresponds to the α shown in (4.1) and effectively prohibits the buggy topics from appearing in successful runs, as we intended. Figure 4.3 shows the ΔLDA graphical model, reflecting these modifications with an additional observed o node selecting between the separate $\alpha^{(s)}$ and $\alpha^{(f)}$ hyperparameters, which in turn influence the usage of the separate usage and buggy topic sets. That is, the probabilistic model is identical to standard LDA, except that the document outcome label o determines whether the document-topic multinomial θ is drawn from a Dirichlet with hyperparameter $\alpha^{(s)}$ (which has zeros for the T_b buggy topics) or $\alpha^{(f)}$ (which has non-zero values for the T_b buggy topics). Letting o_j be the outcome variable associated with document j, the joint probability distribution is given by

$$P(\mathbf{w}, \mathbf{z}, \phi, \theta \mid \alpha, \beta, \mathbf{d}, \mathbf{o}) \propto \left(\prod_t^T p(\phi_t|\beta)\right)\left(\prod_j^D p(\theta_j|\alpha^{(o_j)})\right)\left(\prod_i^N \phi_{z_i}(w_i)\theta_{d_i}(z_i)\right) \tag{4.2}$$

where, as mentioned above, only the $p(\theta_j|\alpha^{(o(d))})$ term has changed from standard LDA.

We derive the collapsed Gibbs sampling and (ϕ, θ) estimation equations under the new ΔLDA model (see Appendix A). We find that the resulting equations are identical to their standard LDA counterparts from Chapter 2, except that we substitute the appropriate value of α ($\alpha^{(s)}$ or $\alpha^{(f)}$) depending on the document outcome flag o.

For the synthetic buggy bars dataset, the topics recovered by ΔLDA are shown in Figure 4.2d. The addition of failure or success side information and the use of both shared and exclusive topics allow ΔLDA to successfully recover the buggy patterns.

4.3 Experiments

Since we assume the success or failure of each individual run is observed, we are not interested in predicting whether or not a program will crash or behave incorrectly. Instead, we are interested in understanding failing runs in two ways. First, can we use our special buggy topics to cluster runs by root cause of failure? Second, can we examine the high-probability predicates in the buggy

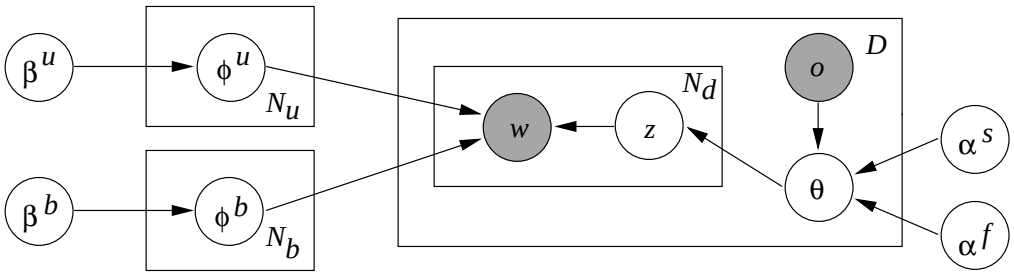


Figure 4.3: The ΔLDA graphical model, with additional observed document outcome label o selecting between separate "success" (α^s) or "failure" (α^f) values for the hyperparameter α. The α hyperparameter then controls the usage of the restricted "buggy" topics ϕ^b which are separated out from the shared "usage" topics ϕ^u.

topics to find the root causes of failure within the code itself? Both of these criteria are directly motivated by the potential for useful application in real-world debugging efforts.

Our dataset consists of CBI-instrumented runs from 4 different programs: `exif` [EXIF Tag Parsing Library,], `grep` [H. Do, 2005, Rothermel et al., 2006], `gzip` [H. Do, 2005, Rothermel et al., 2006], and `moss` [Schleimer et al., 2003]. Table 4.2 summarizes important properties of these datasets: the number of lines of code in the original program, the number of instrumentation predicates for which we have observed counts, how many different successful and failing runs we observe, and how many usage and bug topics we use in our model.

These programs contain *known* bugs, resulting from a mix of actual development and researcher "sabotage" (i.e., intentionally adding bugs for research purposes). In addition to the subsampling that is intrinsic to CBI instrumentation, we also discard uninformative predicates (e.g., predicates which are *always* or *never* triggered during execution). Furthermore, document lengths are "normalized" to 1000 by estimating a multinomial from the original counts and taking 1000 samples.

For each experiment we use fixed hyperparameters $\beta = 0.1$ and $\alpha = 0.5$, setting $\alpha = 0$ for buggy topics in successful runs as described earlier. The number of buggy topics for each program is chosen to be the ground truth number of bugs, and the number of usage topics is chosen to be equal to the number of "use cases" for each program (i.e., how many distinct modes of operation

Table 4.2: Statistical debugging datasets. For each program, the number of bug topics is set to the ground truth number of bugs.

			Runs		Topics	
Program	Lines of Code	Predicate Types	Successful	Failing	Usage	Bug
`exif`	10,611	20	352	30	7	2
`grep`	15,721	2,071	609	200	5	2
`gzip`	8,960	3,929	29	186	5	2
`moss`	35,223	1,982	1727	1228	14	8

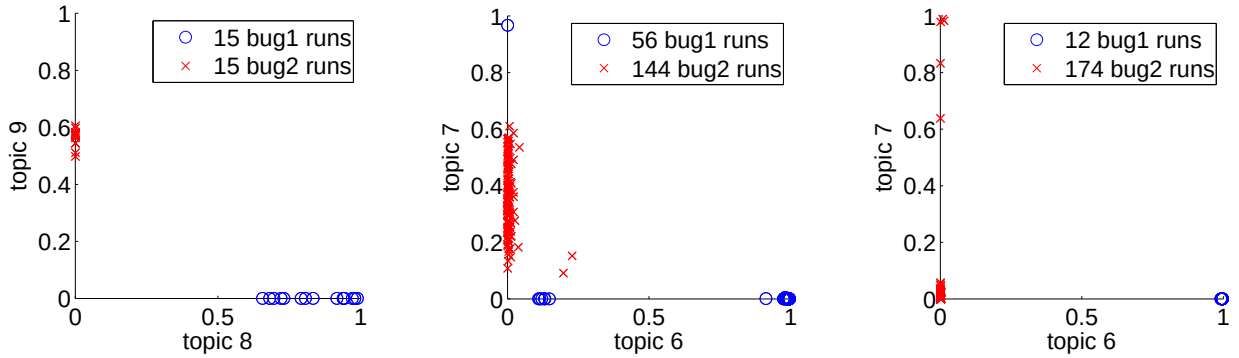
are indicated by the command-line flags). The Gibbs sampler is run for 2000 iterations before estimating ϕ and θ from the final sample.

4.3.1 Clustering runs by cause of failure

If there are multiple bugs present, it will be difficult to begin debugging without first isolating the effects of the underlying bugs from one another. Because of this, the first goal is to partition failing runs by *which* bug caused the undesirable behavior. We cluster failed runs by their document-topic mixing weights $\theta^{(d)}$, partitioning on $\arg\max_{z \in T_b} \theta_z^{(d)}$. We set $|T_b|$ equal to the true number of bugs present for each program, which is known for these controlled experiments. Our clusterings are evaluated against ground truth using the Rand Index [Rand, 1971] for agreement between partitions, where a value of 1 indicates total agreement and 0 indicates total disagreement. These results are found to be competitive with two previous statistical debugging approaches, which we refer to as *PLDI05* [Liblit et al.,] and *ICML06* [Zheng et al., 2006] (Table 4.3).

To aid in our visual understanding of this partitioning, we show runs plotted by $\theta_z^{(d)}$ for $z \in T_b$ and labeled by their ground truth cause of failure in Figures 4.4a, 4.4b, and 4.4c. Visually, these axis-aligned plots demonstrate that runs which suffer different root causes of failure are "explained" by different buggy topics. In Figure 4.5 we have more than three bug topics, so we apply a standard dimensionality reduction technique, Principal Components Analysis (PCA)

[Hastie et al., 2001], in order to facilitate interpretation. These plots all provide compelling visual evidence of the desired alignment between our learned buggy topics and true cause of failure for each program run.



(a) `exif` bug topics (b) `grep` bug topics (c) `gzip` bug topics

Figure 4.4: Bug topics vs true bugs for ΔLDA.

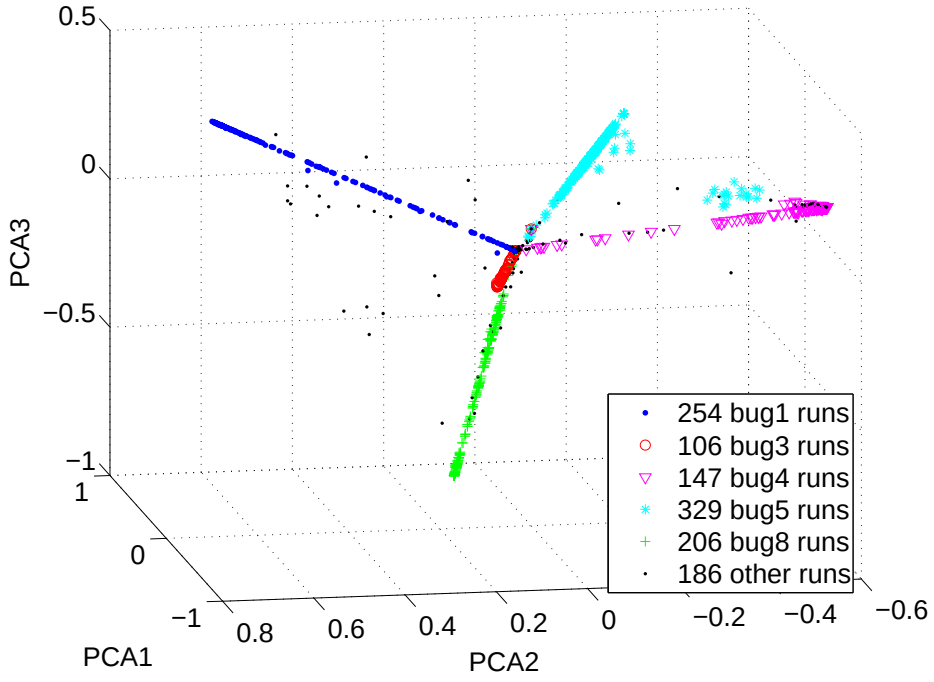


Figure 4.5: `moss` bug topics (with PCA).

Table 4.3: Rand indices showing similarity between computed clusterings of failing runs and true partitionings by cause of failure (complete agreement is 1, complete disagreement is 0.0).

	ΔLDA	ICML06	PLDI05
`exif`	1.00	0.88	1.00
`grep`	0.97	0.71	0.77
`gzip`	0.89	1.00	1.00
`moss`	0.93	0.93	0.96

4.3.2 Identifying root causes of failure

A statistical debugging system could facilitate the debugging process by providing the user with one or more "suspicious" predicates strongly associated with the failing runs. The second goal is therefore to identify predicates which will help the user to discover the bug itself. Hopefully, examining these predicates will provide the user with clues regarding the nature of the actual bug.

We rely upon expert assessments to determine the usefulness of significant predicates within each buggy topic. For each buggy topic z, the predicates w were ranked by calculating $P(z|w)$ from $P(w|z)$ using Bayes rule. This quantity was used instead of $\phi_z(w) = P(w|z)$ to avoid prevalent but uninformative w which have high probability across all topics. For each topic $z = i$ and word $w = j$ we define a confidence score $S_{ij} = \min_{k \neq i} P(z = i|w = j) - P(z = k|w = j)$. For each topic $z = i$ we present words ranked by S_{ij}, omitting words for which S_{ij} is less than zero. Similar lists are formulated for the baseline methods ICML06 and PLDI05.

Evaluation of these ranked lists against the true bugs found ΔLDA to generally be superior to the ICML06 baseline, and roughly comparable to the PLDI05 baseline. For each bug, highly ranked predicates tend to occur in either "smoking gun" code which is a primary cause of failure, or in closely related code where secondary effects of the bug manifest. For some difficult cases (e.g., the true cause of failure was not covered by instrumentation), all methods failed to clearly identify the root cause of failure.

An interesting aspect of ΔLDA is that the usage topics often correspond to well-defined patterns of program operation in interesting ways. For example, in `gzip` we observed that Topic 3 placed high probability on predicates located in functions associated with the fast compression variant of the algorithm, and that Topic 3 itself had high probability in program runs which used a command-line flag explicitly calling for the faster compression algorithm. Expert evaluations discovered meaningful patterns in the usage topics for other programs as well. This suggests that it may be possible to identify *correlations* between usage topics and buggy topics [Blei and Lafferty, 2006a], which could provide additional insights.

Later work [Dietz et al., 2009] models the *path* of execution taken through source code, outperforming ΔLDA on the localization of defective code within programs. However, the full trace information required for this approach is significantly more expensive to obtain than the subsampled counts produced by CBI. Extending ΔLDA to take advantage of richer trace information remains a potential avenue for future work.

4.4 Discussion

One advantage of ΔLDA is that the scores produced by PLDI05 have no meaningful probabilistic interpretation, limiting their use in further analysis. Potential extensions could put the probabilistic topics to practical use in debugging. Other software engineering work [Linstead et al., 2007] applied Author-Topic models [Rosen-Zvi et al., 2004] directly to source code. Learned topics are used to map different portions of a program to the developers with the most relevant expertise. This type of analysis could be combined with the learned topics suggested by ΔLDA in order to intelligently assign developers to bugs.

Furthermore, the unique nature of software admits much richer classes of prior knowledge. In particular, *static analysis* of code [Liblit, 2008] has the potential to reveal interesting relationships among predicates instrumented by the CBI [Liblit, 2007]. The *backwards slice* [Horwitz et al., 1988] from a given line of code L finds all other lines of code which *could have influenced* the outcome of L. If we know the actual point of failure F from a given program run, it then stands to reason that code which lies in the backwards slice of F is more likely to contain the

root cause of the failure and therefore good bug-predictors. In addition, software researchers have discovered various factors associated with buggy code, including recent change activity and code complexity [Graves et al., 2000, Munson and Khoshgoftaar, 1992]. This information could also be profitably incorporated into our model. This type of word-level (i.e., ϕ-side) domain knowledge could potentially be encoded into other models developed as part of this thesis (Chapters 5 and 7).

The principles of *restricted* and *shared* topics developed in ΔLDA can also be applied to natural language processing [Lacoste-Julien et al., 2008]. For example, user generated "tags"[1] can be combined with ΔLDA-like mechanisms [Ramage et al., 2009] to learn special topics related to the tags. The idea of using shared or background components to enhance topic recovery was also previously used to assign topics to broadcast news stories [Sista et al., 2002].

4.5 Summary

This chapter has presented the ΔLDA model and its successful application to the task of statistical debugging. The division between a set of shared topics common to all documents and a set of special topics restricted to appear in specially labeled documents represents an interesting and useful type of domain knowledge. In statistical debugging, we have the knowledge that certain buggy behavior patterns should manifest themselves in failing runs only, while another common set of standard program behavior should appear in all runs. Exploiting this knowledge allows us to both cluster failing runs by bug and to gain actionable insights into the nature of the bugs themselves. The general idea of ΔLDA is also applicable to the text domain, where document labels can come from a variety of sources such as user-generated annotations.

[1] Such as those found on the bookmark sharing site http://www.delicious.com.

KEY IDEAS

- ◇ ΔLDA incorporates document label information into the document-topic distributions (θ-side)
  - ▷ *shared* topics appear in all documents
  - ▷ *restricted* topics appear only in specially labeled documents.
- ◇ The restricted topics capture patterns present only in the labeled documents.
- ◇ Topic modeling can be a useful tool in the statistical debugging domain.

Chapter 5

Topic-in-set knowledge

This chapter describes a mechanism for adding "partial" supervision to LDA. Topic-in-set knowledge [Andrzejewski and Zhu, 2009] allows the user to specify a z-label for each observed word in the corpus. A z-label for observed word w_i consists of a set $C^{(i)}$ of possible values for the corresponding latent topic index z_i, and can be thought of as a (hard or soft) constraint. The probabilities of latent topic assignments $\mathbf{z}$ which violate these constraints are then penalized for a soft constraint, or set to zero in the case of a hard constraint. By modifying the topic probabilities in this way, this extension can be considered a θ-side modification of LDA as discussed in Chapter 3.

By carefully choosing these $C^{(i)}$, the user can encode various types of domain knowledge into the model. A user could provide word sense labels by forcing two occurrences of the same word (e.g., "Apple pie" and "Apple iPod") to be explained by different topics. In this work we explore an application where a domain expert provides "seed" words for a concept of interest, which are then used to learn a topic built around the concept.

5.1 Collapsed Gibbs sampling with z-labels

The z-label constraints are enforced via a multiplicative penalty in the collapsed Gibbs sampling equation. We recall the sampling equation for standard LDA [Griffiths and Steyvers, 2004]

$$P(z_i = v|\mathbf{z}_{-i}, \mathbf{w}, \alpha, \beta) \propto \left(\frac{n_{-i,v}^{(d)} + \alpha}{\sum_u^T (n_{-i,u}^{(d)} + \alpha)}\right)\left(\frac{n_{-i,v}^{(w_i)} + \beta}{\sum_{w'}^W (n_{-i,v}^{(w')} + \beta)}\right) \tag{5.1}$$

where $n_{-i,v}^{(d)}$ is the number of times topic v is used in document d, and $n_{-i,v}^{(w_i)}$ is the number of times word w_i is generated by topic v. The $-i$ notation signifies that the counts are taken omitting the

value of z_i. For convenience, we define q_{iv} to be this unnormalized collapsed Gibbs sampling probability of assigning $z_i = v$.

$$q_{iv} = \left(\frac{n_{-i,v}^{(d)} + \alpha}{\sum_u^T (n_{-i,u}^{(d)} + \alpha)} \right) \left(\frac{n_{-i,v}^{(w_i)} + \beta}{\sum_{w'}^W (n_{-i,v}^{(w')} + \beta)} \right). \tag{5.2}$$

Next, let $C^{(i)}$ be the set of possible z-labels for latent topic z_i. We set a hard constraint by modifying the Gibbs sampling equation with an indicator function $\delta(v \in C^{(i)})$, which takes on value 1 if $v \in C^{(i)}$ and is 0 otherwise:

$$P(z_i = v | \mathbf{z}_{-i}, \mathbf{w}, \alpha, \beta) \propto q_{iv} \delta(v \in C^{(i)}) \tag{5.3}$$

If we wish to restrict z_i to a single value (e.g., $z_i = 5$), this can be accomplished by setting $C^{(i)} = \{5\}$. Likewise, we can restrict z_i to a subset of values $\{1, 2, 3\}$ by setting $C^{(i)} = \{1, 2, 3\}$. Finally, for unconstrained z_i we simply set $C^{(i)} = \{1, 2, ..., T\}$, in which case our modified sampling (5.3) reduces to the standard Gibbs sampling (5.1).

Note that this formulation gives us a highly flexible method for inserting prior domain knowledge into the inference of latent topics, allowing us to set $C^{(i)}$ independently for every single word w_i in the corpus. This effect would be impossible to achieve by setting certain topic-specific asymmetric β vectors to zero, as is done with α in the ΔLDA model (Chapter 4, [Andrzejewski et al., 2007]).

This hard constraint model could also be relaxed. Let $0 \leq \eta \leq 1$ be the strength of our constraint, where $\eta = 1$ recovers the hard constraint (5.3) and $\eta = 0$ recovers unconstrained sampling (5.1). Then we can modify the Gibbs sampling equation as follows:

$$P(z_i = v | \mathbf{z}_{-i}, \mathbf{w}, \alpha, \beta) \propto q_{iv} \left(\eta \delta(v \in C^{(i)}) + 1 - \eta \right). \tag{5.4}$$

5.2 Experiments

5.2.1 Concept expansion

We now demonstrate some example applications for this type of partial supervision. For these experiments, symmetric hyperparameters $\alpha = 0.5$ and $\beta = 0.1$ are used and all MCMC chains are run for 2000 samples before estimating ϕ and θ from the final sample.

The first experiment uses z-labels to do concept expansion, where we begin with some seed terms of a predefined concept and we wish to use topic modeling to expand the concept and find other related terms. For example, a biological expert may be interested in the concept *translation*. The expert can then provide a set of seed words which are strongly related to this concept; here we assume the seed word set $\{$translation, trna, anticodon, ribosome$\}$. We add the hard constraint that $z_i = 0$ for all occurrences of these four words in our corpus of approximately 9,000 yeast-related abstracts from PubMed.

We run LDA with the number of topics $T = 100$, both with and without the z-label knowledge on the seed words. Table 5.1 shows the most probable words in selected topics from both runs. Table 5.1a shows Topic 0 from the constrained run, while Table 5.1b shows the topics which contained seed words among the top 50 most probable words from the unconstrained run.

These top words are annotated for relevance to the target concept (translation) by an outside biological expert. The words in Table 5.1 are then bolded if they are one of the original seed words, italicized if they are judged as relevant, and left undecorated otherwise. From a quick glance, we can see that Topic 0 from the constrained run contains more relevant terms than Topic 43 from the standard LDA run. Topic 31 has a similar number of relevant terms, but taken together we can see that the emphasis of Topic 31 is slightly off-target, more focused on *mRNA turnover* than *translation*. Likewise, Topic 73 seems more focused on the ribosome itself than the process of translation. Overall, these results demonstrate the effectiveness of z-label information for guiding topic models towards a user-seeded concept.

Table 5.1: Standard LDA and z-label topics learned from a corpus of PubMed abstracts, where the goal is to learn topics related to *translation*. Concept seed words are bolded, other words judged relevant to the target concept are italicized.

(a) Topic 0 with z-label

Topic 0	**translation**, *ribosomal*, **trna**, *rrna*, *initiation*, **ribosome**, *protein*, *ribosomes*, is, *factor*, *processing*, *translational*, *nucleolar*, *pre-rrna*, *synthesis*, small, *60s*, eukaryotic, *biogenesis*, *subunit*, *trnas*, *subunits*, large, *nucleolus*, *factors*, 40, synthetase, free, *modification*, *rna*, depletion, *eif-2*, *initiator*, *40s*, *ef-3*, **anticodon**, *maturation*, *18s*, *eif2*, *mature*, *eif4e*, associated, synthetases, *aminoacylation*, *snornas*, *assembly*, *eif4g*, *elongation*

(b) Standard LDA Topics

Topic 31	*mrna*, **translation**, *initiation*, *mrnas*, *rna*, *transcripts*, 3, *transcript*, *polya*, *factor*, *5*, *translational*, *decay*, *codon*, *decapping*, *factors*, *degradation*, *end*, *termination*, eukaryotic, *polyadenylation*, *cap*, required, efficiency, *synthesis*, show, *codons*, abundance, *rnas*, *aug*, *nmd*, *messenger*, *turnover*, *rna-binding*, *processing*, *eif2*, *eif4e*, *eif4g*, cf, occurs, *pab1p*, *cleavage*, *eif5*, cerevisiae, major, *primary*, rapid, *tail*, efficient, upf1p, *eif-2*
Topic 43	type, is, wild, yeast, **trna**, synthetase, both, *methionine*, synthetases, class, *trnas*, *enzyme*, whereas, *cytoplasmic*, because, direct, efficiency, presence, *modification*, *aminoacylation*, **anticodon**, either, eukaryotic, between, different, specific, discussed, results, similar, some, *met*, compared, *aminoacyl-trna*, able, *initiator*, *sam*, not, free, however, *recognition*, several, arc1p, fully, same, forms, leads, identical, responsible, found, only, well
Topic 73	*ribosomal*, *rrna*, *protein*, is, *processing*, **ribosome**, *ribosomes*, *rna*, *nucleolar*, *pre-rrna*, *rnase*, small, *biogenesis*, depletion, *subunits*, *60s*, *subunit*, large, *synthesis*, *maturation*, *nucleolus*, associated, essential, *assembly*, components, **translation**, involved, *rnas*, found, component, *mature*, *rp*, *40s*, accumulation, *18s*, 40, particles, *snornas*, *factors*, *precursor*, during, *primary*, *rrnas*, *35s*, has, *21s*, specifically, results, *ribonucleoprotein*, early

5.2.2 Concept exploration

For our next experiment, we suppose that a user has chosen a set of terms and wishes to discover different topics related to these terms. By constraining these terms to only appear in a restricted set of topics, these terms will be *concentrated* in the set of topics. This modification may then have indirect effects on the topic assignments of non-seed words, resulting in a significantly different set of recovered topics.

To make this concrete, say we are interested in the location *United Kingdom*. We have the Reuters newswire corpus used for the CoNLL-2003 shared task [Tjong Kim Sang and De Meulder, 2003]. This corpus contains manually annotated named entity labels for persons (PER), locations (LOC), organizations (ORG), and miscellaneous (MISC). In order to incorporate this information into our analysis, we pre-pend each tagged token with its entity tag (e.g., "Saddam Hussein" becomes "[PER]Saddam [PER]Hussein"). We use this additional information by seeding our *United Kingdom* topics with the following location-tagged terms {britain, british, england, uk, u.k., wales, scotland, london}. Location-tagged occurrences (e.g., "[LOC]England") of these terms are then restricted to appear only in the first three topics. In order to focus on our target location, we also restrict all other location-tagged tokens to *not* appear in the first three topics. For this experiment we set $T = 12$, arrived at by trial-and-error in the baseline (standard LDA) case.

The 50 most probable words for each topic are shown in Table 5.2 and Table 5.3, and tagged entities are prefixed with their tags for easy identification. Table 5.2 shows the top words for the first three topics of our z-label run. These three topics are all related to the target LOCATION *United Kingdom*, but they also split nicely into *business*, *cricket*, and *soccer*. Words which are highly relevant to each of these three concepts are bolded, italicized, and underlined, respectively.

In contrast, in Table 5.3 we show three topics from standard LDA which contain any of the "United Kingdom" location terms (which are boxed) among the 50 most probable words for that topic. We make several observations about these topics. First, standard LDA Topic 0 is mostly concerned with political unrest in Russia, which is not particularly related to the target location. Second, Topic 2 is similar to our previous business topic, but with a more US-oriented slant. Note

that "dollar" appears with high probability in standard LDA Topic 2, but not in our z-label LDA Topic 0. Standard LDA Topic 8 appears to be a mix of both soccer and cricket words. Therefore, it seems that our topic-in-set knowledge helps in distilling topics related to the seed words.

Given this promising result, we attempted to repeat this experiment with some other nations (United States, Germany, China), but without much success. When we tried to restrict these location words to the first few topics, these topics tended to be used to explain other concepts unrelated to the target location (often other sports). Germany and China location words simply do not occur frequently enough in the corpus for z-label-LDA to "build" topics around them. While the United States location words are more frequent, the results were also not very interesting.

5.3 Principled derivation

We have discussed z-label-LDA in purely procedural terms as a modification to the Gibbs sampling algorithm. However, it is possible to derive z-label-LDA as a principled extension to standard LDA by first converting LDA to an undirected graphical model and then adding clique potentials which encode the penalties for violating z-labels. This notion is generalized further in the LogicLDA model; we defer discussion of further details to Chapter 7.

5.4 Summary

This chapter has described Topic-in-set knowledge, a simple yet powerful type of domain knowledge for topic modeling. Topic-in-set knowledge allows the user to influence the topic assignment z_i of specific words in the corpus. By choosing a handful of "seed" words and constraining their corresponding latent topics to a single topic, LDA can be encouraged to build a topic generalizing beyond the provided seed words.

Table 5.2: z-label topics learned from an entity-tagged corpus of Reuters newswire articles. Topics shown contain location-tagged terms from our *United Kingdom* location term list. Entity-tagged tokens are pre-pended with their tags: PER for person, LOC for location, ORG for organization, and MISC for miscellaneous. Words related to business are bolded, cricket italicized, and soccer underlined.

Topic 0	**million**, **company**, 's, year, **shares**, **net**, **profit**, half, group, **[ORG]corp**, **market**, **sales**, **share**, **percent**, expected, **business**, **loss**, **stock**, **results**, **forecast**, **companies**, **deal**, **earnings**, **statement**, **price**, [LOC]london, **billion**, [ORG]newsroom, **industry**, newsroom, **pay**, **pct**, **analysts**, issue, **services**, **analyst**, **profits**, **sale**, **added**, **firm**, [ORG]london, **chief**, **quarter**, **investors**, **contract**, note, **tax**, **financial**, months, **costs**
Topic 1	[LOC]england, [LOC]london, [LOC]britain, *cricket*, [PER]m., *overs*, *test*, *wickets*, scores, [PER]ahmed, [PER]paul, [PER]wasim, *innings*, [PER]a., [PER]akram, [PER]mushtaq, day, *one-day*, [PER]mark, final, [LOC]scotland, [PER]waqar, *[MISC]series*, [PER]croft, [PER]david, [PER]younis, match, [PER]ian, total, [MISC]english, [PER]khan, [PER]mullally, *bat*, declared, fall, [PER]d., [PER]g., [PER]j., *bowling*, [PER]r., [PER]robert, [PER]s., [PER]steve, [PER]c. *captain*, golf, tour, [PER]sohail, extras, [ORG]surrey
Topic 2	soccer, division, results, played, standings, league, matches, halftime, goals, attendance, points, won, [ORG]st, drawn, saturday, [MISC]english, lost, premier, [MISC]french, result, scorers, [MISC]dutch, [ORG]united, [MISC]scottish, sunday, match, [LOC]london, [ORG]psv, tabulate, [ORG]hapoel, [ORG]sydney, friday, summary, [ORG]ajax, [ORG]manchester, tabulated, [MISC]german, [ORG]munich, [ORG]city , [MISC]european, [ORG]rangers, summaries, weekend, [ORG]fc, [ORG]sheffield, wednesday, [ORG]borussia, [ORG]fortuna, [ORG]paris, tuesday

Table 5.3: Standard LDA topics for an entity-tagged corpus of Reuters newswire articles. Topics shown contain location-tagged terms from our *United Kingdom* location term list. Entity-tagged tokens are pre-pended with their tags: PER for person, LOC for location, ORG for organization, and MISC for miscellaneous. Words related to business are bolded, cricket italicized, and soccer underlined.

Topic 0	police, 's, people, killed, [MISC]russian, friday, spokesman, [LOC]moscow, told, rebels, group, officials, [PER]yeltsin, arrested, found, miles, km, [PER]lebed, capital, thursday, tuesday, [LOC]chechnya, news, saturday, town, authorities, airport, man, government, state, agency, plane, reported, security, forces, city, monday, air, quoted, students, region, area, local, [LOC]russia, [ORG]reuters, military, [LOC]london, held
Topic 2	**percent**, 's, **market**, thursday, july, **tonnes**, week, year, lower, [LOC]u.s., **rate**, **prices**, **billion**, **cents**, **dollar**, friday, **trade**, **bank**, closed, **trading**, higher, close, **oil**, **bond**, **fell**, **markets**, **index**, **points**, **rose**, **demand**, june, **rates**, september, **traders**, [ORG]newsroom, day, **bonds**, **million**, **price**, **shares**, **budget**, government, **growth**, **interest**, monday, [LOC]london, **economic**, august, expected, **rise**
Topic 5	's, match, team, win, play, season, [MISC]french, lead, home, year, players, [MISC]cup, back, minutes, champion, victory, time, n't, game, saturday, title, side, set, made, wednesday, [LOC]england, league, run, club, top, good, final, scored, coach, shot, world, left, [MISC]american, captain, [MISC]world, goal, start, won, champions, round, winner, end, years, defeat, lost
Topic 8	division, [LOC]england, soccer, results, [LOC]london, [LOC]pakistan, [MISC]english, matches, played, standings, league, points, [ORG]st, *cricket*, saturday, [PER]ahmed, won, [ORG]united, goals, [PER]wasim, [PER]akram, [PER]m., [MISC]scottish, [PER]mushtaq, drawn, *innings*, premier, lost, [PER]waqar, *test*, [PER]croft, [PER]a., [PER]younis, declared, *wickets*, [ORG]hapoel, [PER]mullally, [ORG]sydney, day, [ORG]manchester, [PER]khan, final, *scores*

KEY IDEAS

- ◇ Topic-in-set knowledge allows the user to constrain individual z_i.
- ◇ This mechanism can be used to build topics around "seed" words.

Chapter 6

Dirichlet Forest priors

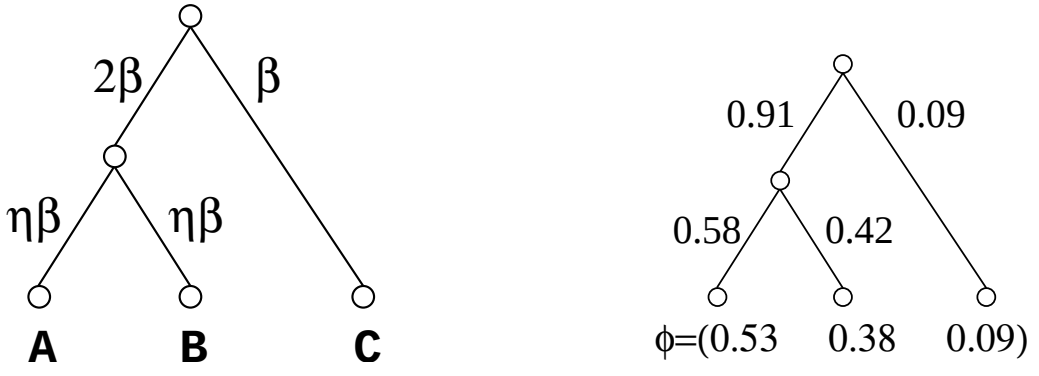
In standard LDA, the topic-word multinomial distributions $\phi_z = P(w|z)$ are drawn from a Dirichlet prior with hyperparameter β. Being conjugate to the multinomial [Gelman et al., 2004], the Dirichlet distribution is a mathematically convenient prior. To some extent, a user can encode domain knowledge into this Dirichlet prior by setting the values in the β hyperparameter vector. The values can be roughly thought of as psuedocounts, so a user can encode a belief that the word "dog" is much more likely in a topic than the word "cat" by setting $\beta_{dog} \gg \beta_{cat}$.

The work presented in this chapter replaces the standard Dirichlet prior on topic-word multinomial distributions $\phi_z = P(w|z)$ with a more expressive Dirichlet Forest Prior (DF) [Andrzejewski et al., 2009]. In the framework of Chapter 3, the resulting model is a ϕ-side variant. This extension allows the user to express rich, relational beliefs about the probabilities associated with different words. For example, we may wish to say that for each topic z we should have $P(w = dog|z) \approx P(w = cat|z)$, meaning that "dog" and "cat" should tend to have very similar probabilities within any given topic. Or we may wish to encode the opposite belief: for each topic z we should not have both $P(w = dog|z)$ and $P(w = cat|z)$ be large, meaning that "dog" and "cat" should never co-occur among the "Top N" most probable words for any given topic. Borrowing names from the constrained clustering literature [Basu et al., 2008], we call these preferences Must-Link and Cannot-Link, respectively. As we will demonstrate, these beliefs cannot be encoded into a standard Dirichlet prior, motivating the development of the novel Dirichlet Forest Prior. This prior can be used within LDA, yielding Dirichlet Forest Latent Dirichlet Allocation (DF-LDA).

6.1 Encoding Must-Link

While the Dirichlet prior has the advantage of conjugacy to our topic-word multinomial model, it is restricted in the sense that all variables (i.e., word probabilities) share a common variance parameter and are mutually independent except for the constraint that they must sum to 1 [Mosimann, 1962]. These limitations prevent us from encoding the Must-Link preference between a pair of words.

The Dirichlet Tree distribution [Dennis III, 1991, Minka, 1999] reparameterizes and generalizes the standard Dirichlet distribution, while maintaining conjugacy to the multinomial. In the Dirichlet Tree, the leaf nodes correspond to the multinomial probabilities. The root node is assigned probability mass 1, which then "flows" to its children in proportion to a sample from a Dirichlet distribution parametrized by the outgoing edge weights. Each internal node then distributes the probability mass it receives to its children in the same way. Since we begin with mass 1, and the Dirichlet random variables governing redistribution to children are non-negative and sum to 1, it is clear that the values which end up at the leaves will form a valid probability distribution. Figure 6.1a shows an example Dirichlet Tree over a vocabulary $\{A, B, C\}$, and Figure 6.1b shows the result of a sample from this Dirichlet Tree. Also, note that we can express the standard Dirichlet distribution as a Dirichlet Tree with depth 1 as shown in Figure 6.2.



(a) A Must-Link Dirichlet Tree.

(b) A single sample from the Must-Link Dirichlet Tree.

Figure 6.1: An example Dirichlet Tree along with example sampled values.

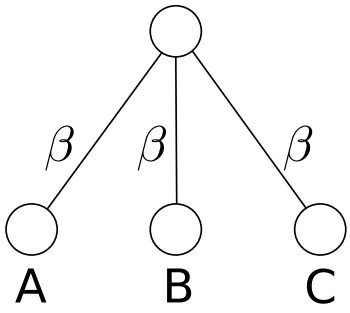


Figure 6.2: The standard Dirichlet as a Dirichlet Tree.

Formally, let $\gamma^{(k)}$ be the Dirichlet tree edge weight *leading into node* k. Let $C(k)$ be the immediate children of node k in the tree, L the leaves of the tree, I the internal nodes, and $L(k)$ the leaves in the subtree under k. To generate a sample $\phi \sim \text{DirichletTree}(\gamma)$, one first draws a multinomial at each internal node $s \in I$ from $\text{Dirichlet}(\gamma^{C(s)})$, i.e., using the weights from s to its children as the Dirichlet parameters. One can think of it as re-distributing the probability mass reaching s by this multinomial (initially, the mass is 1 at the root). The probability $\phi^{(k)}$ of a word $k \in L$ is then simply the product of the multinomial parameters on the edges from k to the root, as shown in Figure 6.1a. It can be shown [Dennis III, 1991] that this procedure gives $\text{DirichletTree}(\gamma)$

$$p(\phi|\gamma) = \left(\prod_{k}^{L} \phi^{(k)^{\gamma^{(k)}-1}}\right)\left(\prod_{s}^{I} \frac{\Gamma\left(\sum_{k}^{C(s)} \gamma^{(k)}\right)}{\prod_{k}^{C(s)} \Gamma\left(\gamma^{(k)}\right)} \left(\sum_{k}^{L(s)} \phi^{(k)}\right)^{\Delta(s)}\right) \tag{6.1}$$

where $\Gamma(\cdot)$ is the standard gamma function, and the notation $\prod_{k}^{L}$ means $\prod_{k \in L}$. The function $\Delta(s) \equiv \gamma^{(s)} - \sum_{k \in C(s)} \gamma^{(k)}$ is the difference between s's in-degree and out-degree. When this difference $\Delta(s) = 0$ for all internal nodes $s \in I$, the Dirichlet tree reduces to a Dirichlet distribution.

The Dirichlet tree distribution is conjugate to the multinomial, just like the Dirichlet distribution. It is possible to integrate out ϕ to get a distribution over word counts directly, similar to the multivariate Pólya distribution:

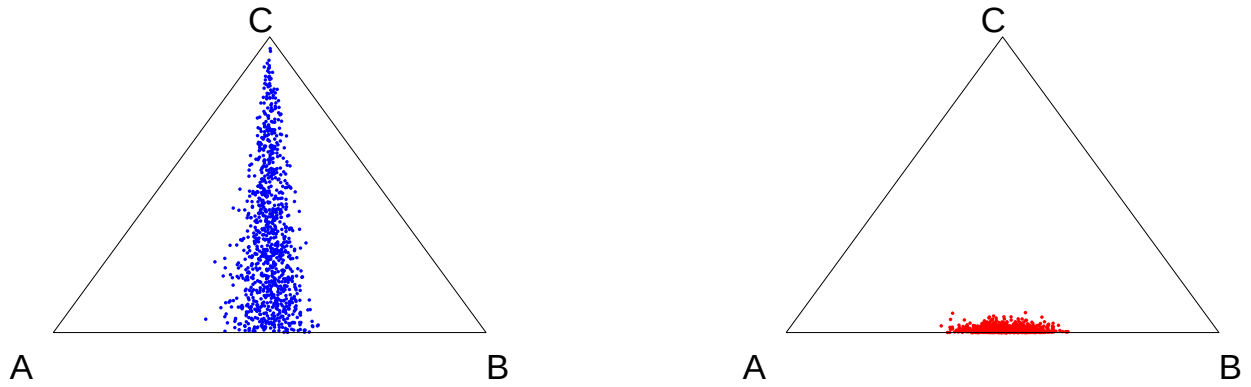
$$p(\mathbf{w}|\gamma) = \prod_s^I \left(\frac{\Gamma\left(\sum_k^{C(s)} \gamma^{(k)}\right)}{\Gamma\left(\sum_k^{C(s)} \left(\gamma^{(k)} + n^{(k)}\right)\right)} \prod_k^{C(s)} \frac{\Gamma\left(\gamma^{(k)} + n^{(k)}\right)}{\Gamma(\gamma^{(k)})} \right) \tag{6.2}$$

Here $n^{(k)}$ is the number of word tokens in $\mathbf{w}$ that appear in $L(k)$.

This more flexible structure gives us the capability to encode our Must-Link preference. Say that we wish to encode the preference Must-Link (A,B). Since each internal node distributes probability mass to its children according to its own Dirichlet, we can simply place words A and B together under an internal node with *very large* outgoing edge weights. This Dirichlet Tree is shown in Figure 6.1a, where the η parameter controls the "strength" of our Must-Link preference. The larger η is, the more uniformly the probability mass arriving at the internal node will be distributed to the A and B leaf nodes. Also, note that setting $\eta = 1$ will result in $\Delta(s) = 0$ for that internal node, yielding a standard Dirichlet distribution with symmetric parameter β.

To see how this actually works, we take samples from this Dirichlet Tree with $\beta = 1$ and $\eta = 50$. Figure 6.3a shows these samples plotted on the simplex where a point in the A corner signifies a $[1, 0, 0]$ sample, and a point on the centroid signifies $[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$. The distribution of the points indicates that $P(A) \approx P(B)$, as we had hoped. Importantly, $P(C)$ is still allowed to vary. Figure 6.3b shows an attempt to recover this behavior with a standard Dirichlet with asymmetric parameter vector $[50, 50, 1]$. While $P(A) \approx P(B)$, we can see that $P(C)$ is unintentionally constrained as well, resulting in the tight cluster of points at the bottom edge of the simplex.

More generally, notice that the semantics of our Must-Link definition are intrinsically transitive. That is, if we want $P(A) \approx P(B)$ and $P(B) \approx P(C)$, this implies that we want $P(A) \approx P(C)$. Therefore, to construct a Dirichlet Tree encoding all Must-Link preferences, we consider an undirected graph where words are nodes and pairwise Must-Link preferences are edges. We then take the transitive closure of this graph and, for each connected component, place the words under an internal node s with incoming edge weight $|L(s)|\beta$, where $|\cdot|$ represents the set size. The outgoing edges from s to the word leaves then each have weight $\eta\beta$. This procedure yields the Dirichlet Tree shown in Figure 6.1a. For $\eta = 1$, this preserves the standard Dirichlet since $\Delta(s) = 0$ for all internal nodes s, since each internal node s will have $|L(s)|$ outgoing edges with



(a) Samples from the Must-Link Dirichlet Tree with parameters $\beta = 1$ and constraint strength $\eta = 50$.

(b) Samples from a standard Dirichlet Must-Link attempt with parameters $\beta = [50, 50, 1]$.

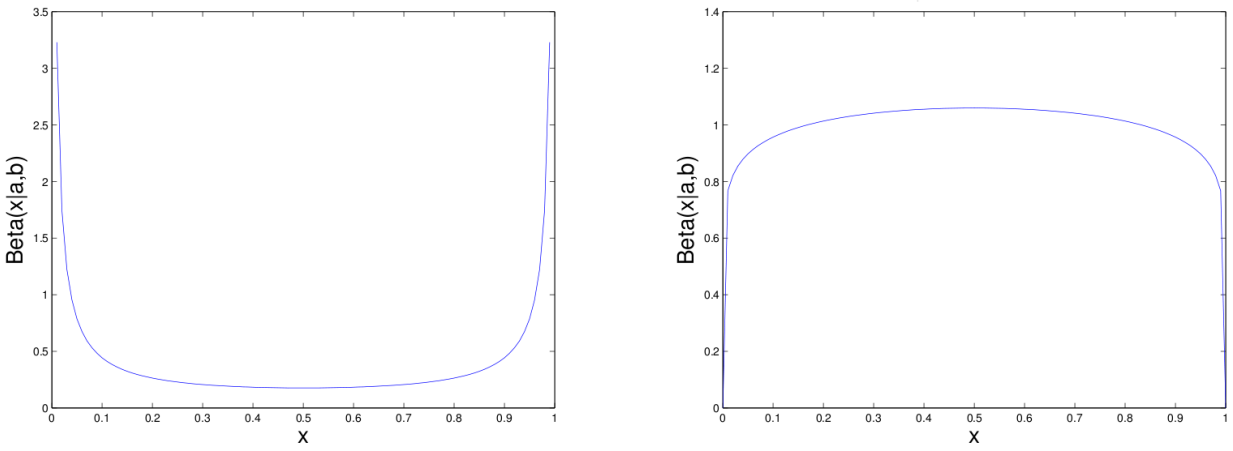
Figure 6.3: Simplex plots of multinomial distributions for Must-Link and standard Dirichlet.

weight β and 1 incoming edge with weight $|L(s)|\beta$. For $\eta > 1$, the re-distribution of probability mass at an internal node s will have increasing *concentration* $\eta\beta$ but the same all-1 base-measure. This tends to redistribute the mass evenly in the transitive closure represented by s, as shown in Figure 6.3a. Importantly, the mass *reaching* s is independent of η, and can still have a large variance. This properly encodes the fact that we want Must-Linked words to have similar, but not always large, probabilities. Otherwise, Must-Linked words would be forced to appear with large probability in *all* topics, which is the clearly undesirable scenario shown in our standard Dirichlet simplex Figure 6.3b.

6.2 Encoding Cannot-Link

We now turn to the more difficult case of Cannot-Link preferences. Since we were able to encode Must-Link by putting words under a low-variance internal node, can we simply put Cannot-Linked words beneath a high-variance internal node? The answer is no, for two reasons. First, Cannot-Link is not an inherently transitive preference. If we want $P(A)$ and $P(B)$ to never have large probabilities in the same topic, and likewise for $P(B)$ nad $P(C)$, it clearly does *not* follow that we want to prohibit $P(A)$ and $P(C)$ from having large probabilities in the same topic. Placing

connected components of the Cannot-Link graph transitive closure under internal nodes would therefore create unintended Cannot-Links between words. The second, and more subtle, issue is that a high-variance Dirichlet distribution is achieved by using very *small* parameter values. These small values can easily be overwhelmed by data, giving us a very *weak* prior. To illustrate this second issue, we examine a Beta distribution with parameters $a = b = 0.1$ in Figure 6.4a. This "spiky" distribution appears to fulfill our goal of putting high probability on outcomes where $P(A) \not\approx P(B)$. However, after observing a single A and a single B, our initial preference for $P(A) \not\approx P(B)$ has been overruled by the data, as the posterior shown in Figure 6.4b indicates.



(a) Beta distribution with $a = b = 0.1$

(b) Beta distribution with $a = b = 1.1$

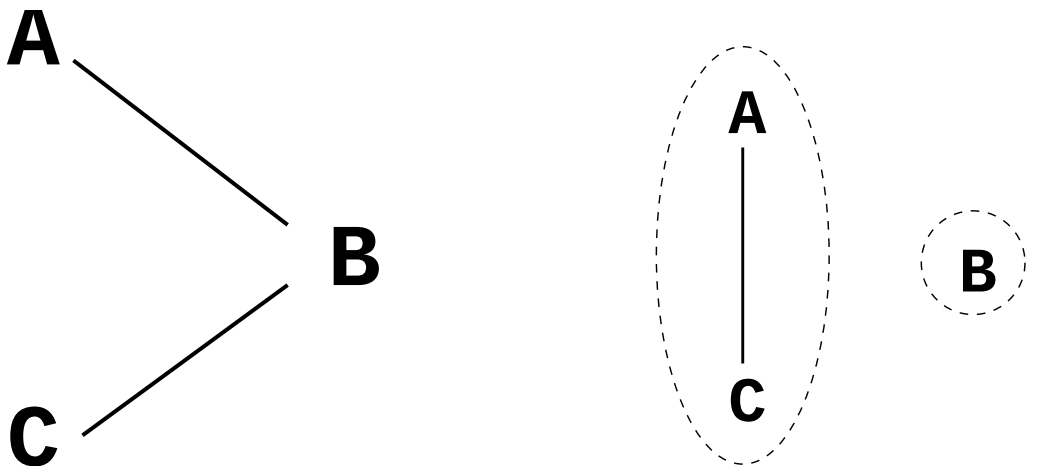
Figure 6.4: Example Beta distributions with different hyperparameters.

To build up our solution to this problem, we again form an undirected graph with words as nodes and Cannot-Link preferences as edges. If words are in the same connected component in the transitive closure of the Must-Link graph, they are considered to "collapse" to a single node in the Cannot-Link graph. This formalizes the notion that Must-Link (A,B) and Cannot-Link (B,C) imply Cannot-Link (A,C). Note that individual connected components of this graph are effectively independent of one another for the purposes of encoding our Cannot-Link preferences.

For example, Cannot-Link (A,B) and Cannot-Link (B,C) give us a Cannot-Link graph with a single connected component as shown in Figure 6.5a. We now consider this single connected

component r, and take the complement graph of this subgraph by removing all present edges and adding edges where none were present (Figure 6.5b).

Let there be $Q^{(r)}$ *maximal cliques* $M_{r1} \ldots M_{rQ^{(r)}}$ in this complement graph. Here $Q^{(r)} = 2$ with $M_{r1} = \{A, C\}$ and $M_{r2} = \{B\}$. In the following, we will simply call these "cliques", but it is important to remember that they are maximal cliques of the complement graph, not the original Cannot-Link-graph. These cliques are the alternative form to Cannot-Links, with the following semantics: Each clique (e.g., $M_{r1} = \{A, C\}$) is the maximal subset of words in the connected component that can "occur together". That is, these words are allowed to simultaneously have large probabilities (subject to normalization) in a given topic without violating any Cannot-Link preferences. By the maximality of these cliques, allowing any word outside the clique (e.g., "B") to also have a large probability will violate at least one Cannot-Link (in this example 2).



(a) A connected component of the Cannot-Link graph.

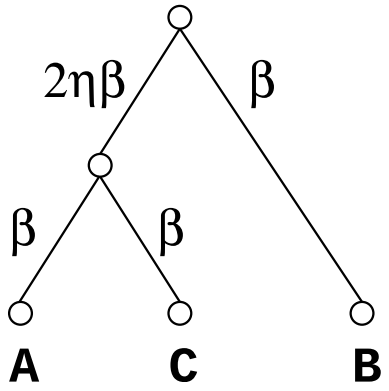
(b) The complement graph of this connected component.

Figure 6.5: Identifying maximal compatible cliques in the complement of the Cannot-Link graph.

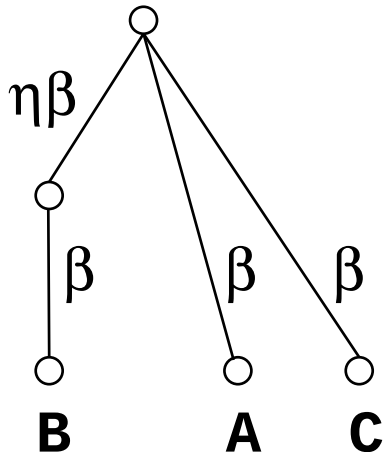
We now discuss the encoding of the Cannot-Link preferences for this connected component r in isolation. Our approach is to create a mixture model of $Q^{(r)}$ Dirichlet subtrees, one for each clique. For each topic, exactly one subtree is selected according to probability

$$P(r) \propto |M_{rq}|, \quad q = 1 \ldots Q^{(r)}. \tag{6.3}$$

Conceptually, the selected subtree indexed by q tends to redistribute nearly all probability mass to the words within M_{rq}. Since there is no mass left for other cliques, it is impossible for a word outside clique M_{rq} to have a large probability. Therefore, no Cannot-Link will be violated. In reality, the subtrees are soft rather than hard, because Cannot-Links are only preferences. The Dirichlet subtree for M_{rq} is structured as follows. The subtree's root connects to an internal node s with weight $\eta|M_{rq}|\beta$. The node s connects to words in M_{rq}, with weight β. The subtree's root also directly connects to words not in M_{rq} (but in the connected component r) with weight β. This will send most probability mass down to s, and then flexibly redistribute it among words in M_{rq}. For example, Figure 6.6a and Figure 6.6b show the Dirichlet subtrees for $M_{r1} = \{A, C\}$ and $M_{r2} = \{B\}$ respectively. Samples from this mixture model are shown in Figure 6.7, representing multinomials in which no Cannot-Link is violated. Such behavior is not achievable by a Dirichlet distribution, or a single Dirichlet tree.



(a) Dirichlet Tree choosing $\{A, C\}$.



(b) Dirichlet Tree choosing $\{B\}$.

Figure 6.6: Cannot-Link mixture components and samples.

Finally, although in the worst case the number of maximal cliques $Q^{(r)}$ in a connected component of size $|r|$ can grow exponentially as $O(3^{|r|/3})$ [Griggs et al., 1988], in the subsequent experiments $Q^{(r)}$ is no larger than 3, due in part to Must-Linked words "collapsing" to a single node in the

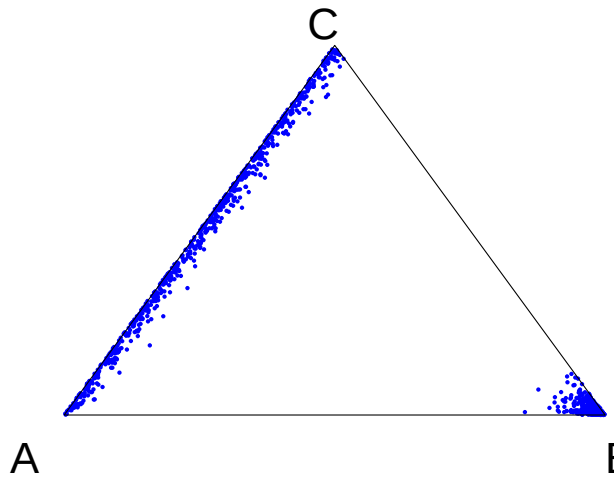


Figure 6.7: Samples from the Cannot-Link mixture of Dirichlet Trees.

Cannot-Link graph. The maximal cliques are discovered via the Bron-Kerbosch branch-and-bound algorithm [Bron and Kerbosch, 1973].

6.3 The Dirichlet Forest prior

We now bring these ideas together. First, we take the transitive closure of the Must-Link graph in order to find our Must-Link connected components. Next, we build the Cannot-Link graph where nodes represent either single words or Must-Link connected components. Note that the domain knowledge must be "consistent" in that no pairs of words are simultaneously Cannot-Linked and Must-Linked (either explicitly or implicitly through Must-Link transitive closure).

Let R be the number of connected components in the Cannot-Link-graph. Our Dirichlet Forest consists of $\prod_{r=1}^{R} Q^{(r)}$ Dirichlet trees, represented by the template in Figure 6.8. Each Dirichlet tree has R branches beneath the root, one for each connected component. The trees differ in which subtrees they include under these branches. For the r-th branch, there are $Q^{(r)}$ possible Dirichlet subtrees, corresponding to cliques $M_{r1} \ldots M_{rQ^{(r)}}$. Therefore, a tree in the forest is uniquely identified by an index vector $\mathbf{q} = (q^{(1)} \ldots q^{(R)})$, where $q^{(r)} \in \{1 \ldots Q^{(r)}\}$.

To draw a Dirichlet tree $\mathbf{q}$ from the prior DirichletForest(β, η), we select the subtrees independently because the R connected components are independent with respect to Cannot-Links:

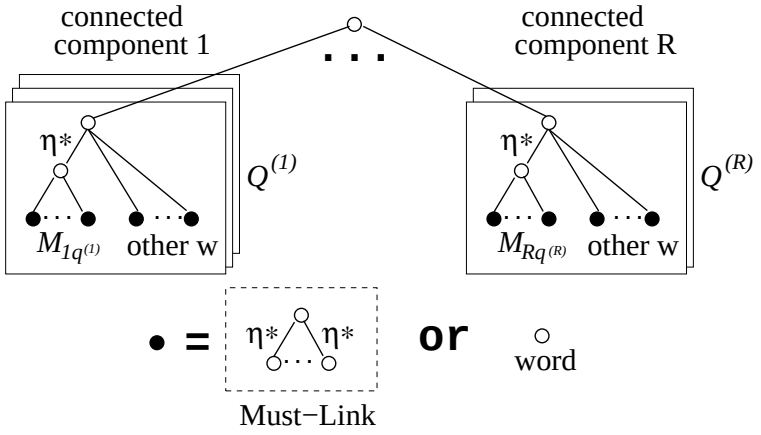


Figure 6.8: Template of Dirichlet trees in the Dirichlet Forest. For each connected component, there is a "stack" of potential subtree structures. Sampling the vector $\mathbf{q} = q^{(1)} \dots q^{(R)}$ corresponds to choosing a subtree from each stack.

$P(\mathbf{q}) = \prod_{r=1}^{R} P(q^{(r)})$. Each $q^{(r)}$ is sampled according to (6.3), and corresponds to choosing a solid box for the r-th branch in Figure 6.8. The structure of the subtree within the solid box has been defined in Section 6.2. The black nodes may be a single word, or a Must-Link transitive closure. In the latter case, the node has the subtree structure shown in the dotted box. The edge weight leading to most nodes k is $\gamma^{(k)} = |L(k)|\beta$, where $L(k)$ is the set of leaves under k. However, for edges coming out of a Must-Link internal node or going into a Cannot-Link internal node, their weights are multiplied by the strength parameter η. These edges are marked by "$\eta*$" in Figure 6.8.

The sampled Dirichlet tree $\mathbf{q}$ is then used to sample a multinomial $\phi \sim \text{Dirichlet}(\mathbf{q})$. Note, as before, when $\eta = 1$ the Dirichlet tree reduces to the Dirichlet distribution with symmetric parameter β. Thus standard LDA is a special case of Dirichlet Forest LDA.

We now define the complete Dirichlet Forest model, integrating out ("collapsing") θ and ϕ. Let $n_j^{(d)}$ be the number of word tokens in document d that are assigned to topic j. The z's are generated the same as in LDA:

$$p(\mathbf{z}|\alpha) = \left(\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\right)^D \prod_{d=1}^{D} \frac{\prod_{j=1}^{T} \Gamma(n_j^{(d)} + \alpha)}{\Gamma(n_\cdot^{(d)} + T\alpha)}. \tag{6.4}$$

There is one Dirichlet tree $\mathbf{q}_j$ per topic $j = 1 \dots T$, sampled from the Dirichlet Forest prior $P(\mathbf{q}_j) = \prod_{r=1}^{R} P(q_j^{(r)})$. Each Dirichlet tree $\mathbf{q}_j$ implicitly defines its tree edge weights $\gamma_j^{(\cdot)}$ using β, η, and its tree structure $L_j, I_j, C_j(\cdot)$. Let $n_j^{(k)}$ be the number of word tokens in the corpus assigned to topic j that appear under the node k in the Dirichlet tree $\mathbf{q}_j$. The probability of generating the corpus $\mathbf{w}$, given the trees $\mathbf{q}_{1:T} \equiv \mathbf{q}_1 \dots \mathbf{q}_T$ and the topic assignment $\mathbf{z}$, can be derived using (6.2): $P(\mathbf{w}|\mathbf{q}_{1:T}, \mathbf{z}, \beta, \eta) =$

$$\prod_{j=1}^{T} \prod_{s}^{I_j} \left(\frac{\Gamma\left(\sum_k^{C_j(s)} \gamma_j^{(k)}\right)}{\Gamma\left(\sum_k^{C_j(s)} (\gamma_j^{(k)} + n_j^{(k)})\right)} \prod_k^{C_j(s)} \frac{\Gamma(\gamma_j^{(k)} + n_j^{(k)})}{\Gamma(\gamma_j^{(k)})} \right). \tag{6.5}$$

Finally, the complete generative model is

$$p(\mathbf{w}, \mathbf{z}, \mathbf{q}_{1:T}|\alpha, \beta, \eta) = p(\mathbf{w}|\mathbf{q}_{1:T}, \mathbf{z}, \beta, \eta) p(\mathbf{z}|\alpha) \prod_{j=1}^{T} p(\mathbf{q}_j).$$

6.4 Inference and estimation

Because a Dirichlet Forest is a mixture of Dirichlet trees, which are conjugate to multinomials, we can efficiently perform inference by Markov Chain Monte Carlo (MCMC). Specifically, we use collapsed Gibbs sampling similar to Griffiths and Steyvers [2004]. However, in our case the MCMC state is defined by both the topic labels $\mathbf{z}$ and the tree indices $\mathbf{q}_{1:T}$. An MCMC iteration in our case consists of a sweep through both $\mathbf{z}$ and $\mathbf{q}_{1:T}$. We present the conditional probabilities for collapsed Gibbs sampling below, the derivation of these equations is provided in Appendix B.

6.4.1 Sampling z

Let $n_{-i,j}^{(d)}$ be the number of word tokens in document d assigned to topic j, excluding the word at position i. Similarly, let $n_{-i,j}^{(k)}$ be the number of word tokens in the corpus that are under node k in topic j's Dirichlet tree, excluding the word at position i. For candidate topic labels $v = 1 \dots T$, we have

$$P(z_i = v|\mathbf{z}_{-i}, \mathbf{q}_{1:T}, \mathbf{w}) \propto (n^{(d)}_{-i,v} + \alpha) \prod_s^{I_v(\uparrow i)} \frac{\gamma_v^{(C_v(s\downarrow i))} + n^{(C_v(s\downarrow i))}_{-i,v}}{\sum_k^{C_v(s)} \left(\gamma_v^{(k)} + n^{(k)}_{-i,v}\right)} \tag{6.6}$$

where $I_v(\uparrow i)$ denotes the subset of internal nodes in topic v's Dirichlet tree that are ancestors of leaf w_i, and $C_v(s\downarrow i)$ is the unique node that is s's immediate child *and* an ancestor of w_i (including w_i itself).

6.4.2 Sampling $\mathbf{q}$

Since the connected components are independent, sampling the tree $\mathbf{q}_j$ factors into sampling the cliques for each connected component $q_j^{(r)}$. For candidate cliques $q' = 1 \ldots Q(r)$, we have

$$p(q_j^{(r)} = q'|\mathbf{z}, \mathbf{q}_{-j}, \mathbf{q}_j^{(-r)}, \mathbf{w}) \propto \left(\sum_k^{M_{rq'}} \beta_k\right) \prod_s^{I_{j,r=q'}} \left(\frac{\Gamma\left(\sum_k^{C_j(s)} \gamma_j^{(k)}\right)}{\Gamma\left(\sum_k^{C_j(s)} (\gamma_j^{(k)} + n_j^{(k)})\right)} \prod_k^{C_j(s)} \frac{\Gamma(\gamma_j^{(k)} + n_j^{(k)})}{\Gamma(\gamma_j^{(k)})}\right) \tag{6.7}$$

where $I_{j,r=q'}$ denotes the internal nodes below the r-th branch of tree $\mathbf{q}_j$, when clique $M_{rq'}$ is selected.

6.4.3 Estimating ϕ and θ

After running MCMC for sufficient iterations, we follow standard practice (e.g. [Griffiths and Steyvers, 2004]) and use the last sample $(\mathbf{z}, \mathbf{q}_{1:T})$ to estimate ϕ and θ. Because a Dirichlet tree is a conjugate distribution, its posterior is another Dirichlet tree with the same structure, but with edge weights updated. The posterior for the Dirichlet tree of the j-th topic is ${\gamma^{post}}_j^{(k)} = \gamma_j^{(k)} + n_j^{(k)}$, where the counts $n_j^{(k)}$ are collected from $\mathbf{z}, \mathbf{q}_{1:T}, \mathbf{w}$. We estimate ϕ_j by the first moment under this posterior [Minka, 1999]

$$\widehat{\phi}_j(w) = \prod_s^{I_j(\uparrow w)} \frac{{\gamma^{post}}_j^{(C_j(s\downarrow w))}}{\sum_{s'}^{C_j(s)} {\gamma^{post}}_j^{(s')}}. \tag{6.8}$$

The parameter θ is estimated the same way as in standard LDA

$$\widehat{\theta}_d(j) = \frac{n_j^{(d)} + \alpha}{n_\cdot^{(d)} + T\alpha}. \tag{6.9}$$

6.5 Experiments

By using large values for the strength parameter η, we should be able to encourage learned topics to conform to our preferences. More interestingly, the learned topics should change in other, *indirect* ways as a result of the Dirichlet Forest Prior. For example, Cannot-Link (A,B) should encourage A and B to appear with high probabilities in separate topics, but we might also expect to see other statistically associated words be placed in topics differently as well. In Section 6.5.1 we construct small, simple synthetic datasets and perform inference multiple times in order to clearly observe the effects of the constraints. In Sections 6.5.2 and 6.5.3 we apply the Dirichlet Forest prior to real text corpora in order to understand how it can be used to influence the learned topics.

6.5.1 Synthetic data

Here we study the effect of our Dirichlet Forest Prior on *very* simple datasets. We use varying values of η in order to observe the progression from standard Dirichlet ($\eta = 1$) to "hard" constraints (large η). Previous studies often take the last MCMC sample ($\mathbf{z}$ and $\mathbf{q}_{1:T}$), and discuss the topics $\phi_{1:T}$ derived from that sample. Because of the stochastic nature of MCMC, we argue that more insight can be gained if multiple independent MCMC samples are considered. In practice, for each dataset, and each DF with a different η, we run a long MCMC chain with 200,000 iterations of burn-in, and take out a sample every 10,000 iterations afterward, for a total of 200 samples. We have some indication that our chain is well-mixed, as we observe that samples with "label switching" (i.e., equivalent up to label permutation) occur with near equal frequency, and all expected modes are observed. For each sample, we estimate its topics $\phi_{1:T}$ with Equation (6.8). We then greedily align the ϕ's from different samples, i.e., permute the T topic labels to remove the label switching effect. Within a dataset, we perform PCA on the baseline ($\eta = 1$) ϕ and project all samples into the resulting space to obtain a common visualization, dithering points to show overlap.

SynData1: Must-Link (B,C)

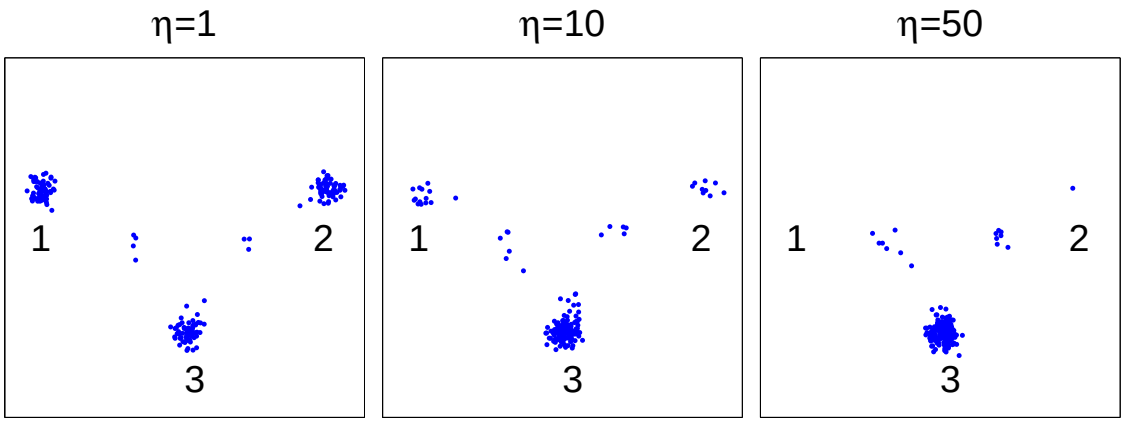
This corpus consists of six documents containing a vocabulary of five words: A,B,C,D,E. The documents themselves are shown in Figure 6.9a. We run standard LDA with $T = 2, \alpha = 0.5, \beta = 0.01$ and examine the permutation-aligned ϕ samples as described above. Figure 6.9b shows the ϕ values for idealized prototypes of the clusters shown in the PCA ϕ plots. Originally the MCMC samples are roughly evenly split among these three different sets of topics, as can be seen in Figure 6.9c. We then add the preference Must-Link (B,C) and repeat the experiment with $\eta = 10$ (Figure 6.9d) and $\eta = 50$ (Figure 6.9e). Examining the plots in Figure 6.9 as η increases, we can see that cluster 3 becomes increasingly likely. Not only are B and C present in a topic together (as the preference enforces), but A and D are "pulled" along as well by their statistical associations with B and C, even though our supplied domain knowledge did not reference A or D.

Documents
ABAB
CDCD
EEEE
ABAB
CDCD
EEEE

(a) Corpus

Cluster		$P(A\|z)$	$P(B\|z)$	$P(C\|z)$	$P(D\|z)$	$P(E\|z)$
1	ϕ_1	0.5	0.5	0	0	0
	ϕ_2	0	0	0.25	0.25	0.5
2	ϕ_1	0.25	0.25	0	0	0.5
	ϕ_2	0	0	0.5	0.5	0
3	ϕ_1	0.25	0.25	0.25	0.25	0
	ϕ_2	0	0	0	0	1

(b) Topics



(c) (d) (e)

Figure 6.9: Corpus and topic clusters for SynData1. Panels 6.9c, 6.9d, and 6.9e show the results of multiple inference runs as constraint strength η increases. For large η, the resulting topics ϕ concentrate around cluster 3, which is in agreement with our domain knowledge.

SynData2: Cannot-Link (A,B)

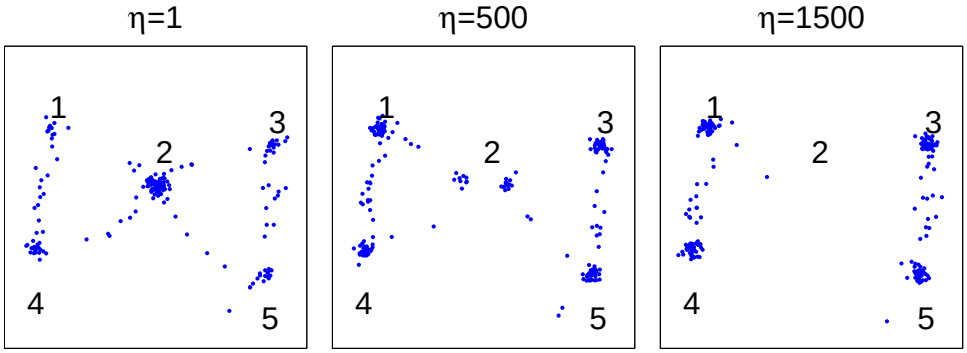
This corpus consists of six documents containing a vocabulary of four words, shown in Figure 6.10a. We run LDA with $T = 3, \alpha = 1, \beta = 0.01$. Figure 6.10b shows the values for three of the resulting six ϕ clusters shown Figure 6.10c. Adding Cannot-Link (A,B), we see cluster 2 disappear as η increases (Figure 6.10d and Figure 6.10e). This makes sense because A and B co-occur in ϕ_1 of cluster 2, violating the preference. Like cluster 1 and 6, the clusters not shown in Figure 6.10b (clusters 3,4, and 5) also obey our Cannot-Link preference and are preserved as η increases.

Documents
ABCCABCC
ABDDABDD
ABCCABCC
ABDDABDD
...

(a) Corpus

Cluster		$P(A\|z)$	$P(B\|z)$	$P(C\|z)$	$P(D\|z)$
1	ϕ_1	0	0.5	0	0.5
	ϕ_2	1	0	0	0
	ϕ_3	0	0	1	0
2	ϕ_1	0.5	0.5	0	0
	ϕ_2	0	0	1	0
	ϕ_3	0	0	0	1
...	...	...	...	...	...
6	ϕ_1	0	0	0.5	0.5
	ϕ_2	1	0	0	0
	ϕ_3	0	1	0	0

(b) Topics



(c) (d) (e)

Figure 6.10: Corpus and topic clusters for SynData2. Panels 6.10c, 6.10d, and 6.10e show the results of multiple inference runs as constraint strength η increases. For large η, the resulting topics ϕ avoid cluster 2, which conflicts with our domain knowledge.

SynData3: Isolate (B)

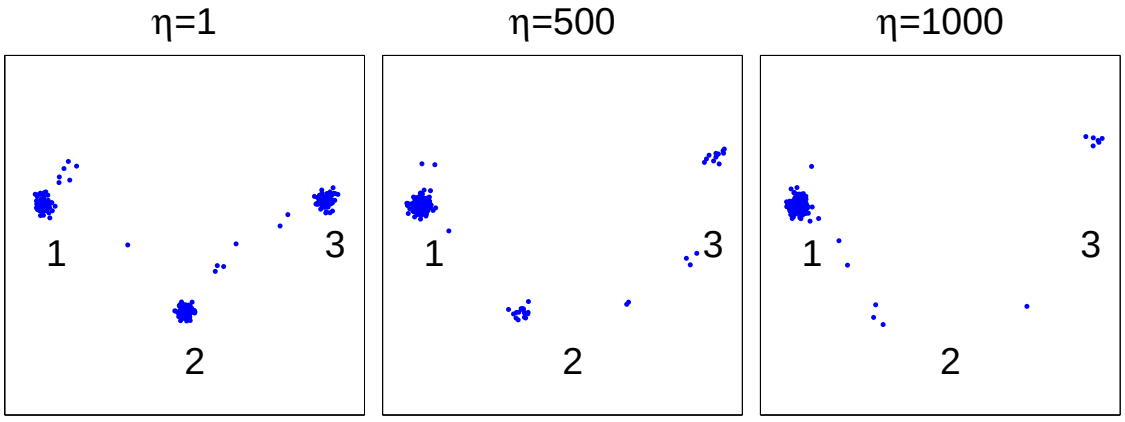
This corpus consists of four documents containing a vocabulary of three words, shown in Figure 6.11a. We run LDA with $T = 2, \alpha = 1, \beta = 0.01$. Figure 6.11b shows the values of the resulting three ϕ clusters shown Figure 6.11c. We wish to **Isolate** B into its own topic, so we add Cannot-Link (A,B) and Cannot-Link (B,C). As η increases, Figure 6.11d and Figure 6.11e show that samples do concentrate in cluster 1, where B is isolated in its own topic.

Documents
ABC
ABC
ABC

(a) Corpus

Cluster		$P(A\|z)$	$P(B\|z)$	$P(C\|z)$
1	ϕ_1	0.5	0	0.5
	ϕ_2	0	1	0
2	ϕ_1	0.5	0.5	0
	ϕ_2	0	0	1
3	ϕ_1	0	0.5	0.5
	ϕ_2	1	0	0

(b) Topics



(c) (d) (e)

Figure 6.11: Corpus and topic clusters for SynData3. Panels 6.11c, 6.11d, and 6.11e show the results of multiple inference runs as constraint strength η increases. For large η, the resulting topics ϕ concentrate around cluster 1, which is in agreement with our domain knowledge.

SynData4: Split (AB,CD)

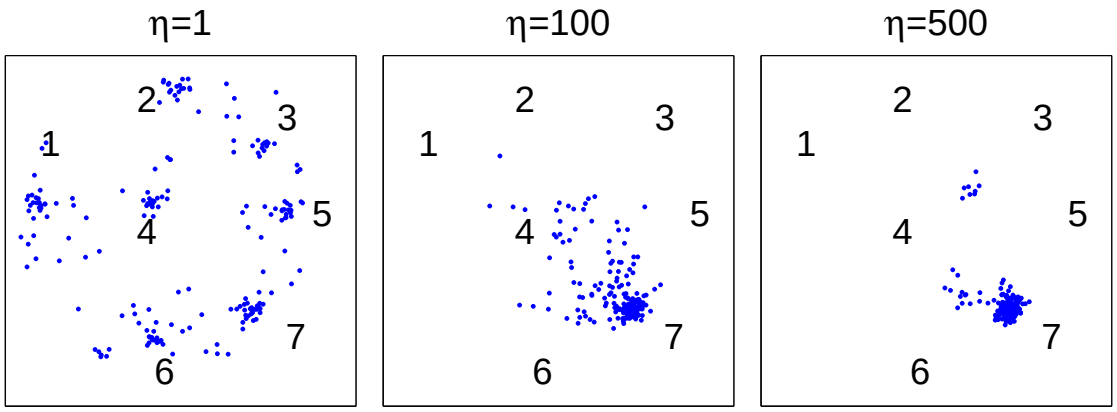
This corpus consists of six documents containing a vocabulary of six words, shown in Figure 6.12a. We first run LDA with $T = 3, \alpha = 0.5, \beta = 0.01$. The plot is not shown, but nearly all samples result in the three topics shown in Figure 6.12b. Now say we wish to break out (A,B) into one topic, and (B,C) into another. First, we increase T to four topics, yielding the clusters shown in Figure 6.12c. In order to **Split** AB from CD, we apply Must-Link (A,B), Must-Link (C,D), and finally Cannot-Link (B,C). For simplicity, Figure 6.12b shows only cluster 7 from Figure 6.12c. We note that the other various topic combination clusters violate our **Split** preferences. As η increases in Figure 6.12d and Figure 6.12e, we can see the samples gravitate towards cluster 7, the only cluster satisfying the constraints.

Documents
ABCDEEEE
ABCDFFFF
ABCDEEEE
ABCDFFFF
ABCDEEEE
ABCDFFFF

(a) Corpus

Cluster		$P(A\|z)$	$P(B\|z)$	$P(C\|z)$	$P(D\|z)$	$P(E\|z)$	$P(F\|z)$
$T = 3$	ϕ_1	0.25	0.25	0.25	0.25	0	0
(not	ϕ_2	0	0	0	0	1	0
shown)	ϕ_3	0	0	0	0	0	1
7	ϕ_1	0.5	0.5	0	0	0	0
	ϕ_2	0	0	0.5	0.5	0	0
	ϕ_3	0	0	0	0	1	0
	ϕ_4	0	0	0	0	0	1

(b) Topics



(c) (d) (e)

Figure 6.12: Corpus and topic clusters for SynData4. Panels 6.12c, 6.12d, and 6.12e show the results of multiple inference runs as constraint strength η increases. For large η, the resulting topics ϕ concentrate around cluster 7, which is in agreement with our domain knowledge.

6.5.2 Wish corpus

We now consider *interactive topic modeling* with the Dirichlet Forest Prior. The corpus consists of a collection of 89,574 New Year's wishes submitted to The Times Square Alliance [Goldberg et al., 2009]. Each wish is treated as a document, downcased but without stopword removal. For each step in this interactive example, we set $\alpha = 0.5$, $\beta = 0.1$, $\eta = 1000$, and run MCMC for 2000 iterations before estimating the topics from the final sample. The domain knowledge encoded into the Dirichlet Forest prior accumulates along the steps (i.e., a Cannot-Link applied in Step 2 remains in effect for Steps 3 and 4).

Step 1: We begin by running standard LDA with $T = 15$. In the learned topics (Table 6.1) we can see that many of the most probable words in the topics are conventional ("to", "and") or corpus-specific ("wish", "2008") stopwords (i.e., commonly occurring and/or uninformative terms), which obscure the meanings of the topics.

Step 2: We then manually create a 50-word stopword list, and issue an **Isolate** preference. This is compiled into Must-Links among this set and Cannot-Links between this set and all other words in the top 50 for all topics. To absorb the newly **Isolated** terms, T is also increased from 15 to 16. We then run inference for LDA with a Dirichlet Forest prior encoding these preferences. Table 6.2 shows the set of learned topics, which now contains two distinct stopword topics. Importantly, with the stopwords explained by these two topics, the top words for the other topics become much more meaningful.

Step 3: One topic (marked as **MIXED** in Table 6.2) conflates two distinct concepts: *enter college* and *cure disease* (see the top eight words: {go, school, cancer, into, well, free, cure, college}). To correct this mixed topic, we issue a **Split** ({go, school, into, college}, {cancer, free, cure, well}) operation in order to separate the concepts. This is compiled into Must-Links within each quadruple, and a Cannot-Link between them. T is then further increased from 16 to 18 in order to accommodate the two concepts. After running DF-LDA and examining the topics (Table 6.3), we can see one of the topics clearly takes on the *college* concept. Significantly, this new topic also picks up related words which we did *not* explicitly encode into our prior {job, good, graduate, accepted, ...}. Another topic does likewise for the *cure* concept (many wishes are like

Table 6.1: High-probability words for standard LDA topics, with uninformative terms highlighted in red.

Topic	Top words sorted by $\phi = p(\text{word}\|\text{topic})$
0	love i you me and will forever that with hope
1	and health for happiness family good my friends
2	year new happy a this have and everyone years
3	that is it you we be t are as not s will can
4	my to get job a for school husband s that into
5	to more of be and no money stop live people
6	to our the home for of from end safe all come
7	to my be i find want with love life meet man
8	a and healthy my for happy to be have baby
9	a 2008 in for better be to great job president
10	i wish that would for could will my lose can
11	peace and for love all on world earth happiness
12	may god in all your the you s of bless 2008
13	the in to of world best win 2008 go lottery
14	me a com this please at you call 4 if 2 www

Table 6.2: High-probability words for each topic after applying **Isolate** to stopwords. The topics marked `Isolate` have absorbed most of the stopwords, while the topic marked **MIXED** seems to contain words from two distinct concepts.

Topic	Top words sorted by $\phi = p(\text{word}\|\text{topic})$
0	love forever marry happy together mom back
1	health happiness good family friends prosperity
2	life best live happy long great time ever wonderful
3	out not up do as so what work don was like
MIXED	go school cancer into well free cure college
5	no people stop less day every each take children
6	home safe end troops iraq bring war husband house
7	love peace true happiness hope joy everyone dreams
8	happy healthy family baby safe prosperous everyone
9	better job hope president paul great ron than person
10	make money lose weight meet finally by lots hope
`Isolate`	and to for a the year in new all my 2008
12	god bless jesus loved know everyone love who loves
13	peace world earth win lottery around save
14	com call if 4 2 www u visit 1 3 email yahoo
`Isolate`	i to wish my for and a be that the in

Table 6.3: High-probability words for each topic after applying **Split** to *school/cancer* topic. The topics marked Split contain the concepts which were previously mixed. The two topics marked **LOVE** both seem to cover the same concept.

Topic	Top words sorted by $\phi = p(\text{word}\|\text{topic})$
LOVE	love forever happy together marry fall
1	health happiness family good friends
2	life happy best live love long time
3	as not do so what like much don was
4	out make money house up work grow able
5	people no stop less day every each take
6	home safe end troops iraq bring war husband
7	love peace happiness true everyone joy
8	happy healthy family baby safe prosperous
9	better president hope paul ron than person
LOVE	lose meet man hope boyfriend weight finally
Isolate	and to for a the year in new all my 2008
12	god bless jesus loved everyone know loves
13	peace world earth win lottery around save
14	com call if 4 www 2 u visit 1 email yahoo 3
Isolate	i to wish my for and a be that the in me get
Split	job go school great into good college
Split	mom husband cancer hope free son well

Table 6.4: High-probability words for final topics after applying **Merge** to *love* topics. The two previously separate topics have been combined into the topic marked Merge

Topic	Top words sorted by $\phi = p(\text{word}\|\text{topic})$
Merge	love lose weight together forever marry meet
success	health happiness family good friends prosperity
life	life happy best live time long wishes ever years
-	as do not what someone so like don much he
money	out make money up house work able pay own lots
people	no people stop less day every each other another
iraq	home safe end troops iraq bring war return
joy	love true peace happiness dreams joy everyone
family	happy healthy family baby safe prosperous
vote	better hope president paul ron than person bush
Isolate	and to for a the year in new all my
god	god bless jesus everyone loved know heart christ
peace	peace world earth win lottery around save
spam	com call if u 4 www 2 3 visit 1
Isolate	i to wish my for and a be that the
Split	job go great school into good college hope move
Split	mom hope cancer free husband son well dad cure

{mom, stays, cancer, free}), again picking up related, but un-encoded, terms ({hope, surgery, cure, pain. . .}). Other topics have only minor changes.

Step 4: Two topics seem to both correspond to romance concepts (marked as **LOVE** in Table 6.3). We apply **Merge** ({love, forever, marry, together, loves}, {meet, boyfriend, married, girlfriend, wedding}), which is compiled into Must-Links between these words. We decrease T from 18 to 17 in order to accommodate the merging of two topics into one. After running DF-LDA and examining the topics (Table 6.4), we can see that one of the romance topics disappears, and the remaining one corresponds to the merged romance topic. An interesting artifact of this merged topic is the continued presence of the terms "lose" and "weight", which were not part of our **Merge** preference but which have strong statistical associations with our **Merge** terms in the corpus. This is due to the prevalent wishes of the form "lose weight and meet a boyfriend", etc. Again, other previous topics survive with only minor changes.

The results of this interactive topic modeling process show how a user can examine the topics, and iteratively use the Dirichlet Forest prior to refine the topics such that they align well with user-interpretable concepts. Examining the final learned topics (Table 6.4) and the original ones (Table 6.1), we can see that the final ones give us better insights into the common themes of people's New Year's wishes.

6.5.3 Yeast corpus

Whereas the previous experiment illustrates the utility of our approach in an interactive setting, we now consider a case in which we use background knowledge from an ontology to guide topic modeling. Here our prior knowledge is based on six concepts. The concepts *transcription*, *translation* and *replication* characterize three important *processes* that are carried out at the molecular level. The concepts *initiation*, *elongation* and *termination* describe *phases* of the three aforementioned processes. Combinations of concepts from these two sets correspond to concepts in the Gene Ontology (e.g., GO:0006414 is *translational elongation*, and GO:0006352 is *transcription initiation*).

We guide our topic modeling using Must-Links among a small set of words for each concept. Moreover, we use Cannot-Links among words to specify that we prefer (i) *transcription*, *translation* and *replication* to be represented in separate topics, and (ii) *initiation*, *elongation* and *termination* to be represented in separate topics. In higher-level terms, these preferences can be thought of as two three-way **Split** operations: **Split** (*transcription*, *translation*, *replication*) and **Split** (*initiation*, *elongation*, *termination*). Note that we do not set any preferences between the *process* topics and the *phase* topics, however.

The corpus that we use for our experiments consists of 18,193 abstracts selected from the MEDLINE database for their relevance to yeast genes. We induce topic models using DF-LDA to encode the Must-Links and Cannot-Links described above, and use standard LDA as a control. We set $T = 100, \alpha = 0.5, \beta = 0.1, \eta = 5000$. For each word that we use to seed a concept, Table 6.5 shows the topics that include it among their 50 most probable words. The middle part of the table shows the topic-word relationships for ordinary LDA and the right part of the table shows the relationships for DF-LDA.

We make several observations about the DF-LDA topics versus the standard LDA topics. First, each concept is represented by a small number of topics and the Must-Link words for each topic all occur as highly probable words in these topics. Second, the Cannot-Link preferences are obeyed in the final topics. Third, the topics use the process and phase topics compositionally. For example, DF-LDA Topic 4 represents *transcription initiation* and DF Topic 8 represents *replication initiation*. Moreover, the topics that are significantly influenced by the prior typically include highly relevant terms among their most probable words. For example, the top words in DF Topic 4 include "TATA", "TFIID", "promoter", and "recruitment" which are all specifically germane to the composite concept of *transcription initiation*. In the case of standard LDA, the seed concept words are dispersed across a greater number of topics, and highly related words, such as "cycle" and "division" often do not fall into the same topic. Many of the topics induced by ordinary LDA are semantically coherent, but the specific concepts suggested by our prior do not naturally emerge without using DF-LDA.

Table 6.5: Yeast corpus topics. The left column shows the seed words in the DF-LDA model. The middle columns indicate the topics in which at least two seed words are among the 50 highest probability words for LDA, the "o" column gives the number of other topics (not shared by another word). Finally, the right columns show the same topic-word relationships for the DF model.

	LDA									DF									
	1	2	3	4	5	6	7	8	o	1	2	3	4	5	6	7	8	9	10
transcription	•			•	•				1	•		•						•	
transcriptional	•			•	•				2	•		•						•	
template					•				1	•		•						•	
translation							•	•			•				•				
translational								•			•				•				
tRNA									1		•				•				
replication		•							2				•			•	•	•	
cycle		•	•										•			•	•	•	
division			•						3				•			•	•	•	
initiation		•		•	•	•		•			•	•		•	•				
start			•	•		•					•	•		•	•				
assembly						•		•	7		•	•		•	•				
elongation					•			•	1									•	
termination							•	•			•								
disassembly											•								
release									2		•								
stop								•			•								

This experiment shows how using a controlled knowledge base (like the Gene Ontology) to encode concepts into a Dirichlet Forest prior can result in interesting topics which are well-aligned with the knowledge base concepts. Importantly, the learned topics both obey the encoded preferences *and* extend these concepts to other related terms by exploiting statistical patterns in the text.

6.6 Summary

This chapter has demonstrated a novel mechanism for encoding domain knowledge and prior beliefs into topic modeling. Unlike any existing approaches, the Dirichlet Forest prior allows the expression of constraints *among sets* of words. Our experiments show the usefulness of this approach in settings where the domain knowledge comes from both user interaction as well as pre-existing knowledge base. The Dirichlet Forest also retains beneficial conjugacy properties with respect to the multinomial distribution, enabling the use of Collapsed Gibbs sampling for inference.

KEY IDEAS

- ◇ The Dirichlet Forest prior on ϕ allows *pairwise* constraints on words.
  - ▷ Must-Link (A,B) $\rightarrow$ A and B should have *similar* probability in each topic.
  - ▷ Cannot-Link (A,B) $\rightarrow$ A and B should not both have high probability in the *same* topic.
- ◇ Specific pairwise constraints can come from a variety of sources:
  - ▷ user feedback,
  - ▷ existing knowledge bases.
- ◇ The Dirichlet Forest prior allows inference via Collapsed Gibbs sampling.

Chapter 7

LogicLDA

This chapter presents LogicLDA, a model which allows the user to express domain knowledge in First-Order Logic (FOL). This formulation generalizes many existing LDA variants, and enables the inclusion of novel forms of domain knowledge. The generality of FOL allows the extension of LDA along all three dimensions discussed in Chapter 3: LDA+X, ϕ-side, and θ-side.

This extension builds on recent research in probabilistic logic modeling such as Markov Logic Networks (MLN) [Richardson and Domingos, 2006]. LogicLDA can be viewed as an instance of a Hybrid Markov Logic Network (HMLN) [Wang and Domingos, 2008], which is a generalization of a MLN. However, existing work in this direction has not integrated topic modeling with FOL. Furthermore, the topic modeling aspects of LogicLDA preclude the use of specialized inference techniques developed for MLNs. This is a significant problem because inference can become intractable for certain types of domain knowledge. In order to make LogicLDA practical, we present a scalable stochastic inference scheme based on mirror descent [Beck and Teboulle, 2003]. This approach may also be useful for MAP inference in MLNs, providing a more scalable alternative to existing techniques such as MaxWalkSAT (MWS) [Selman et al., 1995] and Integer Linear Programming (ILP) [Riedel, 2008].

7.1 The LogicLDA model

We now describe the LogicLDA model. As mentioned earlier in Chapter 5, we begin by representing the standard LDA model in terms of an undirected factor graph which is more natural for

LogicLDA. We then briefly review some important logic terms and concepts. Finally, we present the full LogicLDA model, unifying topic modeling and logical modeling.

7.1.1 Undirected LDA

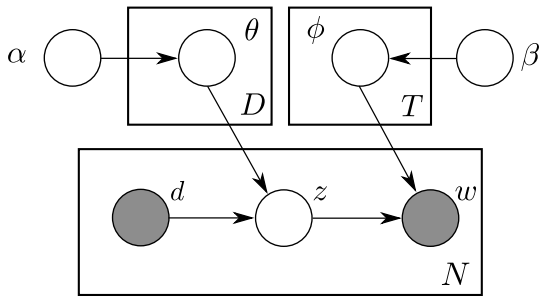
Consider the Latent Dirichlet Allocation (LDA) directed graphical model in Figure 7.1a. Note that the presentation is slightly different than the model shown in Chapter 2 in that the document d is represented explicitly, instead of via a document "plate". We can mechanically convert this to an undirected graphical model or Markov Random Field (MRF) by adding edges between co-parents [Koller and Friedman, 2009], resulting in the model shown in Figure 7.1b. In this representation, the *conditional probability* tables and distributions previously defined for each node and its parents are replaced with *potential functions* for each clique of nodes. This can be represented more explicitly by converting the undirected model to the factor graph shown in Figure 7.1c. In this representation, each maximal clique is now associated with a special factor node (the black squares), and the former clique members are now connected to that factor. Each factor is then associated with a potential function of its neighboring variable nodes. The probability of a specific configuration of all variables $(\theta, \phi, \mathbf{d}, \mathbf{z}, \mathbf{w})$ is now given by multiplying the values of the potential functions and normalizing. We simply take each potential function to be equal to the original conditional probability it is replacing. The Dirichlet priors on θ and ϕ are represented by $f_\alpha(\alpha, \theta^{(d)})$ and $f_\beta(\beta, \phi_z)$, respectively. The document-topic multinomial terms are expressed by $f_\theta(\theta^{(d)}, z_i)$, while the topic-word multinomial contribution is represented by $f_\phi(\phi, z_i, w_i)$.

$$f_\alpha(\alpha, \theta^{(d)}) = \prod_z^T (\theta_z^{(d)})^{(\alpha_z - 1)} \tag{7.1}$$

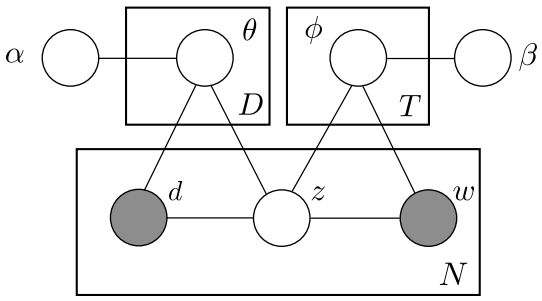
$$f_\beta(\beta, \phi_z) = \prod_w^V (\phi_z^{(w)})^{(\beta_w - 1)} \tag{7.2}$$

$$f_\theta(\theta^{(d)}, z_i) = \theta_{z_i}^{(d)} \tag{7.3}$$

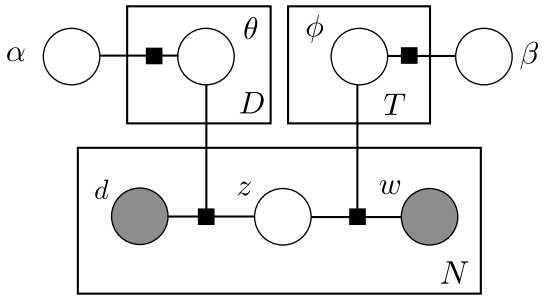
$$f_\phi(\phi, z_i, w_i) = \phi_{z_i}^{(w_i)} \tag{7.4}$$



(a) Standard LDA graphical model.



(b) Undirected LDA graphical model.



(c) LDA factor graph.

Figure 7.1: Conversion of LDA to a factor graph representation. In each diagram, filled circles represent observed variables, empty circles are associated with latent variables or model hyperparameters, and plates indicate repeating structure. The black squares in Figure 7.1c are the *factor nodes*, and are associated with the potential functions given in Equations 7.1, 7.2, 7.3, and 7.4.

Taking the products of all potential functions, it is straightforward to see that the distribution encoded by this factor graph is exactly equal to the distribution encoded by the original LDA directed graphical model. This point should be emphasized: the model itself is *exactly the same*, we simply have a new representation with which to work. This new representation will prove convenient for development of LogicLDA.

7.1.2 Logic background

LogicLDA allows the user to express domain knowledge in the language of First-Order Logic (FOL). We begin by briefly defining some basic logic terms and concepts [Russell and Norvig, 2003, Richardson and Domingos, 2006].

- **Constants** are symbols that represent an actual object in the problem domain, such as the word "saxophone" or the value 3.
- **Variables** are symbols (such as x) that can take on values from the set of constants.
- **Predicates** are symbols that express relations, and evaluate to *true* or *false* for different arguments. For example, $\texttt{isNoun}(\text{saxophone})$ would be *true*, while $\texttt{greaterThan}(3, 5)$ would be *false*.
- **Functions** are symbols that express mappings. For example, the function $\texttt{wordAt}(i)$ could return the word present at position i in our corpus. Functions can be composed with predicates in useful ways. If $w_3 =$ "saxophone", then $\texttt{isNoun}(\texttt{wordAt}(3))$ becomes $\texttt{isNoun}(\text{saxophone}))$, which evaluates to *true*.
- **Terms** are any expressions that refer to objects in our domain. This includes constants, variables, or functions applied to sets of terms. For example, $\texttt{wordAt}(3)$ or x or "saxophone".
- **Atoms** are predicates applied to terms, such as $\texttt{isNoun}(\texttt{wordAt}(3))$. Atoms are also known as **Literals**. They can also be referred to as negative or positive literals if they are negated or not, respectively.

- **Formulas** are constructed from atoms using logical connectives $(\wedge, \vee, \neg, \Rightarrow)$. Variables appearing in a formula must be *quantified*, either universally ($\forall$) or existentially ($\exists$). For example, the formula $\forall x\ \texttt{greaterThan}(x, 5) \Rightarrow \texttt{isNoun}(\texttt{wordAt}(x))$ asserts that all words after the fifth word in our corpus are nouns.

- **Clauses** are formulas consisting of a disjunction of literals. A conjunction of clauses is said to be in **Conjunctive Normal Form** or CNF.

- **Ground** terms, atoms, or formulas contain no variables. For example, $\texttt{isNoun}(x)$ is not grounded, but $\texttt{isNoun}(\text{saxophone})$ is.

Markov Logic Networks (MLNs) [Richardson and Domingos, 2006] are a class of graphical models which operate over this type of logical domain. A "possible world" consists of a set of binary truth assignments for all possible ground predicates. A particular MLN instance then assigns probabilities to all possible worlds.

In order to incorporate logic into LDA, we must represent key LDA variables within this framework. We define the following logical variables and predicates:

- $\texttt{Z}(i, t)$ is *true* if the hidden topic $z_i = t$, and *false* otherwise.

- $\texttt{W}(i, v)$ is *true* if word $w_i = v$, and *false* otherwise.

- $\texttt{D}(i, j)$ is *true* if $d_i = j$, and *false* otherwise.

For the predicates defined above, each of the variables ($i, t, v,$ and j) ranges over a set of integers. For example, the corpus index i takes on values in $[1, \ldots, N]$. Similarly, the latent topic variable $t \in [1, \ldots, T]$, the vocabulary word variable $v \in [1, \ldots, W]$, and the document index $j \in [1, \ldots, D]$.

Importantly, note that for a given index i, the predicate $\texttt{W}(i, v)$ must be *true* for *exactly* one value of v and *false* for all others. Likewise, for an index i, $\texttt{Z}(i, t)$ and $\texttt{D}(i, j)$ are each *true* for exactly one topic t and document j, respectively. These types of special predicate arguments are called "mutually exclusive and exhaustive" [Kok et al., 2009].

Crucially, LogicLDA can incorporate other observed variables. For example, we may wish to analyze a Congressional debate corpus where each speech is a document. In this case we can define a predicate $\texttt{Speaker}(d_i, \text{Rep})$, which is *true* if the speaker for document d_i is a member of the Republican political party. We will use $\mathbf{o}$ to collectively denote these other observed variables.

7.1.3 LogicLDA model

The key to LogicLDA is to allow domain knowledge specified in FOL to influence the values of the hidden topics $\mathbf{z}$, which, in turn, influence the recovered topic-word multinomials ϕ and document-topic multinomials θ. Returning to our Congressional debate corpus, we may specify the rule

$$\forall i : \texttt{W}(i, \text{taxes}) \wedge \texttt{Speaker}(d_i, \text{Rep}) \Rightarrow \texttt{Z}(i, 77), \tag{7.5}$$

which states that for any word $w_i =$"taxes" that appears in a speech by a Republican, it should be in topic $z_i = 77$ (note that this need not be a hard constraint). This rule will have the effect of encouraging the directly affected words to be assigned to Topic 77, but this change may also influence the topic recovery in other indirect ways. For example, words statistically associated with "taxes" in Republican speeches (e.g., {cuts, growth, stimulate}) may also come to be associated with Topic 77

We now describe the LogicLDA modeling process. First, the domain expert specifies the background knowledge by defining a *weighted FOL knowledge base* KB, which is then converted into Conjunctive Normal Form: $KB = \{(\lambda_1, \psi_1), \ldots, (\lambda_L, \psi_L)\}$. The KB consists of L pairs, where each ψ_l represents a FOL rule, and $\lambda_l \geq 0$ is its weight which the domain expert can set to adjust the importance of individual rules.

The knowledge base KB is tied to our probabilistic model via its *groundings*. For each FOL rule ψ_l, let $G(\psi_l)$ be the set of groundings, each mapping the free variables in ψ_l to a specific value. For the "taxes" example above, G consists of all N propositional rules where $i \in [1, \ldots, N]$. For each grounding $g \in G(\psi_l)$, we define an indicator function

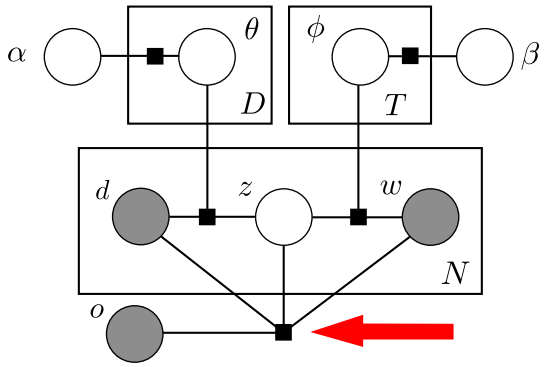


Figure 7.2: LogicLDA factor graph with "mega" logic factor (indicated by arrow) connected to $\mathbf{d}$, $\mathbf{z}$, $\mathbf{w}$, $\mathbf{o}$.

$$\mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o}) = \begin{cases} 1, & \text{if } g \text{ is } \textit{true} \text{ under } \mathbf{z} \text{ and observed } \mathbf{w}, \mathbf{d}, \mathbf{o} \\ 0, & \text{otherwise} \end{cases} \tag{7.6}$$

For example, if w_{100}="taxes", $\texttt{Speaker}(d_{100}, \text{Rep}) = \textit{true}$, and $z_{100} = 88$, then the grounding $g = (\texttt{W}(100, \text{taxes}) \wedge \texttt{Speaker}(d_{100}, \text{Rep}) \Rightarrow \texttt{Z}(100, 77))$ will have $\mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o}) = 0$ because of the mismatch in z_{100}.

To combine logic and LDA, we define a Markov Random Field over latent topic assignments $\mathbf{z}$, topic-word multinomials ϕ, and document-topic multinomials θ, treating words $\mathbf{w}$, documents $\mathbf{d}$, and side information $\mathbf{o}$ as observed. Specifically, in this Markov Random Field the conditional probability $P(\mathbf{z}, \phi, \theta \mid \alpha, \beta, \mathbf{w}, \mathbf{d}, \mathbf{o}, KB)$ is proportional to

$$\exp\left[\sum_{l}^{L} \sum_{g \in G(\psi_l)} \lambda_l \mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o})\right] \left(\prod_{t}^{T} p(\phi_t|\beta)\right) \left(\prod_{j}^{D} p(\theta_j|\alpha)\right) \left(\prod_{i}^{N} \phi_{z_i}(w_i)\theta_{d_i}(z_i)\right). \tag{7.7}$$

This Markov Random Field has two parts, one from logic (the first term in (7.7)), and one from LDA (the other terms in (7.7) which are identical to (2.1)). Every satisfied grounding of FOL rule ψ_l contributes $\exp(\lambda_l)$ to the potential function. Note that in general, the FOL rules *couple* all the components of $\mathbf{z}$, although the actual dependencies will be determined by the particular forms of the FOL rules. We can represent the Markov Random Field for LogicLDA as a factor graph with

an additional "mega factor node" added to the factor graph representation of standard LDA. This new factor graph is shown in Figure 7.2.

The first term in (7.7) is equivalent to a Markov Logic Network [Richardson and Domingos, 2006], while the remaining terms in (7.7) come from the LDA model. Similar to Syntactic Topic Models [Boyd-Graber and Blei, 2008], LogicLDA can therefore be interpreted as a Product of Experts model [Hinton, 2002] where the model probability is the product of the individual MLN and LDA contributions.

Another perspective is that LogicLDA consists of an MLN augmented with continuous variables (θ, ϕ) and associated potential functions. This combination has been proposed in the MLN community under the name of Hybrid Markov Logic Networks [Wang and Domingos, 2008]. However, to our knowledge previous HMLN research has not combined logic with LDA. Furthermore, the general inference technique proposed for HMLNs would be impractically inefficient for LogicLDA.

7.2 Inference

We now turn to the question of inference in LogicLDA. Ultimately, we are interested in learning the most likely ϕ and θ in a LogicLDA model. However, as in standard LDA, the latent topic assignments $\mathbf{z}$ cannot be marginalized out in practice due to their combinatorial nature. We instead aim to find the *maximum a posteriori* estimate of $\mathbf{z}, \phi, \theta$ jointly. This can be formulated as maximizing the logarithm of the unnormalized probability (7.7):

$$\underset{\mathbf{z},\phi,\theta}{\operatorname{argmax}} \sum_{l}^{L} \sum_{g \in G(\psi_l)} \lambda_l \mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o}) + \sum_{t}^{T} \log p(\phi_t|\beta) + \sum_{j}^{D} \log p(\theta_j|\alpha) + \sum_{i}^{N} \log \phi_{z_i}(w_i)\theta_{d_i}(z_i) \tag{7.8}$$

We will see that the inclusion of logic will present unique challenges due to the addition of the ground formula potential functions. These challenges motivate the development of our scalable inference procedure, called Alternating Optimization with Mirror Descent (Mir). However we

begin by presenting several baseline approaches which arise as natural extensions of existing LDA and MLN inference techniques.

7.2.1 Collapsed Gibbs Sampling (CGS)

Earlier in this thesis we have relied on Collapsed Gibbs Sampling to do inference in our topic models. Given the discrete nature of MLNs, Gibbs sampling is an established inference approach for these models as well. Therefore it is quite natural to consider the possibility of doing Gibbs sampling for the joint LogicLDA model.

Let $n_{jt}^{(-i)}$ be the number of word tokens in document j assigned to topic t omitting the word token at position i. Likewise let $n_{tw}^{(-i)}$ be the number of occurrences of word w assigned to topic t throughout the entire corpus, again omitting the word token at position i. The collapsed Gibbs sampler then iteratively re-samples z_i at each corpus position i, with the probability of candidate topic assignment $z_i = t$ given by $P(z_i = t|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{d}, \mathbf{o}, KB, \alpha, \beta) \propto$

$$\left(\frac{n_{d_i t}^{(-i)} + \alpha_t}{\sum_{t'}^{T}(n_{d_i t'}^{(-i)} + \alpha_{t'})}\right)\left(\frac{n_{t w_i}^{(-i)} + \beta_{w_i}}{\sum_{w'}^{W}(n_{t w'}^{(-i)} + \beta_{w'})}\right)\exp\left(\sum_{l}\sum_{\substack{g \in G(\psi_l) \\ g_i \neq \emptyset}} \lambda_l \mathbb{1}_g(\mathbf{z}_{-i} \cup \{z_i = t\})\right), \quad (7.9)$$

where the first two terms are the simple count ratios from LDA collapsed Gibbs sampling [Griffiths and Steyvers, 2004], and the the final term is the MLN Gibbs sampling equation [Richardson and Domingos, 2006]. For a derivation of this equation see Appendix C.

While Gibbs sampling is not aimed at maximizing the objective (7.8), the hope is that the Markov chain will explore some high probability regions of the $\mathbf{z}$ space. We initialize this sampler using the final sample from standard LDA, and keep the sample which maximizes (7.8). A drawback of this approach is the potential for poor mixing in the presence of highly weighted logic rules [Poon and Domingos, 2006].

7.2.2 MaxWalkSAT (MWS)

A naïve way to introduce logic into LDA is the following: perform standard LDA inference (e.g., with collapsed Gibbs sampling), and then post-process the latent topic vector $\mathbf{z}$ in order to maximize the weight of satisfied ground logic clauses. This post-processing corresponds to optimizing the MLN objective only (the first term in (7.8)). This can be done using a weighted satisfiability solver such as MaxWalkSAT (MWS) [Selman et al., 1995], which has previously been used to do MAP inference in MLNs [Domingos and Lowd, 2009].

MWS is a simple but effective stochastic local search algorithm sketched in Algorithm 1. It first selects a currently unsatisfied ground rule, and then attempts to satisfy it by flipping the truth state of a single atom in the clause[1]. With probability p, the atom to flip within the grounding is chosen *randomly*, otherwise the atom is chosen *greedily* to maximize the global impact Δ_{KB} on the overall logic objective. In our case, a local step involves flipping a single $\mathtt{Z}(i,t)$, which is equivalent to changing the value of a single z_i. The impact of a local move setting $z_i = t$ can be calculated as $\Delta_{KB}(i,t) = \sum_l \sum_{g \in G(\psi_l)} \lambda_l \mathbb{1}_g(\mathbf{z}_{-i} \cup \{z_i = t\})$, where $\mathbf{z}_{-i}$ is $\mathbf{z}$ excluding position i. The process is repeated for a prescribed number of iterations, and we keep the best (highest satisfied weight) assignment $\mathbf{z}$ found by MWS. While the simplicity of this method is appealing, it does not allow for any *interaction* between the logical rules and the learned topics. Consequently, the logic post-processing step may actually *decrease* the joint LogicLDA objective (7.8) by selecting a $\mathbf{z}$ configuration deemed unlikely by the LDA parameters (ϕ, θ).

7.2.3 Alternating Optimization with MWS+LDA (M+L)

We can take a more principled approach to integrating the logic and LDA objective by alternating between optimizing (7.8) with respect to the multinomial parameters ϕ, θ while holding $\mathbf{z}$ fixed, and vice versa. The outline of this approach is shown in Algorithm 2.

The optimal ϕ, θ for fixed $\mathbf{z}$ can be easily found in closed-form as the MAP estimate of the Dirichlet posterior

[1] Since our KB is in CNF, each ground formula must be a *disjunction*, therefore flipping a single atom is guaranteed to satisfy a previously unsatisfied formula.

Algorithm 1: MaxWalkSAT weighted satisfiability solver.

Input: Weighted ground formulas G, random step probability p

Output: Best assignment $\mathbf{z}^*$

$(\mathbf{z}, \mathbf{z}^*) =$ Initialize assignment

foreach $i = 1, \ldots, maxiter$ **do**

  sample unsatisfied $g \in G$

  sample $u \sim [0, 1]$

  if $u < p$ **then**

    $\mathbf{z} \leftarrow$ randomly flip atom in g

  else

    $\mathbf{z} \leftarrow$ greedily flip atom in g according to global objective function change Δ

  end

  if $G(\mathbf{z}) > G(\mathbf{z}^*)$ **then**

    $\mathbf{z}^* \leftarrow \mathbf{z}$

  end

end

return $\mathbf{z}^*$

$$\phi_t(w) \propto \max(n_{tw} + \beta - 1, \epsilon) \tag{7.10}$$

$$\theta_j(t) \propto \max(n_{jt} + \alpha - 1, \epsilon) \tag{7.11}$$

where n_{tw} is the number of times word w is assigned to topic t in hidden topic assignments $\mathbf{z}$. Similarly, n_{jt} is the number of times topic t is assigned to a word in document j. The lower bound $\epsilon > 0$ is a small constant to ensure positivity of multinomial elements, a technical condition required by Dirichlet distributions.

Optimizing $\mathbf{z}$ while holding ϕ, θ fixed is more difficult. One can divide $\mathbf{z}$ into an "easy part" and a "difficult part." The easy part consists of all z_i which only appear in trivial groundings. For example, if the knowledge base consists of only one rule $\psi_1 = (\forall i : \mathtt{W}(i, \text{apple}) \Rightarrow \mathtt{Z}(i, 1))$, then the majority of the z_i's (those with $w_i \neq$ apple) appear in groundings which are trivially true. These z_i's only appear in the last term in (7.8). Consequently, the optimizer is simply

$$z_i = \underset{t=1\ldots T}{\operatorname{argmax}} \log(\phi_t(w_i)\theta_{d_i}(t)). \tag{7.12}$$

The difficult part of $\mathbf{z}$ consists of those z_i appearing in non-trivial groundings, subsequently in the first term of (7.8). For our simple rule (assign occurrences of "apple" to Topic 1), this division is shown in Figure 7.3. We denote the "hard" part $\mathbf{z}_{KB}$, and its optimization is performed in the inner loop of Algorithm 2. We can optimize it with MWS+LDA, a form of MWS modified to incorporate the LDA objective in the greedy selection criterion. The algorithm proceeds as in MaxWalkSAT, randomly sampling an unsatisfied clause and satisfying it via either a greedy or a random step. However, greedy steps are now evaluated using $\Delta = \Delta_{KB} + \Delta_{LDA}$, where $\Delta_{LDA}(i, t) = \log(\phi_t(w_i)\theta_{d_i}(z_i))$, which balances the gain from satisfying a logic clause and the gain of a topic assignment given the current ϕ and θ parameters, explicitly aiming to maximize the objective (7.8).

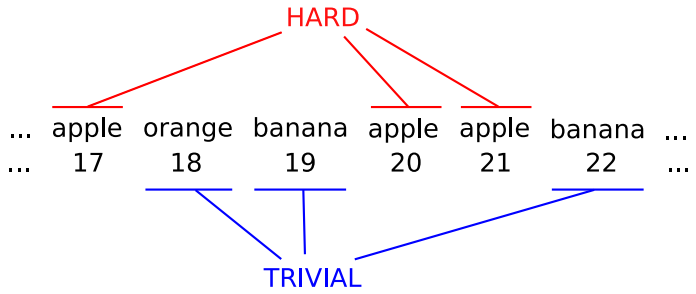


Figure 7.3: Separating out the "hard" cases ($\mathbf{z}_{KB} = \{\ldots, 17, 20, 21, \ldots\}$) for the simple rule $\texttt{W}(i, \text{apple}) \Rightarrow \texttt{Z}(i, 1)$.

Algorithm 2: Alternating optimization for LogicLDA.

Input: $\mathbf{w}, \mathbf{d}, \mathbf{o}, \alpha, \beta, KB$

Output: $(\mathbf{z}^*, \phi^*, \theta^*)$

$(\mathbf{z}, \phi, \theta) =$ Initialize from standard LDA

foreach $n = 1, 2, \ldots, N_{outer}$ **do**

  $\phi, \theta \leftarrow$ MAP estimates via (7.10) and (7.11)

  set $\mathbf{z} \setminus \mathbf{z}_{KB} \leftarrow$ argmax assignment via (7.12)

  foreach $m = 1, \ldots, N_{inner}$ **do**

    $\mathbf{z}_{KB} \leftarrow$ with M+L or Mir

  end

end

return $(\mathbf{z}, \phi, \theta)$

7.2.4 Alternating Optimization with Mirror Descent (Mir)

Optimizing the $\mathbf{z}_{KB}$ component of the original optimization problem (7.8) is challenging due to the fact that the summations over groundings $G(\psi_l)$ are potentially combinatorial. For example, on a corpus with length N, an FOL rule with k universally quantified variables will produce N^k groundings. The previously discussed approach for optimizing $\mathbf{z}_{KB}$, Alternating Optimization with MWS+LDA, requires enumerating these groundings, and may therefore run into scalability problems for certain knowledge bases. This explosion resulting from propositionalization is a well-known problem in the MLN community, and has been the subject of considerable research [Poon et al., 2008, Riedel, 2008, Singla and Domingos, 2008, Kersting et al., 2009].

For instance, one can usually greatly reduce the problem size by considering only *non-trivial* rule groundings [Shavlik and Natarajan, 2009]. As an example, the rule in Equation 7.5 is trivially true for all indices i such that $w_i \neq$ taxes, and these indices can be excluded from logic-related computation. Unfortunately, even after this pre-processing, we may still have an unacceptably large number of groundings.

Furthermore, the inclusion of the LDA terms and the scale of our domain prevent us from directly taking advantage of many techniques developed for MLNs. For example, lifted inference [Singla and Domingos, 2008, Kersting et al., 2009] approaches perform message-passing efficiently by aggregating nodes which are known to send and receive identical messages due to the special graph structure induced by propositionalization. However, the symmetries exploited by these approaches are broken in LogicLDA by the LDA terms, and the discovery of these symmetries requires initial computation over the full groundings, which may be infeasible. Lazy inference [Poon et al., 2008] is a clever caching strategy that takes advantage of the fact that, for many statistical relational learning (SRL) problems, the ground predicates are "sparse" (e.g., the academic advisor-advisee $\texttt{Advises}(x, y)$ predicate is *false* for most (x, y)). Unfortunately this insight does not apply to the query predicate $\texttt{Z}(i, t)$, which must be true for *exactly* one t for each i, lessening the usefulness this technique for LogicLDA inference.

Instead, we use stochastic gradient descent to optimize $\mathbf{z}_{KB}$, dropping a new and more scalable approach into the inner loop of Algorithm 2. The key idea is to first relax (7.8) into a continuous

optimization problem, and then randomly sample groundings from the knowledge base, such that each sampled grounding provides a stochastic gradient to the relaxed problem. Thus, we are no longer limited by the (potentially overwhelming) size of the groundings. We now describe this approach in terms of three steps.

7.2.4.1 Step 1: represent $\mathbb{1}_g$ as a polynomial

The first step is a new representation for the logic grounding indicator function $\mathbb{1}_g$. Because we assume the knowledge base KB is in Conjunctive Normal Form, each non-trivial grounding g consists of a disjunction of $\mathtt{Z}(i,t)$ atoms (positive or negative), whose logical complement $\neg g$ is therefore a *conjunction* of $\mathtt{Z}(i,t)$ atoms (each negated from the original grounding g). In order to avoid double-counting events in our polynomial, let $(\cdot)_+$ be an operator that returns a logical formula equivalent to its argument where we replace all negated atoms $\neg\mathtt{Z}(i,t)$ with equivalent disjunctions over positive atoms $\mathtt{Z}(i,1) \vee \ldots \vee \mathtt{Z}(i,t-1) \vee \mathtt{Z}(i,t+1) \vee \ldots \vee \mathtt{Z}(i,T)$, and eliminate any duplicate atoms. Next, let g_i be the *set* of atoms in g which involve index i. For example, if $g = \mathtt{Z}(0,1) \vee \mathtt{Z}(0,2) \vee \mathtt{Z}(1,0)$, then $g_0 = \{\mathtt{Z}(0,1), \mathtt{Z}(0,2)\}$. Finally, we define $z_{it} \in \{0,1\}$ to be equal to 1 if $\mathtt{Z}(i,t)$ is *true* and 0 otherwise. We can now replace each $\mathtt{Z}(i,t)$ with z_{it} in $\mathbb{1}_g$ in order to yield the polynomial

$$\mathbb{1}_g(\mathbf{z}) = 1 - \prod_{g_i \neq \emptyset} \left(\sum_{\mathtt{Z}(i,t) \in (\neg g_i)_+} z_{it} \right). \tag{7.13}$$

Note the observed variables $\mathbf{w}, \mathbf{d}, \mathbf{o}$ are no longer in (7.13) because g is a non-trivial grounding where the disjunction of $\mathbf{w}, \mathbf{d}, \mathbf{o}$ atoms is always *false*.

7.2.4.2 Step 2: relax z_{it} to continuous values

The second step is to *relax* the binary variables $z_{it} \in \{0,1\}$ to continuous values $z_{it} \in [0,1]$, with the constraint $\sum_t z_{it} = 1$ for all i. Under this relaxation, Equation (7.13) takes on values in the

interval $[0, 1]$, which can be interpreted as the expectation of the original Boolean indicator function, with the relaxed z_{it} representing multinomial probabilities. With this, we have a continuous optimization problem over $\mathbf{z}_{KB}$ (dropping terms that are constant w.r.t. $\mathbf{z}_{KB}$ in (7.8)):

$$\underset{\mathbf{z}\in\mathbb{R}^{|\mathbf{z}_{KB}|}}{\operatorname{argmax}} \sum_{l}^{L} \sum_{g\in G(\psi_l)} \lambda_l \mathbb{1}_g(\mathbf{z}) + \sum_i \sum_t z_{it} \log \phi_t(w_i)\theta_{d_i}(t) \qquad \text{s.t. } z_{it} \geq 0, \quad \sum_t^T z_{it} = 1. \tag{7.14}$$

Critically, this relaxation allows us to use gradient methods on (7.14). However a potentially huge number of groundings in $\cup_k G(\psi_k)$ may still render the full gradient impractical to compute.

7.2.4.3 Step 3: stochastic gradient descent

The third step therefore turns to *stochastic gradient descent* for scalability. Specifically we use the Entropic Mirror Descent Algorithm (EMDA) [Beck and Teboulle, 2003], of which the Exponentiated Gradient (EG) [Kivinen and Warmuth, 1997] algorithm is a special case. Unlike approaches [Collins et al., 2008] which randomly sample *training examples* to produce a stochastic approximation to the gradient, we randomly sample *terms* in (7.14). A term f is either the polynomial $\mathbb{1}_g(\mathbf{z})$ on a particular grounding g, or an LDA term $\sum_t z_{it} \log \phi_t(w_i)\theta_{d_i}(t)$ for some index i. We use a weighted sampling scheme. Let Λ be a length $L+1$ weight vector, where $\Lambda_l = \lambda_l |G(\psi_l)|$ for $l = 1 \ldots L$, and the entry $\Lambda_{L+1} = |\mathbf{z}_{KB}|$ represents the LDA part. To sample individual terms, we first choose one of the $L+1$ entries according to weights Λ. If an FOL rule ψ_l was chosen, we then sample a grounding $g \in G(\psi_l)$ uniformly. If the LDA part was chosen, we uniformly sample an index i from $\mathbf{z}_{KB}$. Once a term f is sampled, we take its gradient ∇f and perform a mirror descent update with step size η

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}. \tag{7.15}$$

The process of sampling terms and taking gradient steps is repeated until convergence, or for a prescribed number of iterations. Finally, we recover a hard $\mathbf{z}_{KB}$ assignment by

$$z_i = \underset{t}{\operatorname{argmax}}\ z_{it}. \tag{7.16}$$

The key advantage of this approach is that it requires only a means to sample groundings g for each rule ψ_k, and can avoid fully grounding the FOL rules. Our experiments show that this stochastic gradient descent approach is effective at satisfying the FOL knowledge base and optimizing the objective (7.14), when many MLN inference approaches fail due to problem size.

Finally, if groundings do not cross document boundaries (i.e., $(g_i \neq \emptyset \wedge g_j \neq \emptyset) \Rightarrow (d_i = d_j)$), additional scalability can be achieved by parallelizing the $\mathbf{z}_{KB}$ inner loop of Algorithm 2 across partitions of the documents, similar to Approximate Distributed LDA (AD-LDA) [Newman et al., 2008].

7.3 Experiments

We conduct experiments on seven datasets, summarized in Table 7.1. We compare the four LogicLDA inference methods: CGS, MWS, M+L, and Mir, as well as two simple baselines: topic modeling alone (LDA, using our own implementation of a collapsed Gibbs sampler) [Griffiths and Steyvers, 2004], and logic alone (Alchemy, using the Alchemy MLN software package) [Kok et al., 2009]. Our experiments demonstrate that: i) LogicLDA successfully incorporates logic into topic modeling, and ii) Mir is a scalable and high quality inference method for LogicLDA that works when other inference methods fail.

In all experiments, we set the parameters as follows. The logic rule weights λ are set to make the scale of the logic contribution comparable to the LDA contribution in the objective function (7.8). For the Mir inner loop in Algorithm 2, at iteration m we set step size $\eta_m = \sqrt{\frac{N_{inner}}{N_{inner}+m}}$. We fix Dirichlet parameters to $\alpha = 50/T$ and $\beta = 0.01$.

We now present the seven datasets and their associated KBs, and qualitatively assess the LogicLDA results on them. Table 7.1 contains information about the datasets themselves, as well as the total number of non-trivial groundings $|\cup_k G(\psi_k)|$ for the given KB.

7.3.1 Synthetic Must-Links and Cannot-Links (S1, S2)

These two small synthetic datasets ($N < 50$ tokens) are designed to demonstrate the ability of LogicLDA to encode Must-Link and Cannot-Link preferences, similar to Chapter 6. Here we take

Table 7.1: Descriptive statistics for LogicLDA experimental datasets: total length of corpus (in words) N, size of vocabulary W, number of documents D, number of topics T, and total number of non-trivial rule groundings $|\cup_k G(\psi_k)|$.

Dataset	N	W	D	T	$\lvert\cup_k G(\psi_k)\rvert$
S1	16	3	4	2	64
S2	32	4	4	3	192
Mac	153986	3652	2000	20	4388860
Comp	482634	8285	5000	20	6295
Con	422229	6156	2740	25	99847
Pol	733072	13196	2000	20	1204960000
HDG	2903640	13817	21352	100	47236814

Must-Link to mean that occurrences of a pair of words should be assigned to the same topic, while a Cannot-Link indicates that pairs of words should not be assigned to the same topic. In S1 we learn $T = 2$ topics, the vocabulary is {apple, banana, motorcycle}, and we encode Must-Link (apple, banana) with the KB: $\mathtt{W}(i, \text{apple}) \wedge \mathtt{W}(j, \text{banana}) \Rightarrow (\mathtt{Z}(i,t) \Leftrightarrow \mathtt{Z}(j,t))$. For the synthetic corpus, this KB results 64 in non-trivial groundings. Here and below, all FOL variables are universally quantified. Similarly, in S2 we learn $T = 3$ topics, our vocabulary is {good, bad, song, album} and we encode Cannot-Link (good, bad) as $\mathtt{W}(i, \text{good}) \wedge \mathtt{W}(j, \text{bad}) \Rightarrow (\neg\mathtt{Z}(i,t) \vee \neg\mathtt{Z}(j,t))$ (192 non-trivial groundings). Example documents for each of the two corpora are shown in Table 7.2 (within a corpus these documents types are simply repeated). We set $\alpha = 1$ and observe that all LogicLDA inference methods are able to correctly enforce these KBs, while standard LDA often fails to do so.

7.3.2 Mac vs PC (Mac)

This dataset consists of the comp.sys.ibm.pc.hardware and comp.sys.mac.hardware groups from the 20 newsgroups dataset [Lang, 1995] ($N = 2 \times 10^5, D = 2000$). Despite this natural

Table 7.2: Synthetic documents for S1 and S2 corpora.

Corpus	Synthetic documents							
S1	apple	motorcycle						
	banana	motorcycle						
S2	good	bad	song	song	good	bad	song	song
	good	bad	album	album	good	bad	album	album

split, the standard LDA topics tend to mix the words "pc" and "mac" in the same topic fairly often. An example is shown in the first column of Table 7.3. We learn $T = 20$ topics, and in order to recover topics which are more aligned with the newsgroup labels, we construct a Cannot-Link rule on "mac" and "pc": $\texttt{W}(i, \text{mac}) \wedge \texttt{W}(j, \text{pc}) \Rightarrow (\neg\texttt{Z}(i,t) \vee \neg\texttt{Z}(j,t))$ (4×10^6 non-trivial groundings). All LogicLDA inference methods are able to discover interesting topics which obey the Cannot-Link constraint, uncovering interesting and specialized Mac-related terms such as {lc, ii, quadra, lciii}. An example pair of separate topics learned by LogicLDA is shown in the second and third columns in Table 7.3.

7.3.3 Comp.* newsgroups (Comp)

This dataset is also a subset of the 20 newsgroups collection, and consists of online posts made to comp.* news groups (making it a superset of the "Mac vs PC" dataset; $N = 5 \times 10^5, D = 8285$). Standard LDA topics mix "hardware" and "software" in the same topics, but a user may wish to construct two separate topics for these terms. We learn $T = 20$ topics and our KB consists of rules to construct separate topics using {hardware, machine, memory, cpu} and {software, program, version, shareware} as seed word sets: $(\texttt{W}(i, \text{hardware}) \vee \ldots \vee \texttt{W}(i, \text{cpu})) \Rightarrow \texttt{Z}(i, 0)$ and $(\texttt{W}(i, \text{software}) \vee \ldots \vee \texttt{W}(i, \text{shareware})) \Rightarrow \texttt{Z}(i, 1)$ (6295 non-trivial groundings). The topics found by LogicLDA inference methods nicely align with our intended concepts: in addition to the seed words, Topic 0 tends to consist of other purely hardware-related terms: {drive, disk, ide, bus,

Table 7.3: Learned topics for the Mac dataset.

Standard LDA	**mac**, only, some, scsi2, chip, scsi1, used, ibm, use, 32bit software, mode, **pc**, not, so, color, since about, up, does, chips, fast, print
LogicLDA “mac”	**mac**, apple, has, new, does, internal, lc, ii, external, macs out, powerbook, box own, print, other, printer, such macintosh, article, apples, some, iisi
LogicLDA “pc”	software, hardware, use, etc, some, will, **pc**, used, one data, may, standard, not, each, two, ibm most, other, send, see, between, programs, ie

install}, while new terms in Topic 1 are software-oriented: {code, image, data, analysis}. Example learned topics are shown in Table 7.4.

Table 7.4: Learned topics for the Comp dataset.

Standard LDA	will, not, all, may, information, should, group, such, more other, **software**, no, some, been, about, see, list questions, number, has, them, already, **hardware**, need
LogicLDA "hardware"	drive, disk, memory, scsi, hard, **hardware**, drives controller, ide, cpu, system, floppy, bios, data, hd disks, bus, tape, interface, install, cdrom, feature, scsi2, dma
LogicLDA "software"	program, **software**, version, data, image, o, graphics line, code, simple, processing, ca, inc, point, center currently, shareware, points, 3d, analysis, model, sgi, digital, include

7.3.4 Congress (Con)

This dataset consists of floor-debate transcripts from the United States House of Representatives ($N = 4 \times 10^5, D = 2740$) [Thomas et al., 2006]. We learn $T = 25$ topics with the goal of discovering interesting political themes. Each speech is considered to be a document, and is labeled with the political party of the speaker: $\texttt{Speaker}(d, \text{Rep})$ or $\texttt{Speaker}(d, \text{Dem})$. We define the predicate $\texttt{HasWord}(d, w)$ to be *true* if word w appears in document d, and define the following KB:

$$\texttt{W}(i, \text{chairman}) \vee \texttt{W}(i, \text{yield}) \vee \texttt{W}(i, \text{madam}) \Rightarrow \texttt{Z}(i, 0)$$

$$\texttt{Speaker}(d, \text{Rep}) \wedge \texttt{HasWord}(d, \text{taxes}) \wedge \texttt{D}(i, d) \Rightarrow \texttt{Z}(i, 1) \vee \texttt{Z}(i, 2) \vee \texttt{Z}(i, 3)$$

$$\texttt{Speaker}(d, \text{Dem}) \wedge \texttt{HasWord}(d, \text{workers}) \wedge \texttt{D}(i, d) \Rightarrow \texttt{Z}(i, 4) \vee \texttt{Z}(i, 5) \vee \texttt{Z}(i, 6).$$

The first rule pulls uninteresting procedure words (as in "Mr. Chairman, I want to thank the gentlewoman for yielding..."), which pollute many standard LDA topics, into their own Topic 0.

The other two rules aim to discover multiple interesting political topics associated with Rep on "taxes" and Dem on "workers". The KB has a total of approximately 10^5 non-trivial groundings. As intended, Topic 0 is able to pull in other procedural words such as {gentleman, thank, colleague}, while topics other than Topic 0 appear much more meaningful without the procedural words. The special Rep +"taxes" topics uncover interesting themes: {budget, billion, deficit, health, education, security, jobs, economy, growth}, and Dem + "workers" topics uncover {pension, benefits, security, osha, safety, prices, gas}. This KB demonstrates how easy it is for LogicLDA to pull in side information to influence topic discovery.

7.3.5 Polarity (Pol)

This dataset consists of movie reviews ($N = 7 \times 10^5, D = 2000$) [Pang and Lee, 2004]. Interestingly, the synonyms "movie" and "film" appear with slightly different frequencies in positive and negative reviews, with "film" being overrepresented in the positive reviews and "movie" being overrepresented in the negative reviews. However, standard LDA tends to mix "movie" and "film" within the same topics. To simulate an analyst who wishes to study this further, we learn $T = 20$ topics with a Cannot-Link rule $\mathtt{W}(i, \text{movie}) \wedge \mathtt{W}(j, \text{film}) \Rightarrow (\neg \mathtt{Z}(i,t) \vee \neg \mathtt{Z}(j,t))$ (1.2×10^9 groundings). The size of the groundings is too large for all methods except Mir, which is able to discover topics obeying the KB. Figure 7.4 shows a word cloud[2] comparison between a standard LDA topic which mixes "movie" and "film," and a pair of Mir topics which contain "film" or "movie" only. The separate Mir topics appear to reveal sentiment differences associated with the two words. For example, the "movie" topic contains terms such as "bad", "worst", and "boring". In contrast, the "film" topic contains the words "great", "best", and "interesting".

7.3.6 Human development genes (HDG)

This dataset consists of PubMed abstracts related to stem cells, and is further filtered down to documents containing the terms "human", "development", and "gene" ($N = 3 \times 10^6, D = 2 \times 10^5$). The goal of this experiment is to discover topics centered around six concepts related to the basic

[2] http://www.wordle.net

(a) LDA *movie+film* topic (b) LogicLDA *movie* topic (c) LogicLDA *film* topic

Figure 7.4: Comparison of topics before and after applying LogicLDA on the polarity dataset

biology of human development. These concepts were specified by a biological expert who provided a handful of "seed" words, stems, and n-grams for each concept, as shown in Table 7.5. The goal of this experiment is to learn topics which are aligned with the target concepts that are of interest to the biological expert. The most probable words for each topic could then be used to help determine whether experimentally interesting genes are related to the target concept.

In order to use the provided terms, we begin by doing a small amount of manual "query expansion", scanning the vocabulary for terms sharing the same stem. This process yields the results shown in Table 7.6.

In order to assess the quality of the learned topics, our biological expert provides relevance judgments for recovered terms. The ultimate purpose of these topics is to aid in understanding relationships between genes and concepts, so we use the following relevance guideline: "Would knowing that this word is (statistically) associated with a gene increase your belief that the gene is related to the target concept?". For each target concept, we show the total number of words annotated as relevant in Table 7.7. In order to lessen annotation cost, note that only a *subset* of the vocabulary is annotated in this way. Later in this section we explain how the subset to be annotated is chosen.

As a naïve baseline, we first use standard LDA to learn $T = 50$ topics. For each target concept, we select the standard LDA topic containing the largest number of seed terms in the top 50 most probable words. The aligned topics are shown in Table 7.8, and while they are roughly aligned

Table 7.5: Concepts and terms provided by a biological expert.

Concept	Provided terms
neural	neur dendro(cyte), glia, synapse, neural crest
embryo	human embryonic stem cell, inner cell mass, pluripotent
blood	hematopoietic, blood, endothel(ium)
gastrulation	organizer, gastru(late)
cardiac	heart, ventricle, auricle, aorta
limb	limb, blastema, zeugopod, autopod, stylopod

Table 7.6: Manual "expansion" of expert-provided terms shown in Table 7.5.

Concept	Expanded terms
neural	oligodendrocyte, oligodendrocytes, oligodendrocytespecific, oligodendrogenesis oligodendroglia, oligodendroglial oligodendroglioma, oligodendrogliomas neuro, neural, neuron, dendrocyte, glia, glial, synapse, synapses, synaptic, neural crest
embryo	human embryonic stem cell, inner cell mass, pluripotent, embryonic stem cell human embryonic, inner cell
blood	hematopoietic, blood, endothelium
gastrulation	organizer, gastrulate, postgastrulation, pregastrulation midgastrulation, gastrulation, gastrulating
cardiac	heart, ventricle, auricle, aorta, ventricles, ventricular, leftventricular
limb	limb, blastema, blastemas, blastemal, zeugopod, autopod, stylopod

Table 7.7: Number of terms annotated as relevant for each target concept. Note that the vocabulary may contain terms which *would* also be annotated as relevant, but for which we have no annotation.

Concept	Number of relevant terms
neural	58
embryo	16
blood	47
gastrulation	15
cardiac	36
limb	15

with the concepts we notice different types of irrelevant terms as well. For example, the highest-overlap *neural* topic contains the terms "disease" and "disorder", which are not aligned with the developmental biology interests of the user. Further down the list are other clinical terms such as {schizophrenia, abnormal, sclerosis}.

A more directed way to use the expert knowledge would be to create "seed word" rules of the form $\texttt{W}(i, \text{hematopoietic}) \Rightarrow \texttt{Z}(i, 2)$, as in Section 7.3.3 and Chapter 5. However, some of the expert-supplied terms are in fact n-grams (e.g., "inner cell mass"). These too can be handled in a principled way within FOL via rules such as:

$$\texttt{W}(i, \text{inner}) \wedge \texttt{W}(i+1, \text{cell}) \wedge \texttt{W}(i+2, \text{mass}) \Rightarrow \texttt{Z}(i, 1)$$
$$\texttt{W}(i-1, \text{inner}) \wedge \texttt{W}(i, \text{cell}) \wedge \texttt{W}(i+1, \text{mass}) \Rightarrow \texttt{Z}(i, 1)$$
$$\texttt{W}(i-2, \text{inner}) \wedge \texttt{W}(i-1, \text{cell}) \wedge \texttt{W}(i, \text{mass}) \Rightarrow \texttt{Z}(i, 1).$$

We construct these rules for all seed terms shown in Table 7.6 and run LogicLDA inference via Collapsed Gibbs Sampling. In order to discourage non-concept words from being assigned to the special concept topics, we also associate the concept topics with a smaller α value[3]. Learned topics along with relevance annotations are shown in Table 7.9, and we can see that the words tend to be strongly related to the target concept.

In order to further improve the quality of the learned topics, we can incorporate additional domain knowledge. As mentioned above, many of the non-relevant terms are instead related to various diseases or disorders. It is therefore a natural approach to seed a separate *disease* topic in order to draw these terms away from our concept topics. We define this topic with seed words {patient, disease, parasite, chronic, virus, condition, disorder, symptom}.

Furthermore, we enforce that this new disease topic (Topic 7) not co-occur in the same sentence as our developmental biology topics. We define the vector of sentence indices $\mathbf{s} = s_1, \ldots, s_N$

[3]The concept topics have $\alpha = 0.005$, versus the standard $\alpha = 50/T$ for all other topics.

Table 7.8: High-probability terms from standard LDA topics chosen according to their overlap between the top 50 most probable terms and each set of concept seed terms. Seed terms and terms labeled as relevant are shown in bold.

neural	*embryo*	*blood*	*gastrulation*	*cardiac*	*limb*
brain	cells	growth	estrogen	development	development
system	differentiation	receptor	endometrial	muscle	muscle
nervous	bone	factor	aromatase	**heart**	heart
neurons	**stem**	receptors	endometrium	zebrafish	zebrafish
neuronal	**cell**	**endothelial**	progesterone	cardiac	cardiac
central	hematopoietic	factors	uterine	skeletal	skeletal
development	**human**	vascular	tamoxifen	embryonic	embryonic
s	marrow	hormone	implantation	defects	defects
neural	blood	pituitary	estrogens	neural	neural
human	vitro	**angiogenesis**	estradiol	gene	gene
gene	culture	human	receptor	embryos	embryos
disease	progenitor	development	endometriosis	homeobox	**homeobox**
function	stromal	ligand	stromal	limb	**limb**
cortex	lineage	transforming	steroid	formation	formation
spinal	differentiated	effects	breast	developing	developing
disorders	**embryonic**	binding	uterus	function	function
developing	vivo	ligands	women	required	required
motor	progenitors	hypoxia	cycle	mouse	mouse
cerebral	cultures	insulin-like	suture	expression	expression
glial	transplantation	proliferation	menstrual	human	human
peripheral	formation	fibroblast	antiestrogen	early	early
cortical	potential	epidermal	phase	expressed	expressed
cord	erythroid	vivo	secretory	failure	failure

Table 7.9: LogicLDA topics learned using seed term rules. Seed terms and terms labeled as relevant are shown in bold.

neural	*embryo*	*blood*	*gastrulation*	*cardiac*	*limb*
neural	cells	**blood**	**mesoderm**	muscle	**limb**
crest	**stem**	**hematopoietic**	xenopus	**heart**	shh
glial	**cell**	hypertension	**embryos**	cardiac	hox
synaptic	**embryonic**	pressure	formation	skeletal	**hedgehog**
tube	es	angiotensin	endoderm	**ventricular**	**patterning**
myelin	marrow	ace	anterior	**hypertrophy**	**homeobox**
neuron	**human**	**erythroid**	**gastrulation**	muscles	sonic
astrocytes	bone	ii	**patterning**	myogenic	hh
nervous	myeloid	hypertensive	**dorsal**	myosin	fgf8
cns	progenitors	**endothelium**	zebrafish	**myocardial**	**bud**
oligodendrocytes	progenitor	globin	**embryo**	hearts	hindbrain
schwann	ckit	ang	**plate**	left	genes
glia	cd34	renin	axis	myod	vertebrate
oligodendrocyte	hematopoiesis	system	ectoderm	failure	**limbs**
system	mast	gata1	ventral	myogenesis	signaling
synapses	pu1	shr	posterior	myocytes	**mesenchyme**
pax3	differentiation	id	early	myoblasts	pharyngeal
ncam	stromal	et1	**notochord**	**myocardium**	arch
myelination	**pluripotent**	epo	vertebrate	**cardiomyocytes**	posterior
mbp	bm	betaglobin	specification	**cardiomyopathy**	anterior
synapse	potential	red	bmp	muscular	embryos
spinal	primitive	peripheral	mesodermal	**atrial**	formation
plp	adult	enzyme	signaling	smooth	**craniofacial**
gfap	lif	at1	induction	heavy	chick
central	hsc	reninangiotensin	nodal	**aorta**	otic

analogously to $\mathbf{d}$, likewise we define a logical predicate $\mathtt{S}(i, s)$ which is *true* if $s_i = s$. For disease Topic 7 and each development concept Topics $1, \ldots, 6$, we define the "exclusion" rule

$$\mathtt{S}(i, s) \wedge \mathtt{S}(j, s) \wedge \mathtt{Z}(i, 7) \Rightarrow \neg(\mathtt{Z}(j, 1) \vee \mathtt{Z}(j, 2) \vee \mathtt{Z}(j, 3) \vee \mathtt{Z}(j, 4) \vee \mathtt{Z}(j, 5) \vee \mathtt{Z}(j, 6)).$$

This rule raises two important points. First, this rule demonstrates how easy it is to include completely new sources of information such as sentence boundaries. Second, this rule raises potential scalability issues. If the total number of sentences is S and the maximum sentence length is M, this rule yields $O(SM^2)$ non-trivial groundings.

For the quantitative inference experiments discussed in Section 7.3.7, we only instantiate this rule for development concept 2 (*blood*). In Table 7.8 we can see that this topic contains the non-relevant disease terms "pressure" and "hypertension". After adding our sentence exclusion logic, these terms no longer appear in Topic 2.

We also wish to ensure that the concepts are aligned with the biologist's interest in development. We define a *development* Topic 6 using seed words {differentiation, maturation, formation, differentiates, develops}. For each target concept topic t, we then define an "inclusion" rule

$$\mathtt{Sentence}(i, i_1, \ldots, i_{S_k}) \wedge \neg\mathtt{Z}(i_1, 6) \wedge \ldots \wedge \neg\mathtt{Z}(i_{S_k}, 6) \Rightarrow \neg\mathtt{Z}(i, t)$$

where `Sentence` is *true* if its corpus index variable arguments constitute a single complete sentence[4]. In English, this rule says: "If the *development* topic does not occur in a given sentence, the biological concept Topic t cannot occur in that sentence either."

For each of the KBs shown in Table 7.10, we perform 10 inference runs with different random number seeds, using collapsed Gibbs sampling for LDA and Alternating Optimization with Mirror Descent for all other KBs. For each target concept, we select the union of the top 50 most probable words for the associated topic over all random seeds and KBs. This set of words is then annotated for relevance by the biological expert. Using these judgements as positive labels, Table 7.11 shows the mean accuracy of the top 50 most probable words over 10 runs for each KB and target concept.

[4]This means the we require separate instantiations of this rule for each possible sentence length, or we must allow predicates to take a *set* of variables as an argument.

Table 7.10: The different KBs used for the relevance assessment experiment. Each rule type is instantiated for all biological concept topics.

KB	Rule types
INCL+EXCL	Seed, inclusion, exclusion
INCL	Seed, inclusion
EXCL	Seed, exclusion
SEED	Seed
LDA	No logic (choose topics according to seed word overlap)

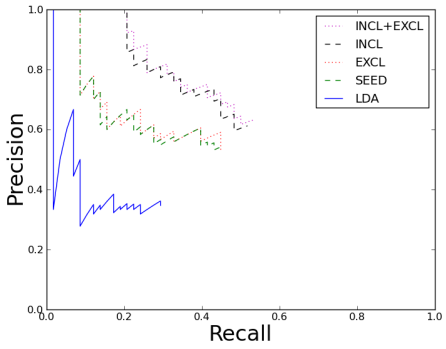
Table 7.11: Mean accuracy of top 50 terms for each KB and target concept, taken over 10 runs with different random seeds. For each target concept, bolded entries are statistically significantly better with $p < 0.001$ according to Tukey's Honestly Significant Difference (HSD) test.

	LogicLDA KBs				
	INCL+EXCL	INCL	EXCL	SEED	LDA
neural	**0.59**	**0.57**	**0.54**	**0.54**	0.31
embryo	**0.24**	**0.24**	**0.23**	**0.23**	0.07
blood	**0.46**	**0.47**	0.40	0.39	0.13
gastrulation	**0.18**	**0.18**	**0.16**	**0.16**	0.00
cardiac	**0.36**	**0.37**	**0.34**	**0.35**	0.08
limb	**0.18**	**0.18**	**0.15**	**0.14**	0.09

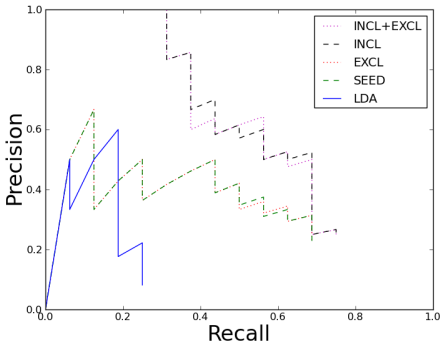
These results show that all LogicLDA approaches outperform our standard LDA baseline, and that KBs using the inclusion rule outperform all others on *blood*.

In order to get a more detailed picture of this performance, we can plot the precision and recall of the top n most probable words for each topic, as $n = 1, \ldots, 50$. We only take up to the 50 most probable words for each topic because words beyond this threshold may not be labeled. Precision is the proportion of returned words which are actually relevant, while recall is the proportion of relevant words which are actually returned. The plots for a single random seed run are shown in Figure 7.5, and tend to show an advantage for KBs which make use of the sentence inclusion rule.

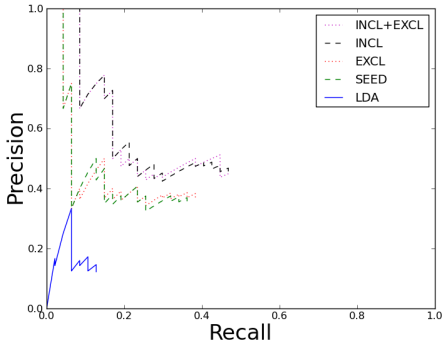
This biological text mining application has shown the advantage of LogicLDA versus standard LDA for discovering terms related to a target concept. In general, the topics learned using seed rules found more relevant terms than the LDA baseline. The addition of our sentence inclusion rule enhances topic quality even further, as measured by both top 50 word accuracy and on precision recall plots. These results highlight the usefulness of logical domain knowledge in adapting topic modeling to a specific task.



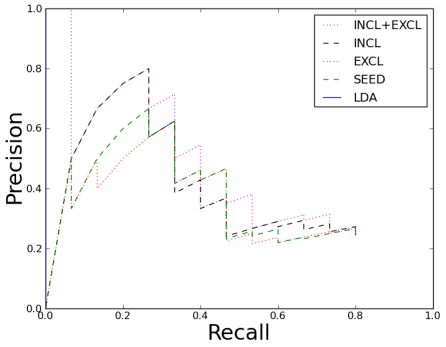
(a) *neural* concept.



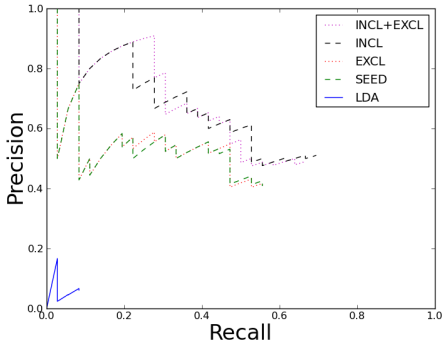
(b) *embryo* concept.



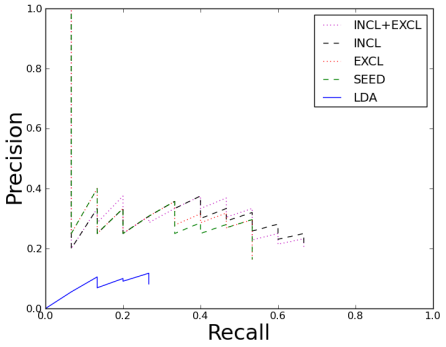
(c) *blood* concept.



(d) *gastrulation* concept.



(e) *cardiac* concept.



(f) *limb* concept.

Figure 7.5: Precision-recall plots from a single inference run for each KB and target concept, taken up to the top 50 most probable words. Note that not all words in the vocabulary are annotated.

7.3.7 Evaluation of inference

We now present results on the quantitative performance of the different inference methods. The criterion we use is the objective function (7.8), which combines logic and LDA terms. A good inference method should attain a large objective function value, because this indicates the recovery of a latent topic assignment $\mathbf{z}$ which reflects both corpus statistics (via the LDA terms) and the logical domain knowledge (via the MLN terms). Furthermore, we would prefer an approach which does well across different datasets and KBs, including cases where there are many non-trivial rule groundings. The goal of this experiment is to evaluate the performance of the presented inference approaches with respect to these two criteria.

Table 7.12 shows the objective function (7.8) values for $(\mathbf{z}, \phi, \theta)$ found by each inference method for each dataset+KB. For each entry, we perform 10 trials with different random seeds (note that *every* method is stochastic in nature), and report the mean and standard deviation. An entry of NC (Not Complete) indicates that each single trial failed to complete within 24 hours on a compute server with 2.33 GHz CPU and 16 GB RAM. Only three out of 10 Alchemy-on-Mac trials completed in 24 hours.

The results in Table 7.12 further demonstrate that: i) All four LogicLDA inference methods are better at optimizing the joint logic and LDA objective (7.8) than topic modeling alone (LDA) or logic alone (Alchemy), and therefore better at integrating FOL into topic modeling. ii) Mir is the only logic inference method able to handle the larger Pol and HDG datasets, and its objective values are consistently among the best of all methods. We conclude that Mir for LogicLDA is a viable and valuable method for combining logic into topic modeling.

7.4 Encoding LDA variants

We have presented LogicLDA, a framework for the inclusion of domain knowledge in topic modeling via FOL. To demonstrate its flexibility, it is illustrative to show how several prior LDA extensions can be (approximately) re-formulated with LogicLDA[5]:

[5]It should be pointed out, however, that LogicLDA as presented is for *inference* only. We assume that rule weights are user-supplied, not learned.

Table 7.12: Comparison of different inference methods for LogicLDA, LDA, and Alchemy on the objective function (7.8). Each row corresponds to a dataset+KB, and the first column contains the objective function magnitude. Parenthesized values are standard deviations over 10 trials with different random seeds, and NC indicates a failed run. The best results for each dataset+KB are bolded (significance $p < 10^{-6}$ using Tukey's Honestly Significant Difference (HSD) test).

	LogicLDA					
	Mir	M+L	CGS	MWS	LDA	Alchemy
S1 $\times 10$	$\mathbf{8.64(1.35)}$	$\mathbf{9.08(0.00)}$	$\mathbf{9.08(0.00)}$	$\mathbf{8.19(1.80)}$	$3.83(4.00)$	$1.43(0.80)$
S2 $\times 10^2$	$\mathbf{3.44(0.05)}$	$\mathbf{3.49(0.03)}$	$\mathbf{3.47(0.06)}$	$\mathbf{3.49(0.03)}$	$2.17(1.51)$	$2.72(0.08)$
Mac $\times 10^5$	$\mathbf{3.35(0.01)}$	$\mathbf{3.36(0.01)}$	$2.48(0.03)$	$2.49(0.02)$	$2.24(0.19)$	$-5.77^{NC}_{(0.00)}$
Comp $\times 10^6$	$\mathbf{2.48(0.00)}$	$\mathbf{2.48(0.00)}$	$2.20(0.01)$	$0.15(0.10)$	$-0.84(0.10)$	NC
Con $\times 10^5$	$\mathbf{18.52(0.16)}$	$\mathbf{19.04(0.04)}$	$16.68(0.05)$	$-3.79(0.81)$	$-4.07(0.79)$	NC
Pol $\times 10^8$	$\mathbf{9.63(0.00)}$	NC	NC	NC	$\mathbf{9.56(0.15)}$	NC
HDG $\times 10^6$	$\mathbf{116.80(0.03)}$	NC	NC	NC	$64.04(1.66)$	NC

7.4.1 Concept-Topic Model

The Concept-Topic Model [Chemudugunta et al., 2008] ties special *concept* topics to specific concepts by constraining these special topics to only emit words from carefully chosen subsets of the vocabulary. For a given concept, let the special words be $w_{c1}, w_{c2}, \ldots, w_{cn}$ and the special topic be t. Then the rule $\forall i\ \mathtt{Z}(i,t) \Rightarrow (\mathtt{W}(i,w_{c1}) \vee \mathtt{W}(i,w_{c2}) \vee \ldots \vee \mathtt{W}(i,w_{cn}))$ encodes the desired constraint.

7.4.2 Hidden Topic Markov Model

The Hidden Topic Markov Model (HTMM) [Gruber et al., 2007] enforces the condition that the same topic be used for an entire sentence and allows topic transitions only between sentences, encoding the assumption that words within a sentence are topically coherent. Using the sentence predicate $\mathtt{S}(i,s)$ previously discussed in the experiments, we can express this as $\forall i,j,s,t\ \mathtt{S}(i,s) \wedge \mathtt{S}(j,s) \wedge \mathtt{Z}(i,t) \Rightarrow \mathtt{Z}(j,t)$. The probabilities of inter-sentence topic transitions can be set by carefully choosing weights for rules of the form $\forall i,s\ \mathtt{S}(i,s) \wedge \neg\mathtt{S}(i+1,s) \wedge \mathtt{Z}(i,t) \Rightarrow \mathtt{Z}(i+1,t')$ for all transition pairs (t,t').

7.4.3 Restricted topic models

ΔLDA [Andrzejewski et al., 2007], Discriminative LDA [Lacoste-Julien et al., 2008], and Labeled LDA [Ramage et al., 2009] employ special "restricted" topics which can be used only in specially labeled documents, with the intention that these special topics model document aspects associated with the document label. If topic t should only be used in documents with label ℓ, we can encode this constraint as $\forall i,d\ \mathtt{Z}(i,t) \wedge \mathtt{D}(i,d) \Rightarrow \mathtt{HasLabel}(d,\ell)$.

7.5 Summary

The LogicLDA model provides a general mechanism for expressing domain knowledge. Logical rules can induce dependencies between topic assignments, as well as exploit arbitrary side information. We have shown several examples of how a user might guide the recovery of topics

over different datasets and knowledge bases. One potential pitfall of LogicLDA is the threat of a combinatorial explosion in the number of ground clauses. The quantitative experiments show that the scalable Mir inference algorithm is applicable where other inference methods fail.

KEY IDEAS

- ◇ LogicLDA enables the user to specify domain knowledge in First-Order Logic (FOL):
  - ▷ Generalizes some existing LDA variants,
  - ▷ Allows relational dependencies among z_i,
  - ▷ Captures side information (e.g., document labels or sentence information).
- ◇ Inference can be difficult due to combinatorial explosion of ground clauses.
- ◇ Alternating Optimization with Mirror Descent scales to cases where other methods fail.

Chapter 8

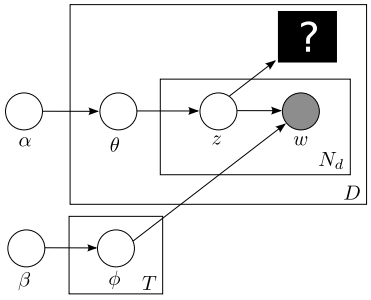
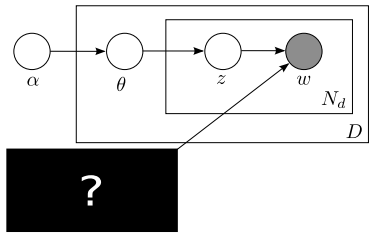
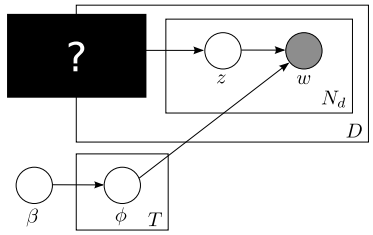
Conclusion

This thesis has presented general mechanisms for incorporating domain knowledge into topic modeling. Like standard unsupervised clustering, purely unsupervised topic modeling may not recover an underlying structure that is relevant to the user. However, the goal of many topic modeling applications is to explore or better understand a dataset, and these requirements are not always adequately addressed by the standard supervised machine learning setting. Topic modeling with domain knowledge allows the user to combine unsupervised pattern discovery with prior knowledge or preferences about the patterns to be discovered.

8.1 Summary

In Chapter 1 we introduced the fundamental concepts of topic modeling, and Chapter 2 covered specific details of Latent Dirichlet Allocation (LDA). Chapter 3 discussed existing variations on the base LDA model, and defined loose categories for these extensions depending on which aspect of LDA they modified. We now return to this framework in order to summarize the contributions of the thesis. Table 8.1 shows the models developed in this thesis along with their relationships to the model categories defined in Chapter 3.

Table 8.1: Models developed in this thesis, viewed in the context of the LDA variant categories introduced in Chapter 3. For each model, the check marks indicate which aspects of LDA are modified with domain knowledge.

Variant family	Diagram	ΔLDA	Topic-in-set	Dirichlet Forest	LogicLDA
LDA+X					✓
ϕ-side			✓	✓	✓
θ-side		✓	✓		✓

Chapter 4 introduced ΔLDA, which encodes domain knowledge about special restricted topics which the user expects to observe only in a subset of the documents. Influencing the document-topic distribution θ in this way places this application in the θ-side family of models. We apply ΔLDA to a statistical debugging dataset, where a single document corresponds to a program execution, and we have document labels indicating whether the run was successful or not. By creating special "buggy" topics restricted to appear in failing runs only, we are able to cluster runs by root cause of failure and gain insights into the individual bugs.

In Chapter 5, we described Topic-in-set knowledge, which allows finer-grained control over the individual z_i topic assignments. Similar to ΔLDA, this mechanism can be used to restrict topics from appearing in certain documents, encoding θ-side preferences. Topic-in-set knowledge can also be used in a ϕ-side context to influence which topics can be associated with particular vocabulary words. We explore a simple yet effective approach where certain "seed" words related to a target concept are encouraged to be assigned to a given topic. Other related terms are then pulled into this topic by statistical association with the seed words. This yields both an expanded set of terms related to the target concept and the associations between the concept and individual documents.

The Dirichlet Forest prior is the subject of Chapter 6. ϕ-side domain knowledge can be encoded by using this distribution in place of the standard Dirichlet prior on the topic-word multinomial ϕ. The Dirichlet Forest prior generalizes the Dirichlet, allowing the encoding of constraints about *relationships* among the vocabulary words. A Must-Link constraint between a pair of words encourages their probabilities to be similar within each topic, while a Cannot-Link constraint discourages a pair of words from both having large probabilities within any given topic. These constraints can be further combined into high-level composite operations, such as **Split**, **Merge**, and **Isolate**. We demonstrate how domain knowledge can improve topic quality in two distinct settings. In the first, domain knowledge is constructed in an interactive context, with the user iteratively learning topics, adding domain knowledge, and re-learning topics. In the second, the domain knowledge is supplied off-line from an existing structured knowledge source.

Finally, we discussed LogicLDA in Chapter 7. LogicLDA allows the user to express domain knowledge as a knowledge base (KB) of weighted rules in first-order logic (FOL). The mechanism used is equivalent to a Markov Logic Network (MLN), and the resulting LogicLDA model can be considered to be the *product* of a stand-alone LDA model and a stand-alone MLN model. The inferred topic assignment $\mathbf{z}$ is chosen to maximize an objective function consisting of the LDA and MLN components of LogicLDA. Depending on the user-defined weights associated with these rules, $\mathbf{z}$ will therefore reflect the statistical properties of the corpus (due to the LDA component), while at the same time attempting to satisfy the user-defined rules (due to the MLN component). FOL is general, and can be used to incorporate various types of side information (including document labels, sentence boundaries, and syntactic parses), allowing LogicLDA to express domain knowledge across all three of our informal categories (LDA+X, ϕ-side, and θ-side) as well as to generalize some existing LDA variants.

We apply LogicLDA to a biological text mining task motivated by the informatics needs of a biological researcher. In this application we begin with "seed" words for a set of target biological concepts. While the use of seed words significantly improves the relevance of the learned topics (versus a standard LDA baseline), we still notice the presence of related but off-concept words in our topics. Furthermore, these off-concept words are themselves somewhat semantically coherent (e.g., we see words related to *blood disease* in our *blood* topic). In order to refine our topics further, we exploit our knowledge of sentence boundaries, seeding a *disease* topic and forbidding it to co-occur within the same sentence as our target biological topic. The resulting topics are indeed more relevant. This rule highlights the flexibility of FOL, both in terms of the types of constraints it can express and the ease with which side information can be incorporated.

8.2 Software

In order to improve the usefulness of topic modeling for researchers and practitioners alike, inference source code for topic modeling with domain knowledge has been made available on the

web. Table 8.2 summarizes the availability of implementations for the models presented in this thesis.

Table 8.2: Released research code.

Model	Implementation	Availability
ΔLDA	Python C extension	http://mloss.org/software/view/161/
		http://pages.cs.wisc.edu/~andrzeje/research/delta_lda.html
Topic-in-set[1]	Python C extension	http://github.com/davidandrzej/pSSLDA
		http://pages.cs.wisc.edu/~andrzeje/research/zl_lda.html
Dirichlet Forest	Python C++ extension	http://mloss.org/software/view/204/
		http://pages.cs.wisc.edu/~andrzeje/research/df_lda.html
LogicLDA	Java	(in submission)

8.3 Conclusion and future directions

Unsupervised data modeling techniques can fill an important application niche for users struggling to make sense of datasets in the absence of a clear supervised learning task. However, as in clustering, unsupervised topic modeling approaches may not discover underlying structure which is truly relevant or helpful to the user. Therefore in this thesis we have developed techniques allowing users to augment topic modeling with domain knowledge. These approaches help prevent topic modeling from being a "black box", giving the user the tools to help adapt topic modeling to their particular application. This line of work aims to make the most of both human and machine resources in order to discover important and useful topics in data. While the models presented here represent significant advances towards this goal, there are still many interesting opportunities for further improvement.

[1] This implementation allows parallelized inference following the approach of Approximate Distributed LDA (AD-LDA) [Newman et al., 2008].

While the inclusion of user preferences or constraints can be a powerful tool, it is not without cost in user effort and time. In order to make the most of user input, it may be advantageous to examine recent advances in *active learning* [Settles, 2008], in which the system makes specific feedback requests to the user which are carefully chosen to maximize the value of user feedback. In an exploratory data analysis setting, it could be quite beneficial to have the system itself suggest candidate topic refinements to the user. The adaptation of existing active learning strategies to topic modeling presents unique challenges, in part due to recent human studies [Chang et al., 2009] suggesting that the purely data-driven objective functions commonly used to evaluate topic models (e.g., held-aside log-likelihood as discussed in Chapter 1) are not necessarily good proxies for human interpretability.

The combination of logic and topic modeling provides interesting directions for future work. The definition of additional query atoms other than $\mathtt{Z}(i, t)$ (as in MLNs) could allow the formulation of relational inference problems in LogicLDA. For example, we could define an unobserved predicate $\mathtt{Citation}(d, d')$ to be *true* if document d cites document d'. Another potential direction is to add predicates and rules encoding syntactic knowledge, such as dependency parse information. For example, we may wish to have the fact that w_i is the nominal subject of w_j influence the topic assignments z_i, z_j.

We could also incorporate user guidance in situations where we have *multi-modal* data. For example, in a biological setting we may have both experimental data and scientific text related to a set of genes. While latent variable modeling is a powerful mechanism for jointly modeling the data, it would be useful to have a general framework for capturing user preferences with respect to the *connections* or *interactions* across the data types. In our simple example, the user may believe that genes with similar text representations are more likely to share biological function. A system incorporating preferences across data types could be especially helpful in cases where one type of data is more easily comprehensible to the user, allowing the guidance of model learning using intuitions based on the more familiar type of data.

LIST OF REFERENCES

[Andrzejewski et al., 2007] Andrzejewski, D., Mulhern, A., Liblit, B., and Zhu, X. (2007). Statistical debugging using latent topic models. In *European Conference on Machine Learning*, pages 6–17. Springer-Verlag.

[Andrzejewski and Zhu, 2009] Andrzejewski, D. and Zhu, X. (2009). Latent Dirichlet allocation with topic-in-set knowledge. In *HLT-NAACL Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 43–48. ACL Press.

[Andrzejewski et al., 2009] Andrzejewski, D., Zhu, X., and Craven, M. (2009). Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *International Conference on Machine Learning*, pages 25–32. Omnipress.

[Asuncion et al., 2008] Asuncion, A., Smyth, P., and Welling, M. (2008). Asynchronous distributed learning of topic models. In *Advances in Neural Information Processing Systems*, pages 81–88. MIT Press.

[Asuncion et al., 2009] Asuncion, A., Welling, M., Smyth, P., and Teh, Y. W. (2009). On smoothing and inference for topic models. In *Uncertainty in Artificial Intelligence*, pages 27–34. AUAI Press.

[Basu et al., 2006] Basu, S., Bilenko, M., Banerjee, A., and Mooney, R. J. (2006). Probabilistic semi-supervised clustering with constraints. In Chapelle, O., Schölkopf, B., and Zien, A., editors, *Semi-Supervised Learning*, pages 71–98. MIT Press.

[Basu et al., 2008] Basu, S., Davidson, I., and Wagstaff, K., editors (2008). *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC Press.

[Beck and Teboulle, 2003] Beck, A. and Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167 – 175.

[Bekkerman et al., 2007] Bekkerman, R., Raghavan, H., Allan, J., and Eguchi, K. (2007). Interactive clustering of text collections according to a user-specified criterion. In *International Joint Conference on Artificial intelligence*, pages 684–689. Morgan Kaufmann Publishers Inc.

[Bhattacharya and Getoor, 2006] Bhattacharya, I. and Getoor, L. (2006). A latent Dirichlet model for unsupervised entity resolution. In *SIAM Conference on Data Mining*, pages 47–58. SIAM Press.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[Blei and Lafferty, 2006a] Blei, D. and Lafferty, J. (2006a). Correlated topic models. In *Advances in Neural Information Processing Systems*, pages 147–154. MIT Press.

[Blei and McAuliffe, 2008] Blei, D. and McAuliffe, J. (2008). Supervised topic models. In *Advances in Neural Information Processing Systems*, pages 121–128. MIT Press.

[Blei et al., 2003a] Blei, D., Ng, A., and Jordan, M. (2003a). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

[Blei et al., 2003b] Blei, D. M., Griffiths, T. L., Jordan, M. I., and Tenenbaum, J. B. (2003b). Hierarchical topic models and the nested Chinese restaurant process. In *Advances in Neural Information Processing Systems*, pages 17–24. MIT Press.

[Blei and Jordan, 2003] Blei, D. M. and Jordan, M. I. (2003). Modeling annotated data. In *ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 127–134. ACM Press.

[Blei and Lafferty, 2006b] Blei, D. M. and Lafferty, J. D. (2006b). Dynamic topic models. In *International Conference on Machine Learning*, pages 113–120. Omnipress.

[Boyd-Graber and Blei, 2008] Boyd-Graber, J. and Blei, D. (2008). Syntactic topic models. In *Advances in Neural Information Processing Systems*, pages 185–192. MIT Press.

[Boyd-Graber et al., 2007] Boyd-Graber, J., Blei, D., and Zhu, X. (2007). A topic model for word sense disambiguation. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1024–1033. ACM Press.

[Bron and Kerbosch, 1973] Bron, C. and Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Communications of the Association for Computing Machinery*, 16(9):575–577.

[Cao and Fei-Fei, 2007] Cao, L. and Fei-Fei, L. (2007). Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *International Conference on Computer Vision*, pages 1–8. Springer.

[Caruana et al., 2006] Caruana, R., Elhawary, M. F., Nguyen, N., and Smith, C. (2006). Meta clustering. In *IEEE International Conference on Data Mining*, pages 107–118. IEEE Computer Society.

[Chang and Blei, 2009] Chang, J. and Blei, D. (2009). Relational topic models for document networks. In *International Conference on Artificial Intelligence and Statistics*, volume 5, pages 81–88. Journal of Machine Learning Research.

[Chang et al., 2009] Chang, J., Boyd-Graber, J., Wang, C., Gerrish, S., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, pages 288–296. MIT Press.

[Chechik and Tishby, 2002] Chechik, G. and Tishby, N. (2002). Extracting relevant structures with side information. In *Advances in Neural Information Processing Systems*, pages 857–864. MIT press.

[Chemudugunta et al., 2008] Chemudugunta, C., Holloway, A., Smyth, P., and Steyvers, M. (2008). Modeling documents by combining semantic concepts with unsupervised statistical learning. In *International Semantic Web Conference*, pages 229–244. Springer.

[Collins et al., 2008] Collins, M., Globerson, A., Koo, T., Carreras, X., and Bartlett, P. L. (2008). Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks. *Journal of Machine Learning Research*, 9:1775–1822.

[Dasgupta and Ng, 2009] Dasgupta, S. and Ng, V. (2009). Topic-wise, sentiment-wise, or otherwise? Identifying the hidden dimension for unsupervised text classification. In *Empirical Methods in Natural Language Processing*, pages 580–589. ACL Press.

[Dasgupta and Ng, 2010] Dasgupta, S. and Ng, V. (2010). Mining clustering dimensions. In *International Conference on Machine Learning*, pages 263–270. Omnipress.

[Daumé, 2009] Daumé, III, H. (2009). Markov random topic fields. In *Joint Conference of Association for Computational Linguistics and International Joint Conference on Natural Language Processing (short papers)*, pages 293–296. ACL Press.

[Dennis III, 1991] Dennis III, S. Y. (1991). On the hyper-Dirichlet type 1 and hyper-Liouville distributions. *Communications in Statistics – Theory and Methods*, 20(12):4069–4081.

[Dietz et al., 2007] Dietz, L., Bickel, S., and Scheffer, T. (2007). Unsupervised prediction of citation influences. In *International Conference on Machine Learning*, pages 233–240. ACM Press.

[Dietz et al., 2009] Dietz, L., Dallmeier, V., Zeller, A., and Scheffer, T. (2009). Localizing bugs in program executions with graphical model. In *Advances in Neural Information Processing Systems*, pages 468–476. MIT Press.

[Domingos and Lowd, 2009] Domingos, P. and Lowd, D. (2009). Markov logic: An interface layer for artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–155.

[Duda et al., 2000] Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification (2nd Edition)*. Wiley-Interscience.

[EXIF Tag Parsing Library,] EXIF Tag Parsing Library. http://libexif.sf.net/.

[Fei-Fei and Perona, 2005] Fei-Fei, L. and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 524–531. IEEE Computer Society.

[Gelman et al., 2004] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian data analysis*. Chapman & Hall/CRC, second edition.

[Goldberg et al., 2009] Goldberg, A., Fillmore, N., Andrzejewski, D., Xu, Z., Gibson, B., and Zhu, X. (2009). May all your wishes come true: A study of wishes and how to recognize them. In *Human Language Technologies: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 263–271. ACL Press.

[Gondek and Hofmann, 2004] Gondek, D. and Hofmann, T. (2004). Non-redundant data clustering. In *IEEE International Conference on Data Mining*, pages 75–82. IEEE Computer Society.

[Graves et al., 2000] Graves, T. L., Karr, A. F., Marron, J. S., and Siy, H. (2000). Predicting fault incidence using software change history. *IEEE Transactions on Software Engineering*, 26(7):653–661.

[Griffiths et al., 2004] Griffiths, T., Steyvers, M., Blei, D., and Tenenbaum, J. (2004). Integrating topics and syntax. In *Advances in Neural Information Processing Systems*, pages 537–544. MIT Press.

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

[Griggs et al., 1988] Griggs, J. R., Grinstead, C. M., and Guichard, D. R. (1988). The number of maximal independent sets in a connected graph. *Discrete Mathematics*, 68(2-3):211–220.

[Gruber et al., 2007] Gruber, A., Rosen-Zvi, M., and Weiss, Y. (2007). Hidden topic Markov models. In *International Conference on Artificial Intelligence and Statistics*, pages 163–170. Omnipress.

[H. Do, 2005] H. Do, S. Elbaum, G. R. (2005). Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact. *Empirical Software Engineering: An International Journal*, 10(4):405–435.

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer.

[Hinton, 2002] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

[Hofmann, 1999] Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, pages 289–296. AUAI Press.

[Horwitz et al., 1988] Horwitz, S., Reps, T. W., and Binkley, D. (1988). Interprocedural slicing using dependence graphs (with retrospective). In *Best of ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 229–243. ACM Press.

[Kersting et al., 2009] Kersting, K., Ahmadi, B., and Natarajan, S. (2009). Counting belief propagation. In *Uncertainty in Artificial Intelligence*, pages 277–284. AUAI Press.

[Kivinen and Warmuth, 1997] Kivinen, J. and Warmuth, M. K. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63.

[Kok et al., 2009] Kok, S., Sumner, M., Richardson, M., Singla, P., Poon, H., Lowd, D., Wang, J., and Domingos, P. (2009). The Alchemy System for Statistical Relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA.

[Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. MIT Press.

[Lacoste-Julien et al., 2008] Lacoste-Julien, S., Sha, F., and Jordan, M. (2008). DiscLDA: Discriminative learning for dimensionality reduction and classification. In *Advances in Neural Information Processing Systems*, pages 897–904. MIT Press.

[Lang, 1995] Lang, K. (1995). Newsweeder: Learning to filter netnews. In *International Conference on Machine Learning*, pages 331–339. Morgan Kaufmann.

[Li and McCallum, 2006] Li, W. and McCallum, A. (2006). Pachinko allocation: DAG-structured mixture models of topic correlations. In *International Conference on Machine Learning*, pages 577–584. Omnipress.

[Liblit, 2007] Liblit, B. (2007). *Cooperative Bug Isolation: Winning Thesis of the 2005 ACM Doctoral Dissertation Competition*, volume 4440 of *Lecture Notes in Computer Science*. Springer.

[Liblit, 2008] Liblit, B. (2008). Reflections on the role of static analysis in Cooperative Bug Isolation. In *International Static Analysis Symposium*, pages 18–31. Springer.

[Liblit et al.,] Liblit, B., Naik, M., Zheng, A. X., Aiken, A., and Jordan, M. I. Scalable statistical bug isolation. In *ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 15–26.

[Linstead et al., 2007] Linstead, E., Rigor, P., Bajracharya, S., Lopes, C., and Baldi, P. (2007). Mining eclipse developer contributions via author-topic models. In *International Workshop on Mining Software Repositories*, pages 30–30. IEEE Computer Society.

[Liu et al., 2009] Liu, Y., Niculescu-Mizil, A., and Gryc, W. (2009). Topic-link LDA: joint models of topic and author community. In *International Conference on Machine Learning*, pages 665–672. Omnipress.

[MacKay, 2003] MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

[Manning and Schütze, 1999] Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press.

[McCallum et al., 2005] McCallum, A., Corrada-Emmanuel, A., and Wang, X. (2005). Topic and role discovery in social networks. In *International Joint Conference on Artificial intelligence*, pages 786–791. Morgan-Kaufmann.

[Miller, 1995] Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

[Mimno et al., 2007] Mimno, D., Li, W., and McCallum, A. (2007). Mixtures of hierarchical topics with pachinko allocation. In *International Conference on Machine Learning*, pages 633–640. ACM Press.

[Mimno and McCallum, 2007] Mimno, D. and McCallum, A. (2007). Expertise modeling for matching papers with reviewers. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 500–509. ACM Press.

[Mimno and McCallum, 2008] Mimno, D. and McCallum, A. (2008). Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *Uncertainty in Artificial Intelligence*, pages 411–418. AUAI Press.

[Mimno et al., 2008] Mimno, D., Wallach, H., and McCallum, A. (2008). Gibbs sampling for logistic normal topic models with graph-based priors. In *NIPS Workshop on Analyzing Graphs*.

[Minka and Lafferty, 2002] Minka, T. and Lafferty, J. (2002). Expectation-propagation for the generative aspect model. In *Uncertainty in Artificial Intelligence*, pages 352–359. Morgan Kaufmann.

[Minka, 1999] Minka, T. P. (1999). The Dirichlet-tree distribution. Technical report. http://research.microsoft.com/~minka/papers/dirichlet/minka-dirtree.pdf.

[Minka, 2000] Minka, T. P. (2000). Estimating a Dirichlet distribution. Technical report, Microsoft Research.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

[Mosimann, 1962] Mosimann, J. E. (1962). On the compound multinomial distribution, the multivariate beta-distribution, and correlations among proportions. *Biometrika*, 49(1-2):65–82.

[Munson and Khoshgoftaar, 1992] Munson, J. and Khoshgoftaar, T. (1992). The detection of fault-prone programs. *IEEE Transactions on Software Engineering*, 18(5):423–433.

[Neal, 1998] Neal, R. M. (1998). Markov chain sampling methods for Dirichlet process mixture models. Technical Report 9815, Deptartment of Statistics, University of Toronto.

[Newman et al., 2008] Newman, D., Asuncion, A., Smyth, P., and Welling, M. (2008). Distributed inference for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1081–1088. MIT Press.

[Newman et al., 2007] Newman, D., Hagedorn, K., Chemudugunta, C., and Smyth, P. (2007). Subject metadata enrichment using statistical topic models. In *ACM/IEEE Joint Conference on Digital Libraries*, pages 366–375. ACM Press.

[Newman et al., 2009] Newman, D., Karimi, S., and Cavedon, L. (2009). External evaluation of topic models. In *Australasian Document Computing Symposium*, pages 11–18. School of Information Technologies, University of Sydney.

[Ng and Jordan, 2001] Ng, A. Y. and Jordan, M. I. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*, pages 841–848. MIT Press.

[Pang and Lee, 2004] Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Association for Computational Linguistics*, pages 271–278. ACL Press.

[Poon and Domingos, 2006] Poon, H. and Domingos, P. (2006). Sound and efficient inference with probabilistic and deterministic dependencies. In *AAAI Conference on Artificial Intelligence*, pages 458–463. AAAI Press.

[Poon et al., 2008] Poon, H., Domingos, P., and Sumner, M. (2008). A general method for reducing the complexity of relational inference and its application to MCMC. In *AAAI Conference on Artificial Intelligence*, pages 1075–1080. AAAI Press.

[Ramage et al., 2009] Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *Empirical Methods in Natural Language Processing*, pages 248–256. ACL Press.

[Rand, 1971] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850.

[Richardson and Domingos, 2006] Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine Learning*, 62(1-2):107–136.

[Riedel, 2008] Riedel, S. (2008). Improving the accuracy and efficiency of MAP inference for Markov logic. In *Uncertainty in Artificial Intelligence*, pages 468–475. AUAI Press.

[Rosen-Zvi et al., 2004] Rosen-Zvi, M., Griffiths, T., Steyvers, M., and Smyth, P. (2004). The author-topic model for authors and documents. In *Uncertainty in Artificial Intelligence*, pages 487–494. AUAI Press.

[Rothermel et al., 2006] Rothermel, G., Elbaum, S., Kinneer, A., and Do, H. (2006). Software-artifact intrastructure repository. http://sir.unl.edu/portal/.

[Russell and Norvig, 2003] Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, second edition.

[Schleimer et al., 2003] Schleimer, S., Wilkerson, D. S., and Aiken, A. (2003). Winnowing: local algorithms for document fingerprinting. In *ACM SIGMOD International Conference on Management of Data*, pages 76–85. ACM Press.

[Selman et al., 1995] Selman, B., Kautz, H., and Cohen, B. (1995). Local search strategies for satisfiability testing. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 521–532. American Mathematical Society.

[Settles, 2008] Settles, B. (2008). *Curious Machines: Active Learning with Structured Instance*. PhD thesis, University of Wisconsin–Madison.

[Shavlik and Natarajan, 2009] Shavlik, J. and Natarajan, S. (2009). Speeding up inference in Markov logic networks by preprocessing to reduce the size of the resulting grounded network. In *International Joint Conference on Artificial intelligence*, pages 1951–1956. Morgan Kaufmann.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

[Singla and Domingos, 2008] Singla, P. and Domingos, P. (2008). Lifted first-order belief propagation. In *AAAI Conference on Artificial Intelligence*, pages 1094–1099. AAAI Press.

[Sista et al., 2002] Sista, S., Schwartz, R., Leek, T., , and Makhoul, J. (2002). An algorithm for unsupervised topic discovery from broadcast news stories. In *Human Language Technology Conference*, pages 110–114. Morgan Kaufmann.

[Sontag and Roy, 2009] Sontag, D. and Roy, D. (2009). Complexity of inference in topic models. In *NIPS Workshop on Applications for Topic Models: Text and Beyond*.

[Teh et al., 2006a] Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006a). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

[Teh et al., 2006b] Teh, Y. W., Newman, D., and Welling, M. (2006b). A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1353–1360. MIT Press.

[Thomas et al., 2006] Thomas, M., Pang, B., and Lee, L. (2006). Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Empirical Methods in Natural Language Processing*, pages 327–335. ACL Press.

[Tishby et al., 1999] Tishby, N., Pereira, F. C., and Bialek, W. (1999). The information bottleneck method. In *Allerton Conference on Communication, Control and Computing*, pages 368–377. Curran Associates, Inc.

[Tjong Kim Sang and De Meulder, 2003] Tjong Kim Sang, E. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of Computational Natural Language Learning*, pages 142–147. ACL Press.

[Wagstaff et al., 2001] Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning*, pages 577–584. Morgan Kaufmann.

[Wallach, 2008] Wallach, H. (2008). *Structured Topic Models for Language*. PhD thesis, University of Cambridge.

[Wallach et al., 2009a] Wallach, H., Mimno, D., and McCallum, A. (2009a). Rethinking LDA: Why priors matter. In *Advances in Neural Information Processing Systems*, pages 1973–1981. MIT Press.

[Wallach, 2006] Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In *International Conference on Machine Learning*, pages 977–984. ACM Press.

[Wallach et al., 2009b] Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. (2009b). Evaluation methods for topic models. In *International Conference on Machine Learning*, pages 1105–1112. ACM Press.

[Wang et al., 2009] Wang, C., Blei, D., and Fei-Fei, L. (2009). Simultaneous image classification and annotation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1903–1910. IEEE Computer Society.

[Wang et al., 2008] Wang, C., Blei, D., and Heckerman, D. (2008). Continuous time dynamic topic models. In *Uncertainty in Artificial Intelligence*, pages 579–586. AUAI Press.

[Wang and Domingos, 2008] Wang, J. and Domingos, P. (2008). Hybrid Markov logic networks. In *AAAI Conference on Artificial Intelligence*, pages 1106–1111. AAAI Press.

[Wang and Grimson, 2008] Wang, X. and Grimson, E. (2008). Spatial latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1577–1584. MIT Press.

[Wang and Mccallum, 2005] Wang, X. and Mccallum, A. (2005). A note on topical n-grams. Technical report, University of Massachusetts.

[Wang and McCallum, 2006] Wang, X. and McCallum, A. (2006). Topics over time: a non-Markov continuous-time model of topical trends. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 424–433. ACM Press.

[Wang et al., 2007] Wang, Y., Sabzmeydani, P., and Mori, G. (2007). Semi-latent Dirichlet allocation: A hierarchical model for human action recognition. In *Workshop on Human Motion at the International Conference on Computer Vision*, pages 240–254. Springer.

[Xing et al., 2002] Xing, E., Ng, A., Jordan, M., and Russell, S. (2002). Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, pages 505–512. MIT Press.

[Xing et al., 2005] Xing, E. P., Yan, R., and Hauptmann, A. G. (2005). Mining associated text and images with dual-wing harmoniums. In *Uncertainty in Artificial Intelligence*, pages 633–641. AUAI Press.

[Zheng et al., 2006] Zheng, A. X., Jordan, M. I., Liblit, B., Naik, M., and Aiken, A. (2006). Statistical debugging: Simultaneous identification of multiple bugs. In *International Conference on Machine Learning*, pages 1105–1112. Omnipress.

[Zhu et al., 2009] Zhu, J., Ahmed, A., and Xing, E. P. (2009). MedLDA: maximum margin supervised topic models for regression and classification. In *International Conference on Machine Learning*, pages 1257–1264. Omnipress.

Appendix A: Collapsed Gibbs sampling derivation for ΔLDA

This appendix describes the derivation of the collapsed Gibbs sampling equations for the ΔLDA model. We proceed along similar lines to collapsed Gibbs sampling for standard LDA, noting important points at which the two models differ.

In order to use this model, we must be able to do inference to calculate the posterior $P(\mathbf{z}|\mathbf{w})$

$$P(\mathbf{z}|\mathbf{w}) = \frac{P(\mathbf{w}, \mathbf{z})}{\sum_{\mathbf{z}} P(\mathbf{w}, \mathbf{z})}. \tag{A.1}$$

Here, and for the rest of this derivation, we are assuming that all probabilities are implicitly conditioned on the model hyperparameters (α, β) as well as the observed document success or failure labels $\mathbf{o}$.

Unfortunately the sum in the denominator is intractable (for corpus length N and T topics we have N^T possible $\mathbf{z}$), but we can approximate this posterior distribution with Gibbs sampling. This involves making draws from $P(z_i = j|\mathbf{z}_{-i}, \mathbf{w})$ for each value of i in sequence to generate samples from $P(\mathbf{z}|\mathbf{w})$. This Gibbs sampling equation can be derived as follows

$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}) = \frac{P(z_i = j, \mathbf{z}_{-i}, \mathbf{w})}{\sum_k P(z_i = k, \mathbf{z}_{-i}, \mathbf{w})}. \tag{A.2}$$

This requires computing the full joint for a given value of $\mathbf{z}$ and the observed $\mathbf{w}$. The full joint $P(\mathbf{z}, \mathbf{w})$ can be expressed as

$$P(\mathbf{w}, \mathbf{z}) = P(\mathbf{w}|\mathbf{z})P(\mathbf{z}). \tag{A.3}$$

Substituting in the multinomials and their the Dirichlet priors gives us

$$P(\mathbf{w}|\mathbf{z}) = \prod_{i \in T} \int P(\phi_i|\beta) \prod_{j \in W} P(w_j|\phi_i)^{n_i^j} \, d\phi_i \tag{A.4}$$

$$P(\mathbf{z}) = \prod_{d \in D} \int P(\theta^{(d)}|\alpha) \prod_{j \in T} P(z_j|\theta^{(d)})^{n_j^d} \, d\theta^{(d)} \tag{A.5}$$

where n values are counts derived from the given vectors. n_i^j is the number of times word i is assigned to topic j, and n_j^d is the number of times topic j occurs in document d.

However recall that for ΔLDA, different documents use different α hyperparameters depending on the observed success or failure variable $o \in \{s, f\}$. Let T_f be the set of topics available for the failing runs (i.e., all buggy and all usage topics), while T_s is the set of topics available for the succeeding runs (i.e., usage topics only). Likewise, the set of succeeding documents ($o_d = s$) is referred to as D_s and the set of failing documents ($o_d = f$) are called D_f. Since the θ across documents are conditionally independent of one another given α, we can rewrite our equation as

$$P(\mathbf{z}) = \prod_{o \in \{s,f\}} \left[\prod_{d \in D_o} \int P(\theta|\alpha_o) \prod_{j \in T_o} P(z_j|\theta)^{n_j^d} \, d\theta \right]. \tag{A.6}$$

Dirichlet-multinomial conjugacy allows us to integrate out[1] the Dirichlet priors for each term. This operation results in a distribution known as the multivariate Pólya distribution

$$P(\mathbf{w}|\mathbf{z}) = \prod_{i \in T_f} \left[\frac{\Gamma(|W|\beta)}{\Gamma(|W|\beta + n_i^*)} \prod_{j \in W} \frac{\Gamma(n_i^j + \beta)}{\Gamma(\beta)} \right], \tag{A.7}$$

$$P(\mathbf{z}) = \prod_{o \in \{s,f\}} \left[\prod_{d \in D_o} \frac{\Gamma(\sum_k^{T_o} \alpha_{ok})}{\Gamma(\sum_k^{T_o} \alpha_{ok} + n_*^d)} \prod_{i \in T_o} \frac{\Gamma(n_i^d + \alpha_{oi})}{\Gamma(\alpha_{oi})} \right]. \tag{A.8}$$

In the above $*$ serves as a wild card, meaning that n_i^* is the count of all words assigned to topic i and n_*^d is the count of all words contained in document d. We then re-arrange the equations

$$P(\mathbf{z}) = \prod_{o \in \{b,g\}} \left[\left(\frac{\Gamma(\sum_k^{T_o} \alpha_{ok})}{\prod_{k \in T_o} \Gamma(\alpha_{ok})} \right)^{|D_o|} \prod_d^{D_o} \frac{\prod_i^{T_o} \Gamma(n_i^d + \alpha_{oi})}{\Gamma(\sum_k^{T_o} \alpha_{ok} + n_*^d)} \right], \tag{A.9}$$

$$P(\mathbf{w}|\mathbf{z}) = \left(\frac{\Gamma(|W|\beta)}{\Gamma(\beta)^{|W|}} \right)^{|T_f|} \prod_{i \in T_f} \frac{\prod_j^{|W|} \Gamma(n_i^j + \beta)}{\Gamma(|W|\beta + n_i^*)}. \tag{A.10}$$

Now we need to further modify these equations to account for "pulling out" one specific word-topic pair in order to calculate $P(z_i = j, \mathbf{z}_{-i}, \mathbf{w})$. It is useful to consider which terms will be

[1] This is the "collapsed" aspect of collapsed Gibbs sampling.

unchanged by the assignment to z_i, because those can be pulled out of the sum in the denominator, and will cancel with the numerator. First we break up each of the two components of the equation into two convenient parts: the part dealing with word i, and everything else.

$$P(z_i = j, \mathbf{z}_{-i}, \mathbf{w}) = \prod_{o \in \{s,f\}} \left[\left(\frac{\Gamma(\sum_k^{T_o} \alpha_{ok})}{\prod_k^{T_o} \Gamma(\alpha_{ok})} \right)^{|D_o|} \left(\prod_{d \in D_o \setminus d_i} \frac{\prod_t^{T_o} \Gamma(n_{-i,t}^d + \alpha_{ot})}{\Gamma(\sum_k^{T_o} \alpha_{ok} + n_{-i,*}^d)} \right) \right] \tag{A.11}$$

$$\left(\left[\frac{\prod_{t \neq j}^{T_{o_{d_i}}} \Gamma(n_{-i,t}^{d_i} + \alpha_{o_{d_i} t})}{\Gamma(\sum_k^{T_{o_{d_i}}} \alpha_{o_{d_i} k} + n_{-i,*}^{d_i} + 1)} \right] \Gamma(n_{-i,j}^{d_i} + 1 + \alpha_{o_{d_i} i}) \right) \tag{A.12}$$

$$\left(\frac{\Gamma(|W|\beta)}{\Gamma(\beta)^{|W|}} \right)^{|T_f|} \left(\prod_{t \neq j}^{T_f} \frac{\prod_w^{|W|} \Gamma(n_{-i,t}^w + \beta)}{\Gamma(|W|\beta + n_{-i,t}^*)} \prod_{w \in W} \Gamma(n_{-i,t}^w + \beta) \right) \tag{A.13}$$

$$\left(\frac{\Gamma(n_{-i,j}^{w_i} + 1 + \beta)}{\Gamma(n_{-i,j}^* + 1 + |W|\beta)} \prod_{w \neq w_i}^{W} \Gamma(n_{-i,t}^w + \beta) \right) \tag{A.14}$$

Here d_i refers to the document containing word i, and T_i and $\alpha_{o_{d_i} i}$ refer to the topic set and Dirichlet hyperparameters associated with that document (depending on the value of o_{d_i} for that document). All n counts with the subscript $-i$ are taken for the rest of the sequence (omitting i).

Next, we use the fact that $\Gamma(n) = (n-1)\Gamma(n-1)$ to push the Γ terms containing $+1$ back into the "everything else" products. This product is then insensitive to the j value assigned to z_i, allowing them to cancel out. This re-arrangement and cancellation leaves us with the following equation

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \left(\frac{n_{-i,j}^{w_i} + \beta}{n_{-i,j}^* + \beta|W|} \right) \left(\frac{n_{-i,j}^d + \alpha_{o_{d_i} j}}{n_{-i,*}^d + \sum_k^{T_{o_{d_i}}} \alpha_{o_{d_i} k}} \right). \tag{A.15}$$

The above expression is then evaluated for every possible value of j to get the normalizing factor. Note that for topics j such that $\alpha_{o_{d_i} j} = 0$ the count $n_{-i,j}^d$ should also be 0, meaning that that topic will *never* be assigned to this word. This equation then allows us to do collapsed Gibbs sampling using easily obtainable count values.

This equation is quite similar to the collapsed Gibbs sampling equation for standard LDA, except that when sampling position i we use the value of α dictated by the document outcome

label o_{d_i}. This additional flexibility is what enables us to encode domain knowledge into the α_o vectors, as in ΔLDA (Chapter 4). Furthermore, in this derivation we o can take on two different values, but allowing arbitrarily many values does not affect the end result. For example, we could say that $o \in \{s, f, c\}$ indicates success, failure (bad output), or crash (program termination).

For topic-specific β, the derivation would proceed along very similar lines. This would result in the specific β_{w_i} in the numerator and the sum over all β_k in the denominator, and could be used to encode topic-word domain knowledge (as is done in the Concept-Topic model).

Appendix B: Collapsed Gibbs sampling for Dirichlet Forest LDA

This appendix contains the derivations for the Collapsed Gibbs Sampling equations for LDA with Dirichlet Forest priors. We assume that all Must-Links and Cannot-Links have been provided, the corresponding graphs have been constructed, and the maximal cliques of the complements of the Cannot-Link connected components have been found, as described in Chapter 6.

Our sampling procedure consists of:

1. For each word w_i in the corpus

   Sample $z_i \sim P(z_i|\mathbf{z}_{-i}, \mathbf{q}, \mathbf{w})$

2. For each topic $u = 0, ..., T-1$

   For each Cannot-Link graph connected components $r = 0, ..., R-1$

   Sample $q_u^{(r)} \sim P(q_u^{(r)}|\mathbf{z}, \mathbf{q}_{-u}^{(-v)}, \mathbf{w})$

B.1 Sampling z

First, we show how to sample each z_i value. The necessary equation we wish to derive is

$$P(z_i = v|\mathbf{z}_{-i}, \mathbf{q}, \mathbf{w}) = \frac{P(z_i = v, \mathbf{z}_{-i}, \mathbf{q}, \mathbf{w})}{\sum_k P(z_i = k, \mathbf{z}_{-i}, \mathbf{q}, \mathbf{w})}. \tag{B.1}$$

Since the numerator on the right-hand side is a full joint, we begin with simplifying the terms from (C.1).

The standard Dirichlet prior $P(\theta|\alpha)$ can be integrated out for each term, resulting in the multivariate Pólya distribution:

$$P(\mathbf{z}) = \prod_d^D \frac{\Gamma(\sum_u^T \alpha_u)}{\Gamma(\sum_u^T (\alpha_u + n_u^{(d)})} \prod_u^T \frac{\Gamma(n_u^{(d)} + \alpha_u)}{\Gamma(\alpha_u)}. \tag{B.2}$$

Given $\mathbf{q}$, we have fully specified Dirichlet Tree priors for each topic multinomial ϕ. These can also be collapsed, resulting in a slightly different equation:

$$P(\mathbf{w}|\mathbf{z},\mathbf{q}) = \prod_{u}^{T} \Omega(\mathbf{q}_u) \tag{B.3}$$

$$\Omega(\mathbf{q}_u) = \prod_{j \in t'(q_u)} \frac{\Gamma(\sum_{k \in s(j)} \gamma_k)}{\Gamma(\sum_{k \in s(j)} (\gamma_k + n_u^{(k)}))} \prod_{k \in s(j)} \frac{\Gamma(\gamma_k + n_u^{(k)})}{\Gamma(\gamma_k)}. \tag{B.4}$$

$$\tag{B.5}$$

The counts mean that $n_u^{(k)}$ is the number of words emitted by topic u which are contained in the subtree rooted at k, or the number of times word k was emitted by topic u, if k is a leaf node. The new notation $t'(q_u)$ means non-terminal nodes in the Dirichlet Tree constructed by vector q_u. This notational shorthand means that within the function $\Omega(\mathbf{q}_u)$, all tree structure and edge values are taken with respect to the tree constructed by the vector q_u.

We first re-arrange (B.2), assuming that our standard Dirichlet priors are the same for all documents.

$$P(\mathbf{z}) = \left(\frac{\Gamma(\sum_u^T \alpha_u)}{\prod_u^T \Gamma(\alpha_u)} \right)^{|D|} \prod_d^D \frac{\prod_u^T \Gamma(n_u^{(d)} + \alpha_u)}{\Gamma(\sum_u^T (\alpha_u + n_u^{(d)}))} \tag{B.6}$$

Now we need to modify these equations to account for "pulling out" one specific word-topic pair in order to calculate $P(z_i = v, \mathbf{z}_{-i}, \mathbf{q}, \mathbf{w})$. It is useful to consider which terms will be unchanged by the assignment to z_i, because in the full conditional expression these can be pulled out of the sum in the denominator, and will cancel with the numerator. First we break up each of the components of this equation into two convenient parts: the part dealing with topic assignment for word i, and everything else.

$$P(z_i = v, \mathbf{z}_{-i}, \mathbf{w}, \mathbf{q}) = \left(\frac{\Gamma(\sum_u^T \alpha_u)}{\prod_u^T \Gamma(\alpha_u)}\right)^D \left(\prod_{d \neq d_i}^D \frac{\prod_u^T \Gamma(n_{-i,u}^{(d)} + \alpha_u)}{\Gamma(\sum_u^T (\alpha_u + n_{-i,u}^{(d)}))}\right) \tag{B.7}$$

$$\left(\left[\frac{\prod_{u \neq v}^T \Gamma(n_{-i,u}^{(d_i)} + \alpha_u)}{\Gamma(\sum_u^T (\alpha_u + n_{-i,u}^{(d_i)}) + 1)}\right] \Gamma(n_{-i,v}^{(d_i)} + 1 + \alpha_v)\right) \tag{B.8}$$

$$\left(\prod_{u \neq v}^T \Omega(\mathbf{q}_u)\right) \Omega(\mathbf{q}_v, -i) \tag{B.9}$$

$$\prod_u^T \prod_r^R \frac{|M_{rq_u^{(r)}}|}{\sum_{q'}^{Q(r)} |M_{rq'}|} \tag{B.10}$$

$$\tag{B.11}$$

Here, $\Omega(\mathbf{q}_v, -i)$ is defined as

$$\Omega(\mathbf{q}_v, -i) = \left(\prod_{j \in t'(y_v) \setminus a(w_i)} \frac{\Gamma(\sum_{k \in s(j)} \gamma_k)}{\Gamma(\sum_{k \in s(j)} (\gamma_k + n_{-i,u}^{(k)}))} \prod_{k \in s(j)} \frac{\Gamma(\gamma_k + n_{-i,u}^{(k)})}{\Gamma(\gamma_k)}\right) \tag{B.12}$$

$$\left(\prod_{j \in a(w_i)} \frac{\Gamma(\sum_{k \in s(j)} \gamma_k)}{\Gamma(\sum_{k \in s(j)} (\gamma_k + n_{-i,u}^{(k)} + 1))} \prod_{k \in s(j)} \frac{\Gamma(\gamma_k + n_{-i,u}^{(k)} + 1)}{\Gamma(\gamma_k)}\right). \tag{B.13}$$

$$\tag{B.14}$$

Here all n counts with the subscript $-i$ are taken for the rest of the sequence (omitting i). We also introduce the notation $a(w_i)$, which is the set of all interior nodes which are ancestors of w_i. We also define $a(w_i, j)$ to be the sole element in $a(w_i) \cap s(j)$, which is the child of node j which is also an ancestor of w_i. The tree structure guarantees that if $j \in a(w_i)$ then it must be true that $s(j)$ contains *exactly* one ancestor of w_i (or w_i itself). Therefore $a(w_i, j)$ is well defined for $j \in a(w_i)$.

Next, we use the identity $\Gamma(n) = (n-1)\Gamma(n-1)$ to push the Γ terms containing $+1$ back into the "everything else" products. Note that these products are then insensitive to the v value assigned to z_i, allowing them to cancel out. This subsequent re-arrangement and cancellation leaves us with the following equation:

$$P(z_i = v|\mathbf{z}_{-i}, \mathbf{w}, \mathbf{q}) \propto \left(\frac{n_{-i,v}^{(d)} + \alpha_v}{\sum_u^T (n_{-i,u}^{(d)} + \alpha_u)} \right) \left(\prod_{j \in a(w_i)} \frac{\gamma_{a(w_i,j)} + n_{-i,v}^{a(w_i,j)}}{\sum_{k \in s(j)} (\gamma_k + n_{-i,v}^{(k)})} \right). \tag{B.15}$$

Note that the $a(\cdot)$ terms in this expression are implicitly taken with respect to the Dirichlet Tree constructed by selection vector q_v.

B.2 Sampling $\mathbf{q}$

In order to sample from $P(q_v^{(c)}|\mathbf{q}_v^{(-c)}, \mathbf{q}_{-v}, \mathbf{z}, \mathbf{w})$ we need to again break apart our joint $P(\mathbf{w}, \mathbf{q}, \mathbf{z})$ into two parts. As before, we will facilitate cancellations by ensuring that these two parts correspond to terms affected by the value of $q_v^{(c)}$, and everything else.

$$P(q_v^{(c)} = m|\mathbf{q}_{-v}, \mathbf{q}_v^{(-c)}, \mathbf{w}, \mathbf{z}) = \frac{P(q_v^{(c)} = m, \mathbf{q}_v^{(-c)}, \mathbf{q}_{-v}, \mathbf{w}, \mathbf{z})}{\sum_a P(q_v^{(c)} = a, \mathbf{q}_v^{(-c)}, \mathbf{q}_{-v}, \mathbf{w}, \mathbf{z})} \tag{B.16}$$

Decomposing the full joint $P(\mathbf{q}, \mathbf{w}, \mathbf{z})$ as before, we get

$$P(q_v^{(c)} = m, \mathbf{q}_v^{(-c)}, \mathbf{q}_{-v}, \mathbf{w}, \mathbf{z}) = \left(\frac{\Gamma(\sum_u^T \alpha_u)}{\prod_u^T \Gamma(\alpha_u)} \right)^{|D|} \left(\prod_d^D \frac{\prod_u^T \Gamma(n_u^{(d)} + \alpha_u)}{\Gamma(\sum_u^T (\alpha_u + n_u^{(d)}))} \right) \tag{B.17}$$

$$\left(\prod_{u \neq v}^T \Omega(\mathbf{q}_u) \right) \Omega(q_v^{(c)} = m, \mathbf{q}_v^{(-c)}) \tag{B.18}$$

$$\prod_u^T \prod_r^R \frac{|M_{rq_u^{(r)}}|}{\sum_{q'}^{Q(r)} |M_{rq'}|}. \tag{B.19}$$

$$\tag{B.20}$$

Here we introduce the new function $\Omega(q_v^{(c)} = m, \mathbf{q}_v^{(-c)})$, defined as:

$$\Omega(q_v^{(c)} = m, \mathbf{q}_v^{(-c)}) = \left(\prod_{j \in t'(\mathbf{q}_v^{(-c)})} \frac{\Gamma(\sum_{k \in s(j)} \gamma_k)}{\Gamma(\sum_{k \in s(j)} (\gamma_k + n_v^{(k)}))} \prod_{k \in s(j)} \frac{\Gamma(\gamma_k + n_v^{(k)})}{\Gamma(\gamma_k)} \right) \tag{B.21}$$

$$\left(\prod_{j \in t'(q_v^{(c)})} \frac{\Gamma(\sum_{k \in s(j)} \gamma_k)}{\Gamma(\sum_{k \in s(j)} (\gamma_k + n_v^{(k)}))} \prod_{k \in s(j)} \frac{\Gamma(\gamma_k + n_v^{(k)})}{\Gamma(\gamma_k)} \right). \tag{B.22}$$

$$\tag{B.23}$$

Here $t'(\mathbf{q}_v^{(-c)})$ refers the set of all non-terminals in the Dirichlet Tree defined by $\mathbf{q}_v$, *except* those contained in the subtree determined by element $q_v^{(c)}$. We then define $t'(q_v^{(c)})$ to be the set of non-terminals in the Dirichlet Tree contained in the subtree determined by element $q^{(c)}$. Since our construction procedure ensures that the different agreeable subtrees are always disjoint, this is always a valid decomposition.

We then observe that the product over $t'(q_v^{(c)})$ in $\Omega(q_v^{(c)} = m, \mathbf{q}_v^{(-c)})$ and the $|M_{cq_v^{(c)}}|$ term are the only terms in (B.20) which will be affected by the value of $q_v^{(c)}$, meaning that everything else will cancel out. After these cancellations, our final sampling equation for each element of $\mathbf{q}$ is then given by

$$P(q_v^{(c)} = m | \mathbf{q}_v^{(-c)}, \mathbf{q}_{-v}, \mathbf{z}, \mathbf{w}) \propto |M_{rm}| \left(\prod_{j \in t'(q_v^{(c)})} \frac{\Gamma(\sum_{k \in s(j)} \gamma_k)}{\Gamma(\sum_{k \in s(j)} (\gamma_k + n_v^{(k)}))} \prod_{k \in s(j)} \frac{\Gamma(\gamma_k + n_v^{(k)})}{\Gamma(\gamma_k)} \right) \tag{B.24}$$

where the edge values and subtree structures are determined by $q_v^{(c)} = m$.

Appendix C: Collapsed Gibbs sampling derivation for LogicLDA

In this appendix, we show the derivation of the collapsed Gibbs sampler for the LogicLDA model. We begin with the full joint distribution, using the function DCM to stand in for the Dirichlet-Compound Multinomials present in the collapsed LDA distribution.

$$DCM_\alpha(\mathbf{w}, \mathbf{z}) = \prod_d^D \frac{\Gamma(\sum_u^T \alpha_u)}{\Gamma(\sum_u^T (\alpha_u + n_u^{(d)})} \prod_u^T \frac{\Gamma(n_u^{(d)} + \alpha_u)}{\Gamma(\alpha_u)} \tag{C.1}$$

$$DCM_\beta(\mathbf{w}, \mathbf{z}) = \prod_u^T \frac{\Gamma(\sum_j^W \beta_u)}{\Gamma(\sum_j^W (\beta_j + n_u^{(j)})} \prod_j^W \frac{\Gamma(n_u^{(j)} + \beta_j)}{\Gamma(\beta_j)} \tag{C.2}$$

$$P(\mathbf{w}, \mathbf{z}|\alpha, \beta, L, \eta) = \frac{1}{Z} DCM_\alpha(\mathbf{w}, \mathbf{z}) DCM_\beta(\mathbf{w}, \mathbf{z}) \exp\left(\sum_\ell^L \eta f_\ell(\mathbf{w}, \mathbf{z})\right) \tag{C.3}$$

$$\tag{C.4}$$

The full conditional probability used for Gibbs sampling is

$$P(z_i = v|\mathbf{z}_{-i}, \mathbf{w}) = \frac{P(z_i = v, \mathbf{z}_{-i}, \mathbf{w})}{\sum_k P(z_i = k, \mathbf{z}_{-i}, \mathbf{w})}. \tag{C.5}$$

The numerator on the right-hand side is the full joint from (C.1). Expanding the DCM terms, we further decompose the equation into two components: one affected by the value of z_i, and one insensitive to it.

$$P(z_i = v|\mathbf{z}_{-i}, \mathbf{w}) = \frac{1}{Z} \left(\frac{\Gamma(\sum_u^T \alpha)}{\prod_u^T \Gamma(\alpha)} \right)^D \left(\prod_{d \neq d_i}^{D} \frac{\prod_u^T \Gamma(n_{-i,u}^{(d)} + \alpha)}{\Gamma(\sum_u^T (\alpha + n_{-i,u}^{(d)}))} \right) \tag{C.6}$$

$$\left(\left[\frac{\prod_{u \neq v}^T \Gamma(n_{-i,u}^{(d_i)} + \alpha)}{\Gamma(\sum_u^T (\alpha + n_{-i,u}^{(d_i)}) + 1)} \right] \Gamma(n_{-i,v}^{(d_i)} + 1 + \alpha) \right) \tag{C.7}$$

$$\left(\frac{\Gamma(\sum_k^W \beta)}{\prod_k^W \Gamma(\beta)} \right)^T \left(\prod_{u \neq v}^{T} \frac{\prod_k^W \Gamma(n_{-i,u}^{(k)} + \beta)}{\Gamma(\sum_u^T (\beta + n_{-i,u}^{(k)}))} \right) \tag{C.8}$$

$$\left(\left[\frac{\prod_{k \neq w_i}^W \Gamma(n_{-i,v}^{(k)} + \beta)}{\Gamma(\sum_k^W (\beta + n_{-i,v}^{(k)}) + 1)} \right] \Gamma(n_{-i,v}^{(w_i)} + 1 + \beta) \right) \tag{C.9}$$

$$\exp\left(\sum_{\ell}^{L_{-i}} \eta f_\ell(z_i = v, \mathbf{z}_{-i}, \mathbf{w})\right) \tag{C.10}$$

$$\exp\left(\sum_{\ell}^{L_i} \eta f_\ell(z_i = v, \mathbf{z}_{-i}, \mathbf{w})\right) \tag{C.11}$$

$$\tag{C.12}$$

Here L_i are the logical formulas whose value depends in z_i, $L_{-i} = L \setminus L_i$, and the count variables n have the same interpretation as elsewhere in this document. We can see that all of the terms unaffected by z_i will cancel out of (C.5). Applying the identity $\Gamma(n) = (n-1)\Gamma(n-1)$, we are left with the desired Gibbs sampling equation

$$P(z_i = v|\mathbf{z}_{-i}, \mathbf{w}) \propto \left(\frac{n_{-i,v}^{(d)} + \alpha}{\sum_u^T (n_{-i,u}^{(d)} + \alpha)} \right) \left(\frac{n_{-i,v}^{(w_i)} + \beta}{\sum_{w'}^W (n_{-i,v}^{(w')} + \beta)} \right) \exp\left(\sum_{\ell}^{L_i} \eta f_\ell(z_i = v, \mathbf{z}_{-i}, \mathbf{w})\right). \tag{C.13}$$