UNIVERSITY OF CALIFORNIA,
IRVINE

Statistical Models for Text Classification and Clustering: Applications and Analysis

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

America Chambers

Dissertation Committee:
Professor Padhraic Smyth, Chair
Professor Rina Dechter
Professor Mark Steyvers

2013

DEDICATION

To God my father

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor Padhraic Smyth. I have been extremely fortunate to have an advisor who both pushed me and allowed me the space to follow my own research interests. I also want to thank my committee members Mark Steyvers, for his enthusiasm about my research, and Rina Dechter, for all the small ways in which she has shown me kindness over the years.

Thank you to Alex Ihler, Ian Porteous, Arthur Ascuncion and Chaitanya Chemudugunta for helpful discussions about my research. Thanks to Tim Rubin for teaching me how to be a more thorough researcher and thanks to all of my labmates (both past and present) for the fun times. In particular, thanks to Jon Hutchins, Drew Frank, Nick Navaroli, Chris DuBois, and Corey Schaninger for a lot of good memories!

I would also like to thank the agencies that have funded my research: the National Science Foundation (NSF), the Intelligence Advanced Research Projects Activity (IARPA) and Microsoft for the Microsoft Research Graduate Women's Scholarship.

To all of the friends who have prayed for me and encouraged me throughout these past few years, thank you! I want to especially thank Rachel. You walked this road before me and were willing to share your own difficulties (often through absurdly humorous stories) to help me through mine. And to Eunsunk who tells the truth in gentleness and love: seeing you pursue your own dreams has humbled me and encouraged me. Thank you both.

To my mother, the strongest woman in the world, this is truly as much your Ph.D. as my own. And to my older sister who even still continues to take care of me. I love you both!

To (unarguably) the best husband in the world: thank you, thank you, thank you! Words are too coarse to ever capture and express all that you mean to me.

Finally, above everyone else and everything else, I want to acknowledge and give thanks to Jesus Christ (who gives freedom, peace, strength, and purpose to anyone who would ask him) and to my father, God, without whom not a single word in this thesis would exist.

ABSTRACT OF THE DISSERTATION

Statistical Models for Text Classification and Clustering:
Applications and Analysis

By

America Chambers

Doctor of Philosophy in Computer Science

University of California, Irvine, 2013

Professor Padhraic Smyth, Chair

The growth of the internet has sparked an explosion in the rate at which text data is produced and published. From traditional sources such as newspapers, magazines, academic journals, and books to newer sources such as product reviews, emails, blogs, tweets, chat sessions, and tags, text data exists in an overwhelming abundance. As such, it is essential that we (1) develop methods for organizing and managing large collections of text documents as well as (2) develop methods for mining and analyzing individual documents. Statistical models provide a principled and mathematically-sound framework in which to accomplish both of these tasks. In this dissertation, we investigate the usefulness of statistical models of text for both organizing large collections of documents as well as mining individual documents.

First, we present an application of latent Dirichlet allocation [5] (a well-known statistical model of text) to sentence classification in scientific articles where the goal is to classify sentences according to function. Sentence classification provides a deeper understanding of the argumentative structure of a scientific article and is an important element for tasks such as document summarization. Next, we present a flexible non-parametric statistical model based on latent Dirichlet allocation for learning concept graphs from text. Concept graphs are useful for summarizing document collections and providing a visualization of the semantic

content and structure of large document sets – a task that is difficult to accomplish using only keyword search. Finally, in the last chapter, we move from application to analysis. We present a theoretical analysis of the accuracy of the multinomial Dirichlet mixture model when used for text classification. The multinomial Dirichlet mixture model is the starting point for many of the more complex statistical models commonly used for probabilistic text classification.

Chapter 1

Introduction

The growth of the internet has sparked an explosion in the rate at which text data is produced and published. From traditional sources such as newspapers, magazines, academic journals, and books to newer sources such as product reviews, emails, blogs, tweets, chat sessions, and tags, text data exists in an overwhelming abundance. As such, it is essential that we (1) develop methods for organizing and managing large collections of text documents as well as (2) develop methods for mining and analyzing individual documents. Example tasks of the former include document clustering, online clustering, and visualization; examples of the latter include automatic summarization, question answering, and sentiment analysis.

Statistical models provide a principled and mathematically sound framework in which to accomplish both of these tasks. One well-known and widely-used statistical model of text is latent Dirichlet allocation [5]. Latent Dirichlet allocation is a latent variable mixture model where a document is modeled as a mixture over T clusters known as *topics*. Informally, a topic is a semantically-focused set of words. For example, a topic about "football" might include words like "football", "touchdown", "interception", and "quarterback". Formally, a topic is represented as a probability vector over some vocabulary where high probability

is assigned to on-topic words and low (or zero) probability is assigned to all other off-topic words. The set of T topics learned from a collection of documents provides a summary of their topical content. Latent Dirichlet allocation has found wide application in the text modeling, natural language processing, and information retrieval domains – e.g. word sense disambiguation [7], document classification [55, 61], phrase detection [37, 51], collaborative filtering [75], and the visualization and analysis of large document collections [3, 50, 4]. In this dissertation, we investigate the usefulness of statistical models of text for both organizing large collections of text documents as well as mining individual text documents.

First, we present an application of latent Dirichlet allocation to sentence classification in scientific articles where the goal is to classify sentences according to function. Within the domain of scientific articles, typical sentence functions include AIM (the sentence states the aim of the article), RESULTS (the sentence states a result of the investigation presented in the article), HYPOTHESIS (the sentence states a hypothesis of the authors'), etc. Sentence classification provides a deeper understanding of the argumentative structure of a scientific article and is an important element for tasks such as automatic summarization [69, 24].

Next, we present a flexible non-parametric statistical model based on latent Dirichlet allocation for learning concept graphs from text. Concept graphs are rooted acyclic graph structures where the nodes represent concepts – i.e. topics – and the edges represent relationships between concepts. Concept graphs are useful for summarizing document collections and providing a visualization of the semantic content and structure of large document sets [3, 50, 4] – a task that is difficult to accomplish using only keyword search.

Finally, in the last chapter, we move from application to analysis. We present a theoretical analysis of the accuracy of the multinomial Dirichlet mixture model when used for text classification. We derive a classification rule using this mixture model for two scenarios where we have access to differing amounts of information. In the first scenario, we observe the parameters of the class multinomial likelihood functions. In the second scenario, we

observe only a set of representative documents from each class. In both cases, we derive a closed-form approximation for the average error rate of the classifier by appealing to a central limit theorem for multinomial sums. We then establish the relationship between the average error rate (given by the approximation) and certain quantities of interest – e.g. the document length – providing insight into a statistical model that is the starting point for many of the models used for probabilistic text classification.

1.1 Major contributions

The major contributions of this dissertation are as follows:

- (Chapter 3) Sentence classification in scientific articles
  - We present two new statistical models, SentenceLDA and Multicorpus SentenceLDA, for sentence classification in scientific articles where the goal is to classify sentences according to function. Within the domain of scientific articles, typical sentence functions include AIM, RESULTS, HYPOTHESIS, etc.
    - SentenceLDA adapts and extends latent Dirichlet allocation to perform sentence classification. One of the key characteristics of SentenceLDA is its generalization of a "document" to "groups" of sentences that exhibit similar distributions over sentence functions.
    - Multicorpus SentenceLDA extends SentenceLDA by incorporating a second mechanism to explain the presence of domain-dependent words in a sentence.
  - We create a data set of sentences from scientific articles that span three different domains: computational biology, machine learning, and psychology. The sentences in the abstract and introduction of each article have been labeled with sentence functions derived from the Argumentative Zones annotation scheme [69].

- We discuss a method for mining this labeled data to create informative priors and have included in Appendix C a complete list of indicator words for each label in our annotation scheme.
- We use this labeled data to evaluate the performance of SentenceLDA and Multicorpus SentenceLDA on a set of test articles and report performance in terms of the F_1 score (a commonly used measure in text classification). We also compare the performance of SentenceLDA and Multicorpus SentenceLDA to five other supervised and semi-supervised classifiers: Dirichlet multinomial regression [44], support vector machines, two naive Bayes classifiers, and a transductive SVM [13].
- We analyze the performance of each classifier for a variety of tasks including label-pivoted binary predictions, label-pivoted rankings, and document-pivoted binary predictions [62, 40]. Both SentenceLDA and Multicorpus SentenceLDA are competitive with, or outperform, the baseline classifiers.

- (Chapter 4) Summarizing document collections using concept graphs
  - We present a flexible non-parametric prior for rooted, directed, acyclic graphs with a possibly infinite number of nodes. We construct this prior by specifying a stick-breaking distribution at each node that governs the probability of transitioning from the given node to another node in the graph.
  - We combine this prior over graphs with latent Dirichlet allocation to create a new generative model called GraphLDA for learning concept graphs from text. A concept graph is a rooted, directed graph whose nodes represent concepts – i.e. semantically-related collections of words – and edges represent relationships between concepts. Concept graphs provide a useful summary and visualization of the semantic content and structure of document sets.
  - We show how GraphLDA can be used to learn a concept graph from a collection of documents or can be used to update an existing graph structure in the presence

of new labeled documents.

  - We illustrate the performance of GraphLDA on a set of simulated documents where we increase the proportion of labeled documents used for training.
  - We compare the performance of GraphLDA to the hierarchical Pachinko allocation model (hPAM) and hierarchical latent Dirichlet allocation (hLDA) using both the empirical likelihood algorithm and the left-to-right algorithm [74] for computing the per-word log likelihood on a hold-out set of documents.
  - We illustrate an application of GraphLDA to Wikipedia's category graph. We show how GraphLDA can be used to update a portion of the Wikipedia category graph rooted at the node MACHINE LEARNING given a collection of machine learning abstracts.

- (Chapter 5) An analysis of the multinomial Dirichlet mixture model for text classification

  - We present an analysis of the accuracy of the multinomial Dirichlet mixture model when used for text classification. The multinomial Dirichlet mixture model is the starting point for many of the more statistical models used for probabilistic text classification.
  - We consider two scenarios in which different information is available. In the first scenario, we observe the parameters of the class multinomial likelihood functions. In the second scenario, we observe only a set of representative documents from each class. In both cases, we analyze the relationship between the error rate of the classifier and certain quantities of interest.
  - Contributions of the first scenario
    * We present an expression for the Bayes error of the log likelihood ratio test and we approximate this expression using Monte Carlo simulations.

    * We present a closed-form approximation to the Bayes error rate by appealing to a central limit theorem for multinomial sums [46].
    * We establish the relationship between the Bayes error rate (computed using our Normal approximation) and the similarity of the class multinomial parameters (as measured by the Jeffrey's divergence), the Dirichlet hyper-parameter, the document length, and the vocabulary size.

- Contributions of the second scenario
    * We derive a fully Bayesian classification rule using the ratio of the *marginal* likelihoods.
    * We present an interpretation of the classification rule that elucidates how evidence is accumulated in favor of both classes.
    * We derive an expression for the average error rate of this classifier.
    * We derive a closed-form approximation to the average error rate by appealing to the same central limit theorem.
    * We establish the relationship between the average error rate (computed using our Normal approximation) and the similarity of the class multinomial parameters (as measured by the Jeffrey's divergence), the Dirichlet hyper-parameter, the document length, and the vocabulary size.

Chapter 2

Background

In this chapter we give a brief introduction to the multinomial, Dirichlet, and Dirichlet compound multinomial distributions. These distributions are commonly used when constructing statistical models of text. We also give a more in-depth introduction to latent Dirichlet allocation including a derivation of the collapsed Gibbs sampling equations. All of these results are standard and are known in the literature [5, 23] and are included here for completeness.

2.1 Probability distributions for text modeling

2.1.1 The categorical and multinomial distributions

Let x be a discrete random variable taking on values in $\{1, \ldots, K\}$. The probability distribution of x can be parameterized by the random vector $p = (p_1, \ldots, p_K)$ where $p(x = i) = p_i$, $0 \leq p_i \leq 1$, and $\sum_i p_i = 1$.

If we have N such independent and identically distributed (iid) random variables $(x_1, \ldots, x_N)$, where $x_j \in \{1, \ldots, K\}$, then their joint probability distribution is also parameterized by p

and is given by

$$
\begin{aligned}
p(x_1, \ldots, x_N) &= \prod_{j=1}^{N} p_1^{\mathrm{I}(x_j=1)} \ldots p_K^{\mathrm{I}(x_j=K)} \\
&= \prod_{i=1}^{K} p_i^{\sum_{j=1}^{N} \mathrm{I}(x_j=i)} \\
&= \prod_{i=1}^{K} p_i^{n_i}
\end{aligned}
\tag{2.1}
$$

where $\mathrm{I}(x_j = i)$ is an indicator function that is 1 if x_j takes on the value i and is 0 otherwise, and $n_i = \sum_{j=1}^{N} \mathrm{I}(x_j = i)$ counts the number of random variables that take on the value i. Note that $\sum_i n_i = N$. Equation 2.1 gives the probability mass function of the *categorical* distribution.

One property of the categorical distribution is that any permutation of the random variables $(x_{\sigma(1)}, \ldots, x_{\sigma(N)})$ has the same probability[1]. The joint probability distribution is independent of how the random variables are ordered[2] and depends only on the count statistics $(n_1, \ldots, n_K)$. Any two sets of random variables, x and x_σ, whose outcomes have the same count statistics have the same probability. Note that there are $\binom{N}{n_1 \ldots n_K}$ such possible sets of random variables with the count statistics $(n_1, \ldots, n_K)$.

As such, we can instead define a probability distribution over the vector of counts $(n_1, \ldots, n_K)$. This is known as the *multinomial distribution*. The probability of the vector $(n_1, \ldots, n_K)$ is

[1] where σ is a permutation of the integers $\{1, \ldots, N\}$

[2] a property known as *exchangeability*

given by,

$$p(n_1, \ldots, n_K) = \binom{N}{n_1 \ldots n_K} \prod_{i=1}^{K} p_i^{n_i} \tag{2.2}$$

Note that the only difference between Equation 2.1 and Equation 2.2 is the presence of the multinomial coefficient $\binom{N}{n_1 \ldots n_K}$.

The K values $\{1, \ldots, K\}$ are often called the *outcomes* and the parameter N is the number of *trials*. Each trial results in a single outcome. In the text modeling literature, the outcomes are associated with words in some vocabulary and each trial is analogous to rolling a K-sided die where each side of the die corresponds to a word (see [40] chapter 12). The probability vector $p = (p_1, \ldots, p_K)$ specifies the probability of rolling each word in the vocabulary. To generate a document of length N, we roll the K-sided die N times. Thus, a document can be viewed as a sample from a multinomial (or categorical) distribution with random vector p and number of trials N.

2.1.2 The Dirichlet distribution

The Dirichlet distribution is a distribution over probabililty vectors $p = (p_1, \ldots, p_K)$ where $0 \leq p_i \leq 1$ and $\sum_i p_i = 1$. The Dirichlet distribution is parameterized by a vector $(\eta_1, \ldots, \eta_K)$ where $\eta_i \in \mathbb{R}_{>0}$. The probability of p is given by

$$
\begin{aligned}
p(p|\eta) &= \frac{1}{B(\eta_1, \ldots, \eta_K)} \prod_{i=1}^{K} p_i^{\eta_i - 1} \\
&= \frac{\Gamma(\sum_i \eta_i)}{\prod_i \Gamma(\eta_i)} \prod_{i=1}^{K} p_i^{\eta_i - 1}
\end{aligned}
$$

The function $B(\eta_1, \ldots, \eta_K)$ is the multivariate Beta function. The function $\Gamma(\eta_i)$ is the Gamma function. The Gamma function is the extension of the factorial function to the real and complex numbers: $\Gamma(a) = (a-1)!$ for integer a. The Beta and Gamma function are linked by the identity

$$
B(\eta_1, \ldots, \eta_K) = \frac{\prod_i \Gamma(\eta_i)}{\Gamma(\sum_i \eta_i)}
$$

Note the similarities between the normalizing constant of the multinomial distribution (i.e. the multinomial coefficient) and the normalizing constant of the Dirichlet distribution (i.e. $B(\eta_1, \ldots, \eta_K)^{-1}$) when both are expressed in their factorial form.

A special case of the Dirichlet distribution is the *symmetric Dirichlet distribution* where $\eta_i = \eta$ for all i. The scalar parameter η is often called the *concentration* parameter. When $\eta < 1$, the probability mass is concentrated on sparse probability vectors that give low probability to most outcomes and high probability to only a few outcomes. When $\eta = 1$, the probability mass is spread uniformly over all probability vectors. When $\eta > 1$, the probability mass is concentrated on probability vectors with nearly uniform (i.e. equal) probability over all outcomes.

The Dirichlet distribution is conjugate to both the multinomial and the categorical distributions. That is, if $x \sim \text{Mult}(p, N)$ and $p \sim \text{Dirichlet}(\eta_1, \ldots, \eta_K)$ then the posterior

distribution over p is Dirichlet$(\eta_1 + n_1, \ldots, \eta_K + n_K)$. The same is true if x is distributed according to a categorical distribution. For this reason, the Dirichlet distribution is often used as a prior over probability vectors p.

In practice, we may wish to estimate the parameters of the Dirichlet distribution given a set of observed multinomial samples. Minka et al. [45] present two iterative algorithms for computing the maximum likelihood estimates of the Dirichlet parameters.

2.1.3 The Dirichlet compound multinomial distribution

The Dirichlet compound multinomial distribution (DCM), also known as the compound multinomial or the multivariate Polya distribution, is a distribution over discrete random variables x that are drawn from a multinomial (or categorical) distribution where the probability vector p is unknown but distributed according to a Dirichlet distribution. The probability of a variable x conditioned on the parameters of the Dirichlet distribution $(\eta_1, \ldots, \eta_K)$ is given by

$$\begin{aligned}
p(x|\eta_1, \ldots, \eta_K) &= \int_p p(x|p) \cdot p(p|\eta_1, \ldots, \eta_K)\, dp \\
&= \int_p \binom{N}{n_1 \ldots n_K} \prod_{i=1}^{K} p_i^{n_i} \cdot \frac{1}{B(\eta_1, \ldots, \eta_K)} \prod_{i=1}^{K} p_i^{\eta_i - 1}\, dp \\
&= \binom{N}{n_1 \ldots n_K} \frac{1}{B(\eta_1, \ldots, \eta_K)} \int_p \prod_{i=1}^{K} p_i^{\eta_i + n_i - 1}\, dp \\
&= \binom{N}{n_1 \ldots n_K} \frac{B(\eta_1 + n_1, \ldots, \eta_K + n_K)}{B(\eta_1, \ldots, \eta_K)}
\end{aligned}$$

To compute the probability of x, we must integrate over the unknown p. We recognize the integral in the third line as the density function of an unnormalized Dirichlet random

variable. If we multiply by the correct normalizing constant $B(\eta_1 + n_1, \ldots, \eta_K + n_K)$, the integral evaluates to 1 and we are left with a product of the multinomial coefficient and a ratio of normalizing constants from the posterior Dirichlet and the prior Dirichlet distributions.

Note the same derivation can be done using a categorical instead of a multinomial distribution, in which case, the multinomial coefficient is not present. For more information on the Dirichlet compound multinomial distribution see [47] and [45].

2.2 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA), also known as a topic model, is a state-of-the-art unsupervised learning technique for extracting a set of topics that describe the thematic content of a document collection [5]. Informally, a *topic* is a semantically-focused set of words. For example, a topic about "football" might include words like "football", "quarterback", "touchdown", and "fumble". Or a topic about "traveling" might include words like "passport", "sight-seeing", "airplane", and "tourist". In this section, we present the generative model of LDA along with the derivation of a collapsed Gibbs sampler for inference.

2.2.1 Generative model

Informally, a topic is a semantically-focused set of words. Formally, LDA represents a topic as a probability vector, or distribution, over the words in a vocabulary. Thus, the topic about "football" would give high-probability to words such as "football", "quarterback", "touchdown", etc. and give low (or zero) probability to all other non-football related words. Similarly, the topic about "traveling" would give high-probability to traveling related words, and low (or zero) probability to non-traveling related words. We follow the convention of using ϕ to represent such a probability vector. Topic t is denoted as $\phi_t = (\phi_{t1}, \ldots, \phi_{tW})$,

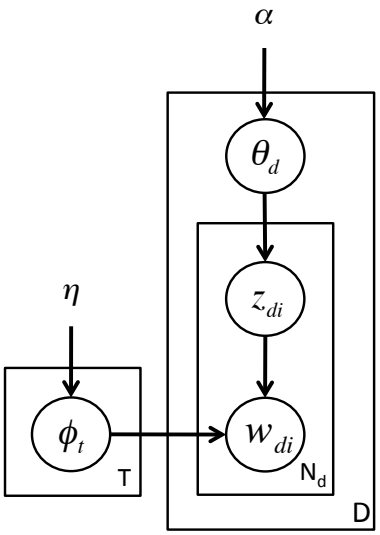

Figure 2.1: The plate notation for latent Dirichlet allocation.

where W is the size of the vocabulary, $0 \leq \phi_{ti} \leq 1$, and $\sum_i \phi_{ti} = 1$. The value ϕ_{ti} is the probability of the ith word in the vocabulary under the t-th topic. We denote all T topics collectively as $\boldsymbol{\phi}$. Note that T is a parameter set by the user. For a non-parameteric extension of LDA, where T is learned from the data see [66].

To generate a document, a distribution over the set of T topics is first sampled. We follow the convention of using θ_d to represent this distribution over topics for the dth document. Then $\theta_d = (\theta_{d1}, \ldots, \theta_{dT})$ where $0 \leq \theta_{dt} \leq 1$, and $\sum_t \theta_{dt} = 1$. The value θ_{dt} is the probability of the t-th topic under the dth document. We denote all D distributions collectively as $\boldsymbol{\theta}$ where D is the total number of documents.

To generate each word in the document, a topic is first sampled from θ_d and then a word is sampled from the corresponding topic. This process is repeated for each word in the document. As an example, a document about drug usage in professional sports might give high-probability to the "football" topic, and a "baseball" topic, as well as a "drug" topic, and give low (or zero) probability to the remaining topics. To generate a word, we would first sample a topic. If we sampled the "baseball" topic, we would look up the corresponding

probability vector and generate a word, e.g. "pitcher."

This process describes a hypothetical statistical procedure for generating a collection of documents. Such a procedure is called a *generative model*. The full generative model of LDA is shown below:

1. For topic $t \in \{1, \ldots, T\}$
   (a) Sample a distribution over words $\phi_t \sim \text{Dirichlet}(\eta_1, \ldots, \eta_W)$
2. For document $d \in \{1, \ldots, D\}$
   (a) Sample a distribution over topics $\theta_d \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_T)$
   (b) For each word $i \in \{1, \ldots, N_d\}$
      i. Sample a topic $z_{di} \sim \text{Multinomial}(\theta_d, 1)$
      ii. Sample a word $w_{di} \sim \text{Multinomial}(\phi_{z_{di}}, 1)$

N_d is the number of words in document d, $w_{di} \in \{1, \ldots, W\}$ is the ith word in the dth document, and $z_{di} \in \{1, \ldots, T\}$ is its topic assignment. Figure 2.1 shows the plate notation for latent Dirichlet allocation.

2.2.2 Inference

The generative model for latent Dirichlet allocation gives rise to the following joint distribution

$$p(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\phi} | \alpha, \eta) = \prod_d \prod_i p(w_{di} | \phi_{z_{di}}) \prod_d \prod_i p(z_{di} | \theta_d) \prod_d p(\theta_d | \alpha) \prod_t p(\phi_t | \eta) \tag{2.3}$$

The vocabulary is usually taken to be the union of all the unique words in the corpus. It is common to remove infrequently occurring words, e.g. all words that occur less than 10 times, as well as common words (often called *stopwords*), e.g. "the", "a", "and." The word variables $\mathbf{w} = \{w_{di} : 1 \leq d \leq D, 1 \leq i \leq N_d\}$ are known and observed. What is not known are the topic assignment variables $\mathbf{z} = \{z_{di}\}$, the document distributions $\boldsymbol{\theta} = \{\theta_d\}$, and the topic distributions $\boldsymbol{\phi} = \{\phi_t\}$. Thus, we are interested in computing the posterior distribution $p(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\phi} | \mathbf{w}, \alpha, \eta)$.

Given the posterior distribution, one option is to compute a point estimate of the random variables $\{\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\phi}\}$, e.g. a posterior mode. However, a Bayesian perspective would seek to make use of the entire posterior distribution (thus reflecting our uncertainty about the true value of the random variables). In this case, we would draw multiple samples from the posterior distribution.

Markov Chain Monte Carlo (MCMC) is a class of algorithms for obtaining samples from a distribution [21]. MCMC algorithms construct a Markov chain whose stationary distribution is the distribution of interest – in our case, the posterior distribution $p(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\phi} | \mathbf{w}, \alpha, \eta)$. A state of the Markov chain corresponds to a complete instantiation of all the random variables. Gibbs sampling, a well-known MCMC algorithm, transitions from one state to another by sampling (updating) each random variable from the conditional distribution of that variable given the value of all the other variables. After a large number of transitions, we are guaranteed that the probability of being in any state is equal to that state's probability (i.e. equal to the probability of the instantiation of the random variables corresponding to the state). See Gelman et al. [21] for a thorough treatment of MCMC techniques.

If we want to use the Gibbs sampling algorithm to obtain samples from our posterior distribution, then we must be able to sample from the following conditional distributions: $p(z_{di} | z_{-di}, \mathbf{w}, \boldsymbol{\theta}, \boldsymbol{\phi})$, $p(\theta_d | \mathbf{z}, \mathbf{w}, \boldsymbol{\theta}_{-d}, \boldsymbol{\phi})$, and $p(\phi_t | \mathbf{z}, \mathbf{w}, \boldsymbol{\theta}, \boldsymbol{\phi}_{-t})$ where the subscripts $-di$, $-t$, and $-d$ mean the set of random variables minus the current variable.

For latent Dirichlet allocation, it is common to use a *collapsed* Gibbs sampler in which the variables $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are marginalized (i.e. collapsed) out of the joint distribution [23]. Collapsing out $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ can improve the mixing properties of the Gibbs sampler [49]. For the collapsed Gibbs sampler, the only distribution we need for sampling is the conditional distribution of z_{di} given the value of all other variables, $p(z_{di}|z_{-di}, \mathbf{w}, \alpha, \eta)$. Note that in this case we obtain posterior samples of $\mathbf{z}$ from which we can compute point estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.

In the remainder of this section, we present the full derivation of the sampling equation $p(z_{di}|z_{-di}, \mathbf{w}, \alpha, \eta)$. This derivation can be safely skipped by the reader if desired.

Equation 2.4 shows the sampling equation $p(z_{di}|z_{-di}, \mathbf{w}, \alpha, \eta)$:

$$\begin{aligned} p(z_{di} = j|\mathbf{z}_{-di}, \mathbf{w}, \alpha, \eta) &= \frac{p(w_{di}, z_{di} = j|\mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta)}{p(w_{di}|\mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta)} \\ &\propto p(w_{di}, z_{di} = j|\mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta) \\ &= p(w_{di}|z_{di} = j, \mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta) \cdot p(z_{di}|\mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta) \\ &= p(w_{di}|z_{di} = j, \mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta) \cdot p(z_{di}|\mathbf{z}_{-di}, \alpha, \eta) \end{aligned} \tag{2.4}$$

The first line in Equation 2.4 is an application of Bayes rule. In the second line, we note that the probability $p(w_{di}|\mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta)$ does not depend upon the value of z_{di} and as such it is a constant that can be safely dropped. The third line is an application of the chain rule. Finally, in the last line we note that z_{di} is conditionally independent of $\mathbf{w}_{-di}$ when we do **not** condition on w_{di}. One way to show this is by computing the probability $p(z_{di}|\mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta)$ which will require another application of Bayes rule and the chain rule. In doing so, we will realize that the probability $p(\mathbf{w}_{-di}|\mathbf{z})$ is again a constant with respect to z_{-di} and can be safely dropped. Or we can read this independence from the plate notation in Figure 2.1

where any path from z_{di} to any word in $\mathbf{w}_{-di}$ is blocked by (1) the converging arrows at x_{di} since we are not conditioning on x_{di} and (2) α and $\mathbf{z}_{-di}$ which are in our conditioning set. This "path" approach to determining independencies is known as d-separation. For an introduction to d-separation in Bayesian networks, see [53]. We must compute Equation 2.4 for $j = 1, \ldots, T$.

The probability of the current word w_{di} given its topic assignment z_{di}

The first factor in Equation 2.4 is the probability of the word w_{di} given the topic assignment $z_{di} = j$. Note that this is equivalent to,

$$p(w_{di}|z_{di} = j, \mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta) = \frac{p(\mathbf{w}|\mathbf{z}, \alpha, \eta)}{p(\mathbf{w}_{-di}|\mathbf{z}_{-di}, \alpha, \eta)}$$

Thus, if we can compute the marginal probability of a set of words $\mathbf{w}$ (or $\mathbf{w}_{-di}$) given the corresponding topic assignments $\mathbf{z}$ (or $\mathbf{z}_{-di}$) then we can use this equivalence to compute the first factor in Equation 2.4. We compute this marginal probability by integrating over the

topic variables $\boldsymbol{\phi}$

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{z},\alpha,\eta) &= \int_{\phi_1} \dots \int_{\phi_T} p(\mathbf{w}|\mathbf{z},\boldsymbol{\phi}) \cdot p(\boldsymbol{\phi}|\mathbf{z},\eta)\; d\phi_1 \dots d\phi_T \\
&= \int_{\phi_1} \dots \int_{\phi_T} \prod_d \prod_i p(w_{di}|z_{di},\boldsymbol{\phi}) \cdot \prod_t p(\phi_t|\eta)\; d\phi_1 \dots d\phi_T \\
&\propto \int_{\phi_1} \dots \int_{\phi_T} \prod_d \prod_i \phi_{z_{di},w_{di}} \cdot \prod_t \frac{1}{B(\eta_1,\dots,\eta_W)} \prod_w \phi_{tw}^{\eta_w - 1}\; d\phi_1 \dots d\phi_T \\
&= \int_{\phi_1} \dots \int_{\phi_T} \prod_t \frac{1}{B(\eta_1,\dots,\eta_W)} \prod_w \phi_{tw}^{\eta_w + n_{tw} - 1}\; d\phi_1 \dots d\phi_T \\
&= \prod_t \frac{1}{B(\eta_1,\dots,\eta_W)} \int_{\phi_t} \prod_w \phi_{tw}^{\eta_w + n_{tw} - 1}\; d\phi_t \\
&= \prod_t \frac{B(\eta_1 + n_{t1},\dots,\eta_W + n_{tW})}{B(\eta_1,\dots,\eta_W)} \\
&= \prod_t \frac{\Gamma(\sum_w \eta_w)}{\Gamma(\sum_w \eta_w + n_{tw})} \prod_w \frac{\Gamma(\eta_w + n_{tw})}{\Gamma(\eta_w)} \\
&\propto \prod_t \frac{\prod_w \Gamma(\eta_w + n_{tw})}{\Gamma(\sum_w \eta_w + n_{tw})}
\end{aligned}
\tag{2.5}
$$

A few observations. First, n_{tw} is the number of times the word w has been assigned to the topic t in the corpus. Second, the function $B(\cdot)$ is the multivariate Beta function which takes a vector as argument. Thus, $B(\eta_1,\dots,\eta_W)$ is the multivariate Beta function with argument $[\eta_1,\dots,\eta_W]$ and $B(\eta_1 + n_{t1},\dots,\eta_W + n_{tW})$ is the multivariate Beta function with vector argument $[\eta_1,\dots,\eta_W] + [n_{t1},\dots,n_{tW}]$. The Beta function can be expressed in terms of the Gamma function according to the identity $B(x) = \frac{\prod_i \Gamma(x_i)}{\Gamma(\sum_i x_i)}$. The Gamma function is the extension of the factorial to the real and complex numbers.

Next, the integral in the fifth line can be recognized as the density of an unnormalized Dirichlet distribution. If we multiply by the correct normalizing constant $B(\eta_1 + n_{t1},\dots,\eta_W + n_{tW})$, the integral evaluates to 1 and we are left with the ratio of normalizing constants from the posterior Dirichlet and the prior Dirichlet distributions.

Finally, note that the third and the last lines are not equalities but proportionalities. Recall that we are ultimately interested in computing the conditional distribution of z_{di}. Thus, any factor that is a constant with respect to z_{di} can be dropped. In the third line, we drop the multinomial coefficients from the multinomial likelihood. In the last line, we drop those factors that are functions of the prior hyper-parameters only.

We now compute the first factor in Equation 2.4, i.e. the probability of w_{di} given $z_{di} = j$.

$$\begin{aligned} p(w_{di}|z_{di} = j, \mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta) &= \frac{p(\mathbf{w}|\mathbf{z}, \alpha, \eta)}{p(\mathbf{w}_{-di}|\mathbf{z}_{-di}, \alpha, \eta)} \\ &= \prod_t \frac{\prod_w \Gamma(\eta_w + n_{tw})}{\Gamma(\sum_w \eta_w + n_{tw})} \cdot \frac{\Gamma(\sum_w \eta_w + n_{tw}^{-di})}{\prod_w \Gamma(\eta_w + n_{tw}^{-di})} \\ &= \prod_t \frac{\Gamma(\sum_w \eta_w + n_{tw}^{-di})}{\Gamma(\sum_w \eta_w + n_{tw})} \cdot \prod_w \frac{\Gamma(\eta_w + n_{tw})}{\Gamma(\eta_w + n_{tw}^{-di})} \end{aligned}$$

where n_{tw}^{-di} is the number of times word w has been assigned to topic t not including the current token w_{di}. This product over t can be greatly simplified when we observe that for $t \neq j$, n_{tw} equals n_{tw}^{-di} and thus the corresponding factor in the product is just 1. Furthermore, even when $t = j$, if w does not equal w_{di} then n_{jw} equals n_{jw}^{-di} and again the corresponding factor in the product over w is just 1. For simplicity, let $w_{di} = v$. Then,

$$\begin{aligned} p(w_{di}|z_{di} = j, \mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta) &= \prod_t \frac{\Gamma(\sum_w \eta_w + n_{tw}^{-di})}{\Gamma(\sum_w \eta_w + n_{tw})} \cdot \prod_w \frac{\Gamma(\eta_w + n_{tw})}{\Gamma(\eta_w + n_{tw}^{-di})} \\ &= \frac{\Gamma(\sum_w \eta_w + n_{jw}^{-di})}{\Gamma(\sum_w \eta_w + n_{jw})} \cdot \frac{\Gamma(\eta_v + n_{jv})}{\Gamma(\eta_c + n_{jv}^{-di})} \end{aligned}$$

Finally, we apply one last identity $\Gamma(x+1) = x\Gamma(x)$,

$$\begin{aligned} p(w_{di}|z_{di} = j, \mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta) &= \frac{\Gamma(\sum_w \eta_w + n_{jw}^{-di})}{\Gamma(\sum_w \eta_w + n_{jw})} \cdot \frac{\Gamma(\eta_v + n_{jv})}{\Gamma(\eta_v + n_{jv}^{-di})} \\ &= \frac{\Gamma(\sum_w \eta_w + n_{jw}^{-di})}{\Gamma(1 + \sum_w \eta_w + n_{jw}^{-di})} \cdot \frac{\Gamma(\eta_v + n_{jv}^{-di} + 1)}{\Gamma(\eta_v + n_{jv}^{-di})} \\ &= \frac{\eta_v + n_{jv}^{-di}}{\sum_w \eta_w + n_{jw}^{-di}} \end{aligned}$$

Thus, after all of that work, we arrive at the simple formula

$$p(w_{di}|z_{di} = j, \mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \eta) = \frac{\eta_v + n_{jv}^{-di}}{\sum_w \eta_w + n_{jw}^{-di}}$$

The probability of the topic assignment z_{di} given all other topic assignments

The second factor in Equation 2.4 is $p(z_{di} = j|\mathbf{z}_{-di}, \alpha, \eta)$ the probability of the topic assignment z_{di} given all other topic assignments. We compute this probability in a similar manner by integrating over the document-specific distribution θ_d.

$$\begin{aligned} p(z_{di} = j|\mathbf{z}_{-di}, \alpha, \eta) &= \frac{p(\mathbf{z}_d|\alpha)}{p(\mathbf{z}_{d,-i}|\alpha)} \\ &= \frac{\Gamma(\sum_t \alpha_t + n_{dt}^{-di})}{\Gamma(\sum_t \alpha_t + n_{dt})} \cdot \prod_t \frac{\Gamma(\alpha_t + n_{dt})}{\Gamma(\alpha_t + n_{dt}^{-di})} \\ &= \frac{\alpha_t + n_{dt}^{-di}}{\sum_t \alpha_t + n_{dt}^{-di}} \\ &\propto \alpha_t + n_{dt}^{-di} \end{aligned}$$

A few observations. First, $\mathbf{z}_{d,-i}$ represents the topic assignments of the words from document

d not including the current token z_{di}. Note that z_{di}, conditioned on $\mathbf{z}_{d,-i}$, is independent of the topic assignments from the other documents. Thus, we need only condition on the value of $\mathbf{z}_{d,-i}$ and we can remove any other topic assignment variables. The first and second lines are derived in the same way as those in the previous section. In the third line, we use again the identity $\Gamma(x+1) = x\Gamma(x)$. Finally, we note that $\sum_t n_{dt}^{-di}$ is the number of words in document d minus one and as such is not a function of z_{di} and can be dropped.

The final sampling equation

We now state the final Gibbs sampling equation for z_{di}:

$$p(z_{di} = j|\mathbf{z}_{-di}, \mathbf{w}, \alpha, \eta) \propto \left(\frac{\eta_v + n_{jv}^{-di}}{\sum_w \eta_w + n_{jw}^{-di}}\right) \cdot \alpha_t + n_{dt}^{-di} \tag{2.6}$$

This probability is computed for $j \in \{1, \ldots, T\}$. Since Equation 2.6 is not an equality, we must normalize so that the probabilities sum to 1. We then sample a new value for z_{di} which transitions us to a new state in the Markov chain.

Latent Dirichlet allocation (LDA) and the collapsed Gibbs sampler presented here provide a basis and starting point for the work in the remaining chapters of this thesis.

Chapter 3

Sentence classification in scientific articles

3.1 Introduction

In this chapter we present an application of latent Dirichlet allocation to sentence classification in scientific articles. Sentence classification refers to the task of labeling sentences according to function. Within the domain of scientific articles, typical sentence functions include:

AIM	The sentence states the aim, or primary goal, of the article
RESULT	The sentence states a result of the investigation presented in the article
HYPOTHESIS	The sentence states a hypothesis of the authors'
PAST WORK	The sentence describes past work
CRITIQUE	The sentence critiques past work

Sentence classification provides a deeper understanding of the argumentative structure of a scientific article [38, 65] and is an important component for tasks such as document sum-

marization [69, 24]. Sentence classification can also be useful for tasks such as identifying scientific debates or identifying influential algorithms or techniques in a given scientific domain [26].

Sentence classification is typically accomplished by training a supervised classifier using a manually-labeled corpus of sentences. While current supervised approaches generally perform well under controlled circumstances, creating an annotated corpus of sentences can be time-consuming. Indeed, according to one of the key researchers in the field of sentence classification, "manual annotation is prohibitively expensive" [68]. As such, we explore statistical models for sentence classification that require a smaller number of labeled sentences from which to extract prior knowledge for informative priors.

Guiding our work is the hypothesis that sentences are composed of two fundamentally different types of words and phrases: domain-dependent and domain-independent. Intuitively, a domain-dependent phrase is specific to a particular scientific domain. If we observe the phrase "support vector machine" in a sentence, we are more likely to believe that the sentence is from a machine learning article. If we observe the phrases "chromosome evaluation" or "functional regions of DNA", we are more likely to believe that the sentence is from computational biology. In contrast, domain-independent phrases are equally likely to be found in sentences from any domain. If we observe the phrases "in this study" or "were derived from", we are no more informed as to whether the domain is machine learning, computational biology or some other discipline. We hypothesize that the domain-dependent words and phrases provide the topical content of the sentence whereas the domain-independent words and phrases indicate the function of the sentence. As such, we look to the domain-independent words and phrases as a strong signal of the sentence function.

3.2 Related work

3.2.1 Supervised classification

The vast majority of existing work on sentence classification employs a supervised learning approach. Common classifiers include conditional random fields [28, 31, 2, 25, 24, 11], naive Bayes classifiers [69, 24], support vector machines [63, 25, 24], hidden Markov models [35] and maximum entropy models [68]. In addition to the type of classifier used, past work can be distinguished in four regards: scope, annotation scheme, features, and domain.

The scope of the task refers to whether classification is performed on the abstract sentences only [28, 63, 24, 11] – which is thought to be an easier task since fewer sentence types occur in the abstract (see [69] pg. 423 and [28]) – or on the entire text of the article [69, 35, 68, 1]. Alternatively, other past work has focused on a specific section within the article [2].

The second aspect in which past work differs is the annotation scheme, i.e. the set of labels used for classification. The most basic annotation scheme is modeled after the scientific method: aim, method, results, conclusion [28, 1, 63][1]. An example of a more complex annotation scheme is Argumentative Zones [69, 70] – which tries to identify, for each knowledge claim, both the author's sentiment towards the claim as well as the originator of the claim – and CoreSC [36] – which is an elaboration on the basic annotation scheme. Comparisons of these three annotation schemes can be found in [24]. Special-purpose annotation schemes have also been developed for more specific classification tasks [2, 11].

The third aspect is the set of features used to train the classifier. This typically includes word features such as unigrams, and bigrams; grammatical information such as part-of-speech, tense, and voice; context information such as features from the preceding and following sentences; positioning information such as the position of the sentence in the article and the

[1] Variations include the introduction, method, results, and discussion annotation scheme

position of the sentence in the paragraph; as well as other features such as the presence of a citation or the length of the sentence.

Finally, application domains include clinical reports [35], biomedical articles [28, 1, 11, 63], non-biomedical scientific articles [2], and legal texts [27] among others.

3.2.2 Semi-supervised and unsupervised classification

Guo et. al [25] use four semi-supervised classifiers for sentence classification: three variants of the support vector machine and a conditional random field model. The semi-supervised classifiers either (1) start with a small set of labeled data and choose, at each iteration, additional unlabeled data to be labeled and added to the training set (known as active learning) or (2) include the unlabeled data in the classifier formulation with an estimate of, or distribution over, the unknown labels. They perform sentence classification on biomedical abstracts using a version of the Argumentative Zones annotation scheme developed specifically for biology articles. They present experiments using only 100 labeled abstracts (approximately 700 sentences) to train the different classifiers.

There are a number of key differences between this work and our own. First, the work of Guo et al. is more focused than our own work: Guo et al. perform classification on abstract sentences from biological articles only. We classify abstract and introduction sentences in scientific articles from various domains (computational biology, machine learning, and psychology). Second, we use less labeled data. We use only 228 sentences (from the abstract and introduction of 15 scientific articles) to construct informative priors for the statistical models presented in this chapter. This is approximately equal to 33 abstracts as opposed to 100 abstracts used by Guo et al. Finally, the annotation scheme used by Guo et al. lacks some of the rarer, and harder to identify, labels from the original Argumentative Zones annotation scheme which is the annotation scheme used in this work.

Wu et al. [29] use a hidden Markov model to label sentences in scientific abstracts. They first label a set of 106 abstracts (709 sentences). They use the labeled data to extract pairs of words from sentences that are strong indicators of a particular label. They then use these word pairs and the labeled sentences to train a hidden Markov model. Again, we use less labeled data than Wu et al. Also, the annotation scheme used by Wu et al. (based on the scientific method) differs from the annotation scheme used in this paper.

Varga et al. [73] use latent Dirichlet allocation to predict the section in which a sentence appears (e.g. abstract, introduction, related works). Varga et al. use no labeled data since they do not attempt to bias the learned multinomial distributions for each topic to reflect the semantics of the different sections in a scientific article. The key difference between our work and Varga et al. is that we are interested in predicting sentence function and not the sentence section. In addition, although we too employ latent Dirichlet allocation, we use labeled data to create informative priors to bias the learned multinomial distributions to reflect the semantics of the different labels in our annotation scheme.

3.3 Annotation scheme

We use an annotation scheme that is derived from Argumentative Zones [69] (AZ). There are five labels in our annotation scheme: OWN, CONTRAST, BASIS, AIM and MISCELLANEOUS. Table 3.1 gives a brief description of each label and shows example sentences.

We had four desiderata when choosing the labels in our annotation scheme: (1) each label should be learnable from the data, i.e. there should exist textual clues indicating the presence of each label (2) each label should exist with some minimal frequency in the data (3) each label should be potentially useful for other applications and (4) each label should be identifiable by human annotators [2].

[2]We experimented with several annotation schemes and found that it is surprisingly hard to find a scheme

Label	Description
AIM	**The primary goal of the article.** *In this paper we answer this question in the negative by explicitly constructing a regularized risk minimization problem for which BMRM takes at least SYMBOL iterations* *This paper examines how one such factor, the Big-Five personality trait of openness-to-experience, influences the effect of previously presented anchors on participants' judgments.*
OWN	**The author's own work; includes the method, results, and conclusions of the article.** *Our factor regression model is fundamentally nonparametric.* *These structures are then analyzed through simulations based on the FORMULA model.*
CONTRAST	**A statement of the limitations/weaknesses of past research, or a contrast/comparison to past research.** *We address three fundamental shortcomings of standard factor analysis approaches...* *However these demonstrations have been restricted primarily to laboratory surveys in which task of recalling examples then making an overall assessment may seem somewhat artificial to participants and the responses of little consequence*
BASIS	**A statement that the author's work is an extension of, adaptation of, or similar to, past research.** *In this study, we extended and refined earlier work by Allen et al., which proposed a stoichiometric formalism to model protein synthesis and illustrated it on some E. coli genes and operons CITATION.* *Using a previously well-validated realistic model of a CA1 pyramidal neuron we demonstrate that stochastic ion channel gating influences spike output in response to dendritic synaptic input.*
MISC	**Any other sentence.** *Non-native contacts are considered repulsive.* *A simple approach is to add noise sources to deterministic models.*

Table 3.1: The labels in our annotation scheme (based on Argumentative Zones) along with example sentences.

The AZ annotation scheme includes one additional label TEXTUAL which describes sentences that discuss the structure of the article, e.g. "In Section 3, we show that...". We removed the label TEXTUAL because it was not of obvious use for other applications. We also collapsed two of the labels in AZ – NEUTRAL and OTHER – into one label MISCELLANEOUS. The label NEUTRAL describes sentences that refer to past work in a neutral way. The label OTHER describes sentences that state generally accepted background information. We collapsed these two labels because it was often hard, in practice, to distinguish between them.

Thus the five labels above fit our requirements. The labels CONTRAST and BASE are of particular interest because if we were able to identify such sentences with high accuracy, then we could potentially identify scientific debates or identify influential algorithms and techniques in a given scientific domain.

3.4 Statistical models for sentence classification

In this section, we present two statistical models for sentence classification based on latent Dirichlet allocation [5].

We begin this section with an introduction to latent Dirichlet allocation (LDA). We then introduce our first model which modifies LDA in order to better apply it to the task of sentence classification. For simplicity, we will refer to this model as SentenceLDA. Finally, we introduce a second model which refines SentenceLDA by explaining the presence of both domain-dependent and domain-independent words in a sentence. We refer to this model as Multicorpus SentenceLDA.

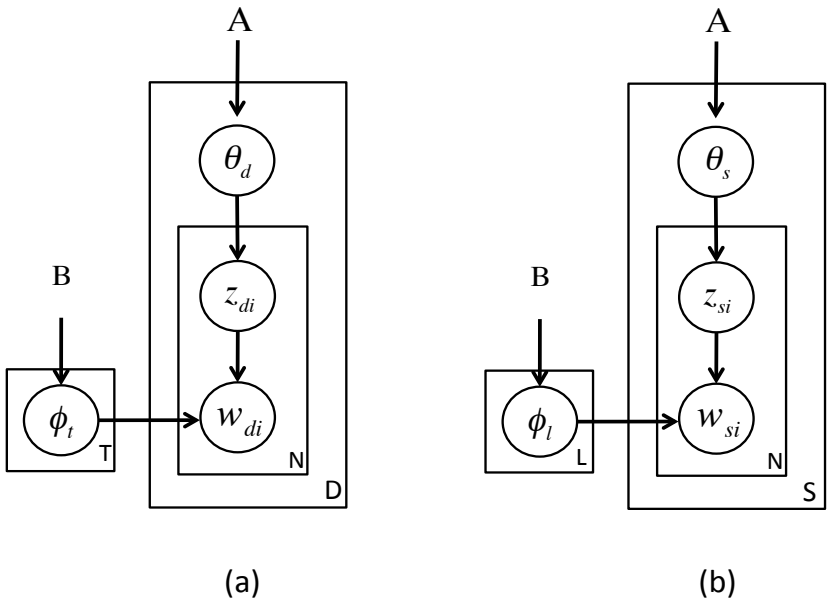

Figure 3.1: (a) plate notation for latent Dirichlet allocation (b) plate notation for latent Dirichlet allocation where the T topics have been replaced by L labels and the D documents have been replaced by S sentences.

3.4.1 Latent Dirichlet allocation

Figure 3.1 (a) shows the plate notation for latent Dirichlet allocation (LDA). Each topic $t \in \{1, \ldots, T\}$ is associated with a multinomial distribution over the words in some vocabulary. The probability vector of this multinomial distribution is denoted $\phi_t = (\phi_{t1}, \ldots, \phi_{tW})$ where W is the size of the vocabulary, $0 \leq \phi_{ti} \leq 1$, and $\sum_i \phi_{ti} = 1$. Similarly, each document $d \in \{1, \ldots, D\}$ is associated with a multinomial distribution over topics denoted $\theta_d = (\theta_{d1}, \ldots, \theta_{dT})$. To generate the ith word in the dth document, a topic assignment is first sampled $z_{di} \sim \text{Multinomial}(\theta_d)$. Given this topic assignment $z_{di} \in \{1, \ldots, T\}$, a word is then sampled from the corresponding topic $w_{di} \sim \text{Multinomial}(\phi_{z_{di}})$. This process is repeated for each word in the document.

Both the document distributions θ_d and the topic distributions ϕ_t are sampled from Dirichlet priors with hyper-parameters given by the matrices A and B respectively. It is through these

that fulfills all four requirements!

hyper-parameter matrices that we incorporate prior knowledge into the statistical model. We discuss this in more detail in Section 3.5.2.

3.4.2 SentenceLDA

In our first model SentenceLDA (Sent-LDA), we make three modifications to LDA in order to better apply it to the task of sentence classification.

Modification 1

The first modification we make to LDA is cosmetic, modifying only the semantics of the random variables in LDA. Instead of learning a distribution over **topics** for each **document** we want to learn a distribution over the **labels** in our annotation scheme for each **sentence**. Thus we replace the T multinomial distributions that correspond to topics with L multinomial distributions that correspond to the L labels in our annotation scheme. We denote these L multinomial distributions as $\phi_l = (\phi_{l1}, \ldots, \phi_{lW})$ for $l \in \{1, \ldots, L\}$. Also, instead of aggregating words at the document level, we aggregate words at the sentence level. This means that instead of indexing words by document we index words by sentence – i.e. x_{si} denotes the ith word in the sth sentence and $z_{si} \in \{1, \ldots, L\}$ is the corresponding latent assignment variable. Figure 3.1 (b) shows the plate notation after these changes.

Note that the sentences in a document are now conditionally independent given the label distributions ϕ_l and the sentence distributions θ_s. Modification 3 (presented below) allows us to recover this "grouping by document" as well as explore other groupings that may be more natural for sentence classification.

Modification 2

In LDA, each word has its own latent assignment variable (denoted as z_{di} in Figure 3.1 (a) and z_{si} in Figure 3.1 (b)). Since we are ultimately interested in predicting the label of each sentence (and not each word), it might be more effective to have one latent assignment variable for the entire sentence. We experiment with both options: one latent assignment variable for each word $z_{si} \in \{1, \ldots, L\}$ that indicates the label assignment of the word versus one latent assignment variable for each sentence $z_s \in \{1, \ldots, L\}$ that indicates the label assignment of the sentence. The latter case corresponds to a mixture model where every word in the sentence is assigned to the same label. For ease of reference, we refer to SentenceLDA with one latent assignment variable for each word as Sent-LDA-W and SentenceLDA with one latent assignment variable for each sentence as Sent-LDA-S.

Modification 3

Finally, we note that it might be difficult to learn a distribution over labels θ_s for each sentence since sentences are relatively short. Instead we propose to group sentences together in order to share "statistical strength". We group together sentences that have a similar distribution over labels and then utilize all of the sentences in the group to learn a single distribution.

To make this clearer, Figure 3.2 shows the final plate notation for Sent-LDA combining all three of our proposed modifications. Figure 3.2 (a) shows the plate notation for Sent-LDA-W and Figure 3.2 (b) shows the plate notation for Sent-LDA-S.

Note that the sentence distributions θ_s have been replaced by **group** distributions $\theta_g = (\theta_{g1}, \ldots, \theta_{gL})$. We introduce an index variable for each sentence $\gamma_s \in \{1, \ldots, G\}$ that indicates the group membership of sentence s. Here G is the total number of groups.

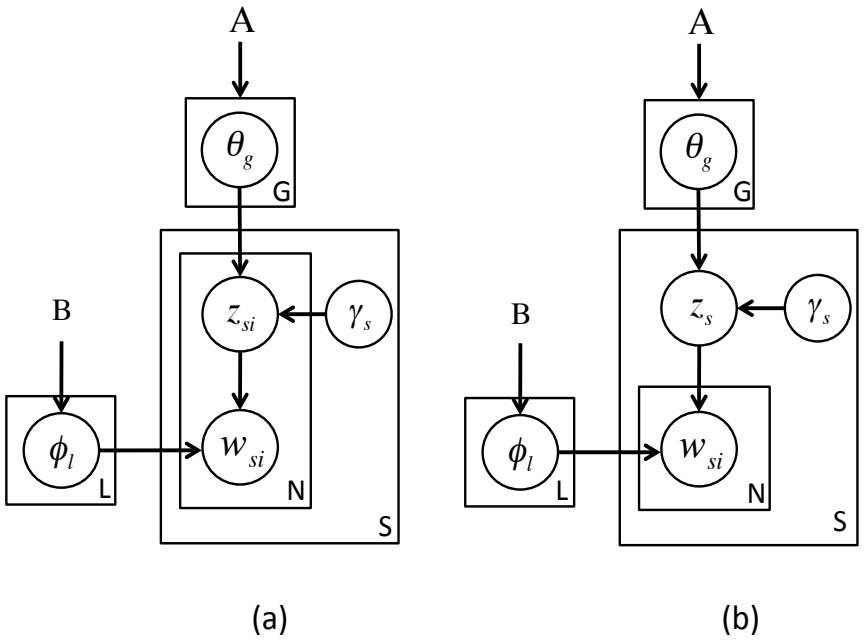

Figure 3.2: (a) SentenceLDA with one latent assignment variable for each word. We refer to this model as Sent-LDA-W (b) SentenceLDA with one latent assignment variable for each sentence. We refer to this model as Sent-LDA-S

We can recover the sentence distributions θ_s by having each sentence be its own group. In this case, $G = S$ (the number of sentences) and we would estimate a separate distribution θ_s for each sentence. However, our framework is more general and allows us to experiment with other groupings of sentences. For example, one might expect sentences in the abstract of an article to exhibit a different distribution over labels than sentences in the introduction (e.g. abstract sentences rarely reference past work). In this case, we might group all abstract sentences together in one group and all introduction sentences together in another. For each of these two groups we would estimate a different distribution over labels. The distribution estimated from the introduction sentences would give higher probability to the labels CONTRAST and BASE then the distribution estimated from the abstract sentences. In our experiments, we explore several different ways of grouping sentences.

SentenceLDA is similar to an author-topic model [60]. If we modified the author-topic model by replacing documents with sentences (Modification 1) and restricted each sentence

to have only one "author" which corresponded to our notion of groups, then we would recover SentenceLDA.

As a final note, the group distributions θ_g and the label distributions ϕ_l are sampled from Dirichlet priors:

$$\theta_g \sim \text{Dirichlet}(A_{g1}, \ldots, A_{gL})$$
$$\phi_l \sim \text{Dirichlet}(B_{l1}, \ldots, B_{lW})$$

where A_{gl} is the prior count for label l in group g and B_{lw} is the prior count for word w in label l. It is through these hyper-parameters that we incorporate prior knowledge into the statistical model. We discuss this in more detail in Section 3.5.2.

3.4.3 Inference for SentenceLDA

Given a collection of sentences, the word variables $\mathbf{w} = \{w_{si}\}$ and the group membership variables $\boldsymbol{\gamma} = \{\gamma_s\}$ are observed and fixed. What is unknown are the latent assignment variables: either $\mathbf{z} = \{z_{si}\}$ for Sent-LDA-W or $\mathbf{z} = \{z_s\}$ for Sent-LDA-S. The label distributions $\boldsymbol{\phi} = \{\phi_l\}$ and the group distributions $\boldsymbol{\theta} = \{\theta_g\}$ are also unknown. Thus we would like to compute, and draw samples from, the posterior distribution over the unknown variables conditioned on the data, i.e. $p(\mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\theta} | \mathbf{w}, \boldsymbol{\gamma})$.

There are many inference algorithms that could be used to sample from this posterior distribution. We employ Gibbs sampling, a Markov Chain Monte Carlo (MCMC) technique [21]. In particular, we marginalize the label distributions $\boldsymbol{\phi}$ and the group distributions $\boldsymbol{\theta}$ out of the posterior and use a collapsed Gibbs sampler to draw samples of the latent assignment variables $\mathbf{z}$. This is a commonly used technique when performing inference for latent

Dirichlet allocation [23]. Using a collapsed Gibbs sampler has a number of potential benefits including faster mixing of the Gibbs sampler and a simple implementation.

In the remainder of this section, we present the collapsed Gibbs sampling equations for Sent-LDA-W and Sent-LDA-S. We condition on the hyper-parameter matrices A and B and the group membership variables $\boldsymbol{\gamma}$ in all equations even if this is not explicitly shown.

Sent-LDA-W: one latent variable per word

Recall that $z_{si} \in \{1, \ldots, L\}$ is the latent assignment variable of the ith word in the sth sentence. The Gibbs sampling algorithm samples each variable in the statistical model from its conditional distribution given all other random variables. Thus, we need to compute the conditional distribution of z_{si} given all other latent assignment variables $\mathbf{z}_{-si}$ and all words $\mathbf{w}$:

$$p(z_{si} = j|\mathbf{z}_{-si}, \mathbf{w}) \propto p(w_{si}|z_{si} = j, \mathbf{z}_{-si}, \mathbf{w}_{-si}) \cdot p(z_{si} = j|\mathbf{z}_{-si}) \tag{3.1}$$

for $j \in \{1 \ldots L\}$. The first factor in Equation 3.1 is the probability of the word w_{si} given the label assignment $z_{si} = j$. This probability can be computed by marginalizing out $\boldsymbol{\phi}$ from the joint distribution $p(\mathbf{w}, \boldsymbol{\phi}|z_{si} = j, \mathbf{z}_{-si})$. For simplicity, let $w_{si} = w$. Then we have

$$p(w_{si}|z_{si} = j, \mathbf{z}_{-si}, \mathbf{w}_{-si}) = \frac{B_{jw} + n_{jw}^{-si}}{\sum_{w'} B_{jw'} + n_{jw'}^{-si}} \tag{3.2}$$

where B_{jw} is the prior count for word w in label j and n_{jw} is the total number of times the

word w has been assigned to the label j. The superscript $-si$ indicates that we do not count the current token w_{si} in this total.

The second factor in Equation 3.1 is the probability of the latent assignment $z_{si} = j$ conditioned on the other latent assignment variables $\mathbf{z}_{-si}$. This conditional probability is also computed by marginalizing out $\boldsymbol{\theta}$ from the joint distribution $p(z_{si} = j, \mathbf{z}_{-si}, \boldsymbol{\theta})$:

$$p(z_{si} = j | z_{-si}, \gamma_s = g) = \frac{A_{gj} + n_{gj}^{-si}}{\sum_{j'} A_{gj'} + n_{gj'}^{-si}} \tag{3.3}$$

where A_{gj} is the prior count for label j in group g and n_{gj} is the total number of words in group g assigned to label j. To be more precise, n_{gj} is actually the total number of words assigned to label j whose sentence's group membership γ_s is g. Again, the superscript $-si$ indicates that we do not include the current token in this total. See Section 2.2.2 for a full derivation of Equation 3.2 and Equation 3.3.

Updating the latent assignment variable z_{si} for every word in the corpus requires $O(NL)$ time where N is the total number of words in the corpus. For each such word, we must compute Equation 3.1 for $j \in \{1, \ldots, L\}$. Computing Equation 3.1 takes $O(1)$ time if we store the counts n_{jw} and n_{gj} (in an LxW and GxL matrix respectively). We must also store the total number of words currently assigned to the label j, $\sum_w n_{jw}$, in a vector of length L. The remaining terms $\sum_w B_{jw}$, $\sum_j A_{gj}$, and $\sum_j n_{gj}$ can be pre-computed.

Sent-LDA-S: one latent variable per sentence

Instead of having one latent variable per word, we may choose to have one latent variable per sentence enforcing the constraint that all words in the sentence must be assigned to the

same label. In this case, $z_s \in \{1, \ldots, L\}$ is the latent assignment variable for sentence s. We need to compute the conditional distribution of z_s given all other latent assignment variables $\mathbf{z}_{-s}$ and all words $\mathbf{w}$. This conditional distribution is given by,

$$p(z_s = j|\mathbf{z}_{-s}, \mathbf{w}) \propto p(\mathbf{w}_s|z_s = j, \mathbf{z}_{-s}, \mathbf{w}_{-s}) \cdot p(z_s = j|\mathbf{z}_{-s}) \tag{3.4}$$

for $j \in \{1 \ldots L\}$. The first factor in Equation 3.4 is the probability of *all* the words in sentence s given the label assignment $z_s = j$. We again compute this by marginalizing out the label distributions $\boldsymbol{\phi}$ from the joint distribution $p(\mathbf{w}, \boldsymbol{\phi}|z_s = j, \mathbf{z}_{-s})$.

$$p(\mathbf{w}_s|z_s = j, \mathbf{z}_{-s}, \mathbf{w}_{-s}) = \frac{\Gamma(\sum_{w'} B_{jw'} + n_{jw'}^{-s})}{\Gamma(\sum_{w'} B_{jw'} + n_{jw'})} \prod_{w=1}^{W} \frac{\Gamma(B_{jw} + n_{jw})}{\Gamma(B_{jw} + n_{jw}^{-s})} \tag{3.5}$$

B_{jw} and n_{jw} have the same semantics from the previous section however the superscript $-s$ indicates that we remove **all** words from sentence s from the total count n_{jw}. The function $\Gamma(\cdot)$ is known as the Gamma function. The Gamma function is the extension of the factorial function to the real and complex numbers.

The second factor in Equation 3.4 is the probability of the latent assignment variable $z_s = j$ conditioned on the other latent assignment variables $\mathbf{z}_{-si}$.

$$p(z_s = j|z_{-s}, \gamma_s = g) = \frac{A_{gj} + n_{gj}^{-s}}{\sum_{j'} A_{gj'} + n_{gj'}^{-s}} \tag{3.6}$$

where n_{gj} is now the total number of sentences (not words) with group membership g assigned

to label j, and the superscript $-s$ indicates we do not include the current sentence in this total count.

Updating the latent assignment variables z_s for every sentence in the corpus requires $O(SL(N_s + W))$ time where S is the number of sentences in the corpus, L is the number of labels in the annotation scheme, N_s is the average sentence length, and W is the size of the vocabulary. For each of the S sentences in the corpus, we must compute Equation 3.4 for $j \in \{1, \ldots, L\}$. Again, assuming we store the counts n_{jw}, n_{gl}, and $\sum_w n_{jw}$ and we pre-compute $\sum_w B_{jw}$, $\sum_j A_{gj}$, and $\sum_j n_{gj}$ then we can compute Equation 3.4 for a given value of s and j in $O(N_s + W)$ time. We must first iterate over the words in the sentence s and decrement the corresponding count n_{jw} for each word which requires $O(N_s)$ time. Once the counts have been decrement, we can compute Equation 3.5 in $O(W)$ time and Equation 3.6 in $O(1)$ time.

3.4.4 Multicorpus SentenceLDA

One of the drawbacks of SentenceLDA is that every word must be assigned to one of the L labels. Regardless of whether we have one latent variable per word or one latent variable per sentence, every word has to be assigned to a label. Table 3.2 shows results from running Sent-LDA-W on a collection of articles from three different domains: computational biology, machine learning, and psychology[3]. We show the ten words with the highest probability for each label in the annotation scheme. CITATION, EQUATION, and NUMBER are markers used for word tokens corresponding to citations, equations, and numbers respectively.

Note the presence of domain-dependent words such as "protein", "decision", "information" and "people" for the first four labels. Also, note the overwhelming presence of domain-independent words such as "at", "each", and "all" which act as noise.

[3]The implementation details of how we trained this model are not discussed here since our goal is to illustrate a shortcoming of SentenceLDA as opposed to discussing the details of the model.

AIM	OWN	CONTRAST	BASE	MISC
this	we	CITATION	more	between
we	our	EQUATION	one	when
data	their	not	two	than
at	information	been	other	also
decision	but	may	they	EQUATION
each	used	problem	NUMBER	models
protein	results	learning	number	function
how	algorithm	different	if	into
set	both	all	percent	probability
about	were	some	people	studies

Table 3.2: For each label we show the top 10 words with the highest probability. These word distributions were learned using Sent-LDA-W

The problem is that the labels in our annotation scheme must account for every word in the corpus – whether or not that word is indicative of the sentence's function. To solve this problem, we propose an extension of Sent-LDA-W. Our new model utilizes articles from different scientific domains to isolate domain-dependent words, e.g. "protein" or "decision", and uses prior knowledge to sort the remaining domain-independent words into those that are truly indicative of the sentence's function as opposed to those that are simply noise[4]

Figure 3.3 shows the plate notation for this new model which we call Multicorpus SentenceLDA (MC-LDA). The generative process of MC-LDA combines two mechanisms for generating a word. The first mechanism is analogous to Sent-LDA-W. Sentences are partitioned into G groups. Each group is associated with a multinomial distribution over labels θ_g. A label is sampled from θ_g and a word is generated from the corresponding label distribution ϕ_l.

The second mechanism is analogous to traditional LDA. Sentences are grouped into documents (we use the term "document" to mean an actual document or article). Each document

[4]We removed only a small list of 25 stopwords (e.g. "the" , "a"). All remaining words we kept in the vocabulary since many traditional stopwords are actually useful for our task. For example, "we", "our", "this", and "how" are typically removed as stopwords however they are strong indicators of the labels AIM and OWN. The full list of stopwords we remove can be found in Appendix C

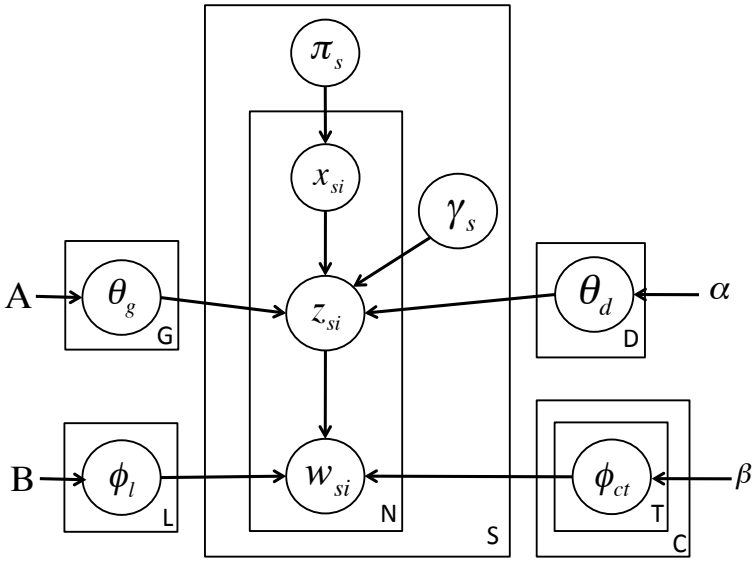

Figure 3.3: Graphical model that incorporates multiple corpora.

is associated with a multinomial distribution, denoted θ_d, over a set of T **corpus-specific** topics. A topic is sampled from θ_d and a word is generated by the corresponding topic distribution ϕ_{ct}. Here c is the corpus index and t is the topic index. Each of the C corpora have a set of T corpus-specific topics.

Only a word from a document in corpus c can be assigned to one of the T corpus-specific topics. However, any word from any document in any corpus can be assigned to one of the L labels ϕ_l. This arrangement encourages those words that only occur within a corpus to be explained by the corpus-specific topics and those words that occur across corpora to be explained by the labels. Thus MC-LDA requires at least two corpora from different domains to effectively separate domain-dependent words from domain-independent words.

The variable x_{si} is a binary switch that controls which of these two mechanisms generates the word w_{si}. If $x_{si} = 0$, the word w_{si} is generated by one of the L labels. If $x_{si} = 1$, the word w_{si} is generated by one of the corpus-specific topics. We sample the switch variables x_{si} from a sentence-specific Bernoulli distribution parameterized by $\pi_s \in (0, 1)$.

Note that we still retain the informative priors captured by the hyper-parameter matrices A and B. However, the document distributions θ_d and the topics ϕ_{ct} are sampled from symmetric Dirichlet priors with scalar parameter α and β respectively.

MC-LDA is similar to the background-words (BW) topic model by Chemudugunta et al. [10] which also uses a binary variable to control whether a word is generated by a set of global topics or by a corpus-specific topic. In the BW topic model, each document has a distribution over T global topics. The T global topics in the BW topic model are analogous to the L label distributions ϕ_l in SentenceLDA and Multicorpus SentenceLDA. In the BW topic model, when the switch variable equals 1 a topic is sampled from the document's distribution over the global topics and a word is emitted from the sampled topic. When the switch variable equals 2, a word is emitted from a single corpus-specific topic. This single corpus-specific topic is analogous to the corpus-specific topics in MC-LDA. The main difference between MC-LDA and the BW topic model is that MC-LDA models two different aspects of a document. In the first aspect, sentences are explicitly represented and a document is modeled as a collection of independent sentences. That is, the sentences in a document are partitioned into groups and given the group distribution over labels θ_g, two sentences in a document are independent. In the second aspect, a document is modeled as a bag of topically-related words. In this aspect, there is no explicit representation of sentences – only words. In contrast, in the BW topic model, a document is only modeled as a bag of words. There is no dual representation where sentences are explicitly modeled.

Furthermore, in MC-LDA, we use informative priors to bias the meaning of the L label distributions to reflect the labels in our annotation scheme. In the BW topic model, the T global topics have no external meaning and are used only to isolate background words. These background words are not explicitly used to capture any meaningful aspect of the document collection.

We present the entire generative process for MC-LDA in Table 3.3 for the reader's conve-

1. For label $l \in \{1, \ldots, L\}$
   Sample distribution over words $\phi_l \sim \text{Dirichlet}(B_{l1}, \ldots, B_{lW})$
2. For group $g \in \{1, \ldots, G\}$
   Sample a distribution over labels $\theta_g \sim \text{Dirichlet}(A_{g1}, \ldots, A_{gL})$
3. For corpus $c \in \{1, \ldots, C\}$ and topic $t \in \{1, \ldots, T\}$
   Sample corpus-specific topic $\phi_{ct} \sim \text{Dirichlet}(\beta)$
4. For document $d \in \{1, \ldots D\}$
   Sample document-specific distribution over corpus topics $\theta_d \sim \text{Dirichlet}(\alpha)$
5. For sentence $s \in \{1, \ldots, S\}$
   Sample Bernoulii parameter $\pi_s \sim \text{Beta}(a, b)$
   For each word $i \in \{1, \ldots, N_{si}\}$
      Sample switch $x_{si} \sim \text{Bernoulli}(\pi_s)$
      Sample $z_{si} \sim \text{Multinomial}(\theta_{\gamma_s})$ if $x_{si} = 0$
      Sample $z_{si} \sim \text{Multinomial}(\theta_d)$ if $x_{si} = 1$
      Sample word $w_{si} \sim \text{Multinomial}(\phi_{z_{si}})$

Table 3.3: Generative model for multi-corpora LDA (MC-LDA)

nience.

3.4.5 Inference for Multicorpus SentenceLDA

Inference for MC-LDA is similar to inference for Sent-LDA-W. Given a collection of articles, the word variables $\mathbf{w} = \{w_{si}\}$ and the group assignment variables $\boldsymbol{\gamma} = \{\gamma_s\}$ are known and observed[5].

We again employ a collapsed Gibbs sampler. We marginalize over the group distributions $\boldsymbol{\theta}_g = \{\theta_g\}$, the label distributions $\boldsymbol{\phi}_l = \{\phi_l\}$, the document distributions $\boldsymbol{\theta}_d = \{\theta_d\}$, and the topic distributions $\boldsymbol{\phi}_{ct} = \{\phi_{ct}\}$. All that remains to be sampled are the latent cluster assignments $\mathbf{z} = \{z_{si}\}$, the switch variables $\mathbf{x} = \{x_{si}\}$, and the Bernoulli parameters $\boldsymbol{\pi} = \{\pi_s\}$.

[5]We also observe the document id for each sentence and the corpus id for each sentence.

Block sampling the latent assignment variables z and switches x

Let z_{si} denote the latent assignment of the ith word in sentence s. If the binary switch $x_{si} = 0$, then $z_{si} \in \{1, \ldots, L\}$. Otherwise, $z_{si} \in \{1, \ldots, T\}$. Since the assignment variable z_{si} changes as the binary switch x_{si} changes, we block sample both variables together. Thus, we need to compute the conditional distribution of z_{si} and x_{si} given all other cluster assignments $\mathbf{z}_{-si}$, words $\mathbf{w}$, switches $\mathbf{x}_{-si}$, and parameters $\boldsymbol{\pi}$:

$$p(z_{si} = j, x_{si} | \mathbf{z}_{-si}, \mathbf{w}, \mathbf{x}_{-si}, \boldsymbol{\pi}) \propto p(w_{si} | \mathbf{z}, \mathbf{w}_{-si}) \cdot p(z_{si} = j | \mathbf{z}_{-si}, \mathbf{x}) \cdot p(x_{si} | \boldsymbol{\pi}) \tag{3.7}$$

The first factor in Equation 3.7 is the probability of the word w_{si} given the assignment $z_{si} = j$. This probability is analogous to the sampling equation for Sent-LDA-W shown in Equation 3.2. Let $w_{si} = w$. Then,

$$\begin{aligned} p(w_{si} | z_{si} = j, \mathbf{z}_{-si}, \mathbf{w}_{-si}) &= \frac{B_{jw} + n_{jw}^{-si}}{\sum_{w'} B_{jw'} + n_{jw'}^{-si}} \qquad \text{if } x_{si} = 0 \\ p(w_{si} | z_{si} = j, \mathbf{z}_{-si}, \mathbf{w}_{-si}) &= \frac{\beta + n_{jw}^{-si}}{\sum_{w'} \beta + n_{jw'}^{-si}} \qquad \text{if } x_{si} = 1 \end{aligned} \tag{3.8}$$

The counts n_{jw} depend upon the switch variable x_{si}. If $x_{si} = 0$, then n_{jw} is the total number of words w assigned to the label j. Otherwise, n_{jw} is the total number of words w assigned to the corpus specific topic j. The superscript $-si$ indicates that we do not include the current token in this total. Likewise, the prior count is either B_{jw} (if $x_{si} = 0$) or β (if $x_{si} = 1$).

The second factor in Equation 3.7 is the probability of the cluster assignment $z_{si} = j$ and is

also analogous the sampling equation for Sent-LDA-W shown in Equation 3.3.

$$
\begin{aligned}
p(z_{si} = j|\mathbf{z}_{-si}, \mathbf{x}, \gamma_s = g) &= \frac{A_{gj} + n_{gj}^{-si}}{\sum_l A_{gl} + n_{gl}^{-si}} \qquad \text{if } x_{si} = 0 \\
p(z_{si} = j|\mathbf{z}_{-si}, \mathbf{x}, \gamma_s = g) &= \frac{\alpha + n_{dj}^{-si}}{\sum_t \alpha + n_{dt}^{-si}} \qquad \text{if } x_{si} = 1
\end{aligned}
\tag{3.9}
$$

Here n_{gj} is the total number of words in group g assigned to label j and n_{dj} is the total number of words in document d assigned to topic j. The prior count is either A_{gj} (if $x_{si} = 0$) or α (if $x_{si} = 0$).

The final factor in Equation 3.7 is the probability of the switch x_{si} given the Bernoulli parameter π_s which is given by $p(x_{si}|\pi_s) = \pi_s^{x_{si}}(1 - \pi_s)^{(1-x_{si})}$.

Sampling Bernoulli parameters

The final sampling equation we need for MC-LDA is for the Bernoulli parameters $\boldsymbol{\pi}$. The Bernoulli parameter for sentence s, denoted π_s, is sampled from a Beta distribution with scalar parameters a and b. Since the Beta distribution is conjugate to the Bernoulli distribution, the probability of π_s given a, b, and the switch variables from sentence s is given by,

$$
p(\pi_s|\mathbf{x}, \boldsymbol{\pi}_{-s}, a, b) \sim \text{Beta}(a + \sum_i x_{si},\ b + N_s - \sum_i x_{si}) \tag{3.10}
$$

where N_s is the number of words in sentence s.

3.5 Experimental data sets

We constructed a data set of labeled sentences from scientific articles spanning three different domains: computational biology, machine learning, and psychology.

We created three corpora using articles from PLoS Computational Biology (PLOS), articles from the machine learning repository of arXiv (ARXIV), and articles from the psychology journal Judgment and Decision Making (JDM). PLOS contains articles from 2005 to 2011; ARXIV contains articles from 2006 to 2009; JDM contains articles from 2006 to 2012. We chose PLOS, ARXIV, and JDM because they spanned different scientific domains and because they were easily processed.

We cleaned the articles by removing any html, xml, or latex tags, replacing citations, numbers, and equations with special tokens, and splitting on sentence boundaries[6]. We then split each article into sections and kept only the abstracts and introductions. Our initial goal was to perform sentence classification on the full text of the articles. However, given the difficulty of predicting sentence function on full text [69, 28], we instead focused on the abstract and introduction as two sections that together display a high variation in sentence function.

We removed a small set of 25 stopwords (see Appendix C for a complete list) as well as any word that occurred only once. We did not perform any stemming or bigram detection. The resulting vocabulary contained $16,284$ unique word tokens.

We randomly selected 300 articles from each corpus to create a training set, 5 articles from each corpus to create a validity set, and 5 articles from each corpus to create a test set.

[6]We wrote our own script for splitting on sentence boundaries. We first split the text at every occurrence of a period, question mark, or exclamation mark. We then run a series of checks to determine if any splits should be undone. For example, if we split a piece of text at a period, and the text that follows the period is a number then we combine the text back together. This happens for instance with decimal numbers, e.g. 0.2. From spot checking the results of our algorithm, we were satisfied with its performance.

Thus, the total number of articles in the train, validity, and test sets were 900, 15, and 15 articles respectively. The total number of sentences in the train, validity, and test sets were $34,422$ sentences, 567 sentences, and 472 sentences respectively.

A number of individuals helped to provide labels for the sentences in the validity set and the test set. Details on the how annotation was performed and inter-annotator agreement scores can be found in Appendix B. We then used the labeled sentences in the validity set to construct informative priors for our proposed models SentenceLDA and Multicorpus SentenceLDA. Details on how these informative priors are constructed from the data are provided in Section 3.5.2.

3.5.1 Statistics of the labeled data

Recall the labels in our annotation scheme: AIM (a sentence that states the primary goal of the article), OWN (a sentence that describes the author's own work), CONTRAST (a sentence that states a limitation/weakness of past work), BASE (a sentence that states the author's own work is based on past work), MISC (any other sentence).

Table 3.4 shows a breakdown of the number of sentences with each label across the three domains. The validity set contains a total of 567 sentences. The test set contains a total of 472 sentences. The most frequent label is MISC making up 60% of the sentences in the validity set and 61% of the sentences in the test set. BASE sentences are quite rare making up only 2% of the validity set and 2% of the test set. Both AIM and CONTRAST are also relatively rare accounting for less than 10% of the validity and test sets respectively. Such label imbalance makes classification harder [9].

The top 5 most frequent words per label in the validity set and the test set are shown in Table 3.5 and Table 3.6 respectively. Many of the same words are in the top 5 for multiple

	AIM		OWN		CONTRAST		BASE		MISC	
	test	valid.	test	valid.	test	valid.	test	valid.	test	valid.
PLOS	9	8	38	65	6	17	2	8	92	126
ARXIV	12	11	58	59	2	6	5	1	72	104
JDM	11	12	30	37	10	3	3	0	122	110
Total	**32**	**31**	**126**	**161**	**18**	**26**	**10**	**9**	**286**	**340**
Percent	**7%**	**5%**	**27%**	**28%**	**4%**	**5%**	**2%**	**2%**	**61%**	**60%**

Table 3.4: The number of sentences with each label across the three data sets.

AIM	OWN	CONTRAST	BASE
we (21)	we (71)	ion (10)	were (3)
this (13)	this (33)	this (9)	we (3)
paper (8)	our (21)	channels (7)	al (3)
study (8)	section (20)	not (6)	et (3)
model (7)	model (16)	models (5)	this (3)

Table 3.5: Top 5 most frequent words per label in the **validity** set with the number of occurrences shown in parentheses.

labels: "we", "this", "model", and "us". Also note the presence of domain-specific words such as "channel", "face", "bound", and "self-control."

Both "paper" and "study" are unique to AIM. The words "et" and "al" show up in the top 5 for BASE in the validity set. The word "section" shows up in the top 5 for OWN in the validity set – this comes from sentences such as "In section 3, we show that..." The word "not" shows up in the top 5 for CONTRAST for both the validity and test set. It is clear from these two tables that a successful classifier must be able to ignore domain-dependent words – e.g. "neuron", "ion", "self-control" – and focus on the more discriminative words such as "paper", "study", "et al", and "section."

AIM	OWN	CONTRAST	BASE
this (18)	we (55)	not (7)	our (6)
we (15)	our (19)	use (6)	face (5)
paper (14)	bounds (13)	choices (5)	model (3)
model (6)	between (11)	been (5)	we (3)
study (6)	self-control (11)	however (5)	if (3)

Table 3.6: Top 5 most frequent words per label in the **test** set with the number of occurrences shown in parentheses.

3.5.2 Using labeled sentences to create informed priors

In this section, we describe how we used the labeled sentences in the validity set to create informative priors for SentenceLDA and Multicorpus SentenceLDA. This process is extremely important. Unlike a supervised classifier that has access to labeled data, our models have only the informative priors to bias the learned word distributions to reflect the semantics of each label. The performance of our models is closely tied to the quantity and quality of information encoded in these informative priors.

Step 1: Creating lists of indicative words

First, we reviewed each sentence in the validity set that was labeled with AIM, OWN, CONTRAST, or BASE – a total of 227 sentences. For each such sentence, the author of this thesis selected those words in the sentence that appeared to be semantically indicative of the label. This was a subjective process. Table 3.7 shows the top 10 most frequent indicative words per label in the validity set. We refer to these words as *indicator words*.

The total number of indicator words for OWN was 105, the total number of indicator words for AIM was 52, the total number of indicator words for CONTRAST was 66, and the total number of indicator words for BASE was 13. The full list of indicator words for each label is given in Appendix C.

AIM	OWN	CONTRAST	BASE
we	we	not	using
this	section	difficult	extend
paper	our	limited	relies
study	this	however	spirit
show	results	assume	previously
present	used	only	foundation
new	new	many	on
model	show	although	reuse
introduce	model	typically	similar
current	were	directly	refined

Table 3.7: Top 10 most frequent indicator words for each label.

Step 2: Constructing the hyper-parameter matrix B

The hyper-parameter matrix B has L rows (one for each label) and W columns (one for each word). The entry B_{li} is the prior count for word i in label l.

Let $m_l = [m_{l1}, \ldots, m_{lW}]$ be a vector of word counts where m_{li} is the number of times word i occurred in those sentences with label l in the validity set. If i is an indicator word for label l, then we set $B_{li} = m_{li}$. For non-indicator words, we consider two different approaches:

- We sum the word counts for all words that are *not* indicator words and distributed this total equally across all non-indicator words. For example, if there are 5 words in our vocabulary, $m_l = [10\ 0\ 0\ 3\ 5]$, and words 1 and 5 are indicator words for this label then the hyper-parameter vector would be [10 1 1 1 5]. The sum of the word counts for the non-indicator words (words 2, 3, and 4) is $0 + 0 + 3$. We distribute this total equally across the three non-indicator words giving a hyper-parameter count of 1 to each.

- We distribute a fixed count of 1 equally across all non-indicator words[7]. The hyper-parameter vector from the preceding scenario would now be $[10\ \frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\ 5]$.

[7]This can be generalized to distributing a fixed count k equally across all non-indicator words

We use the first approach for SentenceLDA where all words in the vocabulary must be assigned to one of the L labels. We use the second approach for Multicorpus SentenceLDA where many of the words in the vocabulary are absorbed by the corpus-specific topics leaving a smaller set of words to be explained by the L label clusters.

Through experimentation, we realized that it is important that each hyper-parameter vector $B_l = (B_{l1}, \ldots, B_{lW})$ sum to the same value. Each label has a different number of indicator words and a different number of sentences in the validity set. As a result, the sum of the hyper-parameter vectors B_l can differ markedly from one label to the next leading to unexpected results. Thus, once we have constructed the hyper-parameter matrix B we normalize the rows of B and scale by $\eta \in \{50, 100, 500\}$. We chose these values through experimentation.

Step 3: Constructing the hyper-parameter matrix A

Finally, we also construct the hyper-parameter matrix A which has G rows (one for each group) and L columns (one for each label). The entry A_{gl} is the prior count for label l in group g.

Our first step is to partition the sentences in the validity set into groups. For example, if we group sentences by section then we would have $G = 2$ groups corresponding to the abstract and the introduction.

For each group g, we computed the maximum likelihood estimate (MLE) of the Dirichlet hyper-parameters conditioned on the labeled sentences in the group. Unlike the hyper-parameter matrix B where we needed to manually enforce that indicator words had higher probability, we had no such constraints here and thus were free to simply estimate the Dirichlet hyper-parameters from the data. We used code provided by Minka et al. [45] to find the MLE estimates.

As before, once we estimated the Dirichlet hyper-parameter matrix A we normalized each row and scaled by $\alpha \in \{50, 100\}$. Our reasoning for normalizing and re-scaling was different than before – in this case, the estimated Dirichlet hyper-parameters were very small (often clustered around 1). Thus, we normalized and scaled so that we could control the strength of the Dirichlet priors.

Table 3.8 shows the Dirichlet hyper-parameters estimated when we split sentences by section ($G = 2$) and use $\alpha = 50$. The hyper-parameters mirror the distribution of labels in each group. Abstracts are shorter and tend to focus more on the author's own work being presented in the article – hence AIM and OWN have the highest prior counts in the first group. However, introductions are much longer and describe the background and context of the author's work. Usually only the last few sentences in the introduction state the aim of the paper and describe the author's work. Hence, the label MISC has the highest prior probability in the second group.

Group	AIM	OWN	CONT	BASE	MISC
g = 1 (abstract)	16.9	15.3	3.3	3.1	11.2
g=2 (introduction)	4.6	6.4	2.6	1.6	34.7

Table 3.8: Dirichlet hyper-parameters when grouping by section $G = 2$

We briefly describe the different groups used in our experiments:

- (*global*) All sentences are in one group. In this case, $G = 1$.
- (*section*) Sentences are grouped by section: abstract or introduction. In this case $G = 2$.
- (*position*) Sentences are grouped by relative position within the section. In this case $G = 4$. Sentences in the first quarter of any section are assigned to the first group, sentences in the second quarter of any section to the second group, and so on[8]

[8]We noticed a strong dependence between the position of a sentence in a section and its label. For

- (*sentence*) Each sentence is its own group. In this case, $G = S$.
- (*section-position*) Sentences are grouped by both their section (abstract or introduction) and position. In this case $G = 8$.

3.6 Experiments

In this section, we describe the details of our experiments including the training and testing process for SentenceLDA and Multicorpus SentenceLDA as well as the 5 supervised and semi-supervised baseline classifiers.

3.6.1 Inference and parameter estimation – training

Sent-LDA-W, Sent-LDA-S, and MC-LDA were trained on the set of 900 unlabeled scientific articles in the training set using the hyper-parameter matrices A and B estimated from the 15 articles in the validity set (as described earlier).

Model	Hyper-parameters	Random variables	Other	Collapsed
Sent-LDA-W	A, B	z_{si}	–	ϕ_l, θ_g
Sent-LDA-S	A, B	z_s	–	ϕ_l, θ_g
MC-LDA	T, A, B, α, β	z_{si}, x_{si}, π_s	a, b	ϕ_l, θ_g, ϕ_{ct}, θ_d

Table 3.9: Summary of the hyper-parameters and random variables for Sent-LDA-W, Sent-LDA-S, and MC-LDA

Table 3.9 provides a summary of the hyper-parameters and random variables for each of the models. For MC-LDA, T is the number of topics for each corpus; α is the hyper-parameter

example, a common pattern for introductions is to start with background sentences – i.e. MISC sentences – followed by an AIM sentence, followed by a group of OWN sentences elaborating on the aim. Using 4 groups was a simple way of approximating this pattern. However, this idea could be generalized and each section split into K groups where K is fixed or learned.

of the symmetric Dirichlet prior for the document-specific distributions θ_d; β is the hyper-parameter of the symmetric Dirichlet prior for the topic distributions ϕ_{ct} (where c is the corpus index and t is the topic index). In all of our experiments, we used $T = 5$ since each corpus consisted of only 300 articles. We set $\alpha = 0.001$ and $\beta = 0.001$ since smaller prior values allow the data to determine the sparsity of the learned multinomials. We also sampled a and b – the parameters of the Beta prior – from Gamma hyper-priors with scale and shape parameters $(1, 1)$ and $(15, 5)$ respectively.

For each model, we performed $5,000$ Gibbs sampling iterations where a single iteration consisted of sampling every random variable from its conditional distribution. This included sampling a and b using a slice sampler [48].

After $5,000$ iterations, samples were taken every 100 iterations for a total of 10 samples (i.e. $6,000$ iterations) from the posterior distribution. All results for Sent-LDA and MC-LDA are averaged over these 10 samples.

3.6.2 Inference and parameter estimation – testing

For Sent-LDA-W and Sent-LDA-S, a sample from the posterior distribution consists of an assignment of every word in the training set to one of L labels. Recall that for MC-LDA, a word can be assigned to one of L labels *or* one of T corpus-specific topics.

Given the assignment of each word to a label and the hyper-parameter matrices A and B, we compute the following point-estimates for Sent-LDA-W and Sent-LDA-S:

$$\hat{\phi}_{lw} = \frac{B_{lw} + n_{lw}}{\sum_{w'} B_{lw'} + n_{lw'}} \tag{3.11}$$

$$\hat{\theta}_{gl} = \frac{A_{gl} + n_{gl}}{\sum_{l'} A_{gl'} + n_{gl'}} \tag{3.12}$$

where n_{lw} is the total number of times w was assigned to label l. The semantics of n_{gl} depend on whether we are using Sent-LDA-W, in which case n_{gl} is the total number of **words** assigned to label l whose sentence's group assignment is g, or Sent-LDA-S, in which case n_{gl} is the total number of **sentences** assigned to label l in group g.

For MC-LDA, we use Equation 3.11 and Equation 3.12 to compute a point-estimate for the label distributions and the group distributions respectively. For the corpus-specific topics, we compute a similar point estimate:

$$\hat{\phi}_{ct,w} = \frac{\beta + n_{ct,w}}{\sum_{w'} \beta + n_{ct,w'}} \tag{3.13}$$

where $n_{ct,w}$ is the total number of times w was assigned to topic t in corpus c in the posterior sample.

For each model, we fix these point estimates at test time and sample only the label assignment variables for the words in the test set. We perform $1,000$ such Gibbs iterations and store the assignment of words to labels from the last iteration.

For each sentence, we then normalize these word assignment counts to compute the probability of each label in the sentence. For MC-LDA, any word in the test set assigned to one of the T corpus-specific topics is assumed to be domain-dependent and as such is not included in the word counts when making predictions.

3.6.3 Baseline classifiers

We trained 5 additional supervised and semi-supervised classifiers:

- Dirichlet multinomial regression [44] (DMR) is a variation of LDA that incorporates additional user-specified features beyond just the "bag-of-words." Similar to Sent-LDA and MC-LDA, DMR is trained on the 900 unlabeled articles in the training set and uses the informative prior matrix B to bias the learned word distributions so that they reflect the semantics of each label. For the details of our implementation, including the features we used, see Appendix D.

- We independently trained L binary support vector machines (SVM) – one for each label – using the LibLinear [16] implementation. All parameters were set to their default value except for the weight parameter for positive instances. The weight parameter for positive instances (which adjusts the penalty for misclassifying a positive instance) was determined using cross-validiation on the sentences in the validity set.

- We trained a naive Bayes classifier using a uniform smoothing constant of $\beta = 0.001$ when computing the probability of a word given a label and a uniform smoothing constant of $\alpha = 0.001$ when computing the prior probability of each label. We denote this classifier as NB-unif.

- We trained a second Naive Bayes classifier using the informative priors A and B to perform the smoothing. We experimented with $\alpha \in \{50, 100\}$ and $\beta \in \{50, 100, 500\}$ (recall that α and β are the scaling constants for the hyper-parameter matrices). The configuration with the highest macro-F_1 score occurred for $\beta = 500$ and $\alpha = 50$ or $\alpha = 100$. We denote this classifier as NB-inform (where "inform" stands for informative prior).

- We used the UniverSVM [64] implementation to train a transductive SVM – a type

of semi-supervised SVM that uses both labeled and unlabeled data. We denote this classifier as TransSVM. TransSVM was trained using both the 900 unlabeled articles in the training set as well as the labeled sentences in the validity set.

Table 3.10 shows the data used to train each of the classifiers. The first column **Validity set** refers to the 567 labeled sentences in the validity set. The second column **Training set** refers to the 900 unlabeled articles in the training set (which consisted of approximately $34,000$ unlabeled sentences). The third column **Inform. priors** refers to the informative prior matrices A and B constructed from the validity set.

	Validity set	**Training set**	**Inform. priors**
Sent-LDA-S		x	x
Sent-LDA-S		x	x
MC-LDA		x	x
DMR		x	x
TransSVM	x	x	
SVM	x		
NB-unif	x		
NB-inform	x		x

Table 3.10: Table showing the data used to train each of the classifiers.

All classifiers were evaluated on the labeled sentences in the test set. Similar to Sent-LDA and MC-LDA, the output of DMR is an assignment of the words in a sentence to the L labels. For each sentence, we normalize these word assignments to compute the probability of each label in the sentence. The naive Bayes classifiers NB-unif and NB-inform also output the probability of each label in the sentence.

The output of SVM and TransSVM is the distance of each sentence from a separating hyperplane (i.e. decision boundary) for each of the L independently trained classifiers. A support vector machine uses labeled training data to learn a hyperplane that separates positive instances of the class from negative instances of the class [8]. At test time, the support vector machine returns the distance from this hyperplane for each of the instances

(i.e. sentences) in the test set. A large distance from the separating hyperplane indicates the support vector machine is more confident about the sentence's class whereas a short distance (in which the sentence is on or close to the separating hyperplane) indicates less confidence about the sentence's class.

3.7 Evaluation metrics

We present results for three types of tasks: label-pivoted binary predictions, label-pivoted rankings, and document-pivoted binary predictions. The distinction between "label-pivoted" and "document-pivoted" is a matter of focus. For label-pivoted tasks, we are interested in predicting *for each label* the relevant sentences. For document-pivoted tasks, we are interested in predicting *for each sentence* the relevant labels. For more information on label-pivoted versus document-pivoted see [62].

For each of these options (label-pivoted or document-pivoted) we may choose to perform a binary prediction task – where we are interested in making hard assignments – or a ranking task – where we are interested in returning a ranked list of instances. For more information on metrics used for binary prediction tasks versus metrics used for ranking tasks see chapter 8 of [40].

3.7.1 Label-pivoted binary predictions

For each classifier and each label, we first rank the sentences in the test set. The sentences are ranked either by the probability of the label in the sentence or by the distance of the sentence from the decision boundary (depending on the classifier). We then threshold the ranked list to produce a binary classification. We use the marginal probability of each label in the validity set to determine the threshold. For example, 60% of sentences in the validity

set are assigned MISC. Thus, at test time we classify the top 60% of the ranked sentences as positive instances of MISC and the remaining 40% as negative. This is known as *proportional thresholding* [62] and is an alternative to fixed thresholding where the top K sentences are classified as positive instances for fixed K. Note that this binary prediction task is done independently for each of the L labels.

Given the binary predictions, we compute the F_1 score for each label. The F_1 score is a commonly used measure of text classification performance [40]. The F_1 score is the harmonic mean of the precision P and the recall R given by the equation,

$$F_1 = \frac{2PR}{P+R}$$

The F_1 score is between 0 and 1 where a higher F_1 score indicates better performance. We also report the macro-averaged F_1 score and the micro-averaged F_1 score. The macro-averaged F_1 score is the average F_1 score across all L labels. The micro-averaged F_1 score is computed by first pooling the true positive, false positive, and false negatives across all L labels and then computing the F_1 score. Macro-averaging gives equal weight to each label whereas micro-averaging gives more weight to those labels with more instances in the test set.

3.7.2 Document-pivoted binary predictions

For each classifier and each sentence in the test set, we first rank the labels (again either by the probability of the label in the sentence or the distance of the sentence from the decision boundary of each of the L classifiers corresponding to the L labels). Since sentences can have only one label, we take the top ranked label as the binary prediction. We then compute

the accuracy over all sentences in the test set, i.e. we compute the proportion of sentences for which the predicted label was the true label.

3.7.3 Label-pivoted rankings

For each classifier and each label, we again rank the sentences in the test set (either by probability or distance from the decision boundary depending on the classifier). We then compute the precision and recall for the top K ranked sentences for $K \in \{1, 2, \ldots, S\}$ where S is the number of sentences in the test set. We can then plot the precision against the recall for all values of K to produce a precision-recall curve. Note that the precision-recall curve will be jagged since each additional sentence either improves both the precision and the recall (resulting in an increase along the x- and y-axis) or decreases the precision (resulting in a decrease along the y-axis). To smooth this curve, the *interpolated* precision can be computed for fixed recall values. The interpolated precision at a recall of R is the highest precision for any recall $R' \geq R$. We plot the interpolated precision for fixed recall values of $R \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Again, see chapter 8 in [40] for more information on precision-recall curves and the interpolated precision.

3.8 Results comparing SentenceLDA and Multicorpus SentenceLDA

3.8.1 Learned word distributions

Table 3.11 shows the top 10 words with highest probability for each label in the annotation scheme learned by the best configuration of Sent-LDA-S. A configuration corresponds to a specification of the grouping along with the value of the scaling parameters α and β for the

informative priors. The *best* configuration is the configuration that resulted in the highest Macro-F_1 score on the test set (where ties were broken using the Micro-F_1 scores)[9]. Any indicator words are shown in bold. Table 3.12 and Table 3.13 show the top 10 words with the highest probability for each label learned by the best configuration of Sent-LDA-W and MC-LDA respectively. Table 3.14 also shows the top 5 words with highest probability for each of the corpus-specific topics learned by the best configuration of MC-LDA.

AIM	OWN	CONT	BASE	MISC
NUMBER	CITATION	SYMBOL	SYMBOL	CITATION
than	**this**	we	percent	we
when	**we**	CITATION	we	this
CITATION	not	this	if	model
they	more	algorithm	this	protein
more	decision	learning	any	been
participants	NUMBER	data	set	data
was	their	problem	probabilitiy	between
were	people	our	class	at
one	they	section	stochastic	our

Table 3.11: Top 10 words with the highest probability for each label learned by the best configuration of Sent-LDA-S. Indicator words are in bold.

AIM	OWN	CONT	BASE	MISC
we	**we**	CITATION	many	data
NUMBER	**our**	SYMBOL	been	when
this	**section**	this	proteins	models
use	**results**	been	learning	their
different	**show**	will	systems	information
between	also	not	problem	its
algorithm	this	example	applications	methods
two	finally	one	decision	but
not	experiment	if	disease	time
at	**method**	each	often	function

Table 3.12: Top 10 words with the highest probability for each label learned by the best configuration of Sent-LDA-W. Indicator words are in bold.

The top 10 words learned by Sent-LDA-S for the labels CONTRAST and BASE (Table 3.11) do

[9]In the next section, we give details on the best configuration for each of the three models

AIM	OWN	CONT	BASE	MISC
this	**we**	**not**	both	CITATION
two	**our**	other	first	one
also	**results**	**may**	**using**	been
models	**model**	**however**	more	this
data	**this**	**only**	**similar**	at
use	**show**	each	based	their
but	between	all	here	between
how	provide	than	general	many
time	**used**	**while**	result	some
model	different	under	predictions	they

Table 3.13: Top 10 words with the highest probability for each label learned by the best configuration of MC-LDA. Indicator words are in bold.

not correspond to the semantic meaning of the labels. In particular, the words with highest probability for the label BASE include domain-specific words, e.g. "percent", "probability", and "stochastic". Similarly, the words with highest probability for the label CONTRAST are general words such as "we", "problem", "learning" and "data". Only two indicator words are in the top 10 for the label OWN and only one indicator word is in the top 10 for the label AIM.

In Table 3.12, we see that the top 10 words learned by Sent-LDA-W for the label OWN include significantly more indicator words. In addition, the words "experiment" and "finally" (which were not indicator words) were identified from the training data as having high probability for the label OWN. We see a similar though less dramatic improvement for the labels AIM and CONTRAST. However, the top 10 words learned for the label BASE still do not reflect the semantic meaning of BASE. In particular, many domain-dependent words such as "proteins", "decision", and "disease" appear in the top 10.

In Table 3.12, we see a marked improvement for the labels CONTRAST and BASE. In particular, the top 10 highest probability words for CONTRAST include the indicator words "not", "may", "only", and "however". Similarly, the top 10 highest probability words for

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
PLOS	CITATION	CITATION	CITATION	CITATION	CITATION
	network	sequence	protein	cell	models
	genes	genome	proteins	cells	neurons
	gene	sequences	structure	model	model
	networks	dna	binding	signaling	neural

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
JDM	recognition	NUMBER	decision	NUMBER	NUMBER
	NUMBER	judgments	CITATION	risk	effect
	heuristic	people	making	people	than
	when	participants	information	decision	choice
	cue	when	research	their	participants

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
ARXIV	learning	learning	decision	SYMBOL	SYMBOL
	data	data	CITATION	we	percent
	SYMBOL	algorithm	making	algorithm	mt
	we	algorithms	information	percent	def
	algorithm	section	research	problem	endobj

Table 3.14: Top 5 words with the highest probability for each corpus-specific topic learned by the best configuration of MC-LDA

BASE include the indicator words "using" and "similar". Furthermore, the word "based" was identified from the training data as having high probability for BASE. From Table 3.14, we see that the corpus-specific topics absorbed many of the domain-dependent words in the training set leaving the remaining domain-independent words to be explained by the labels.

3.8.2 Label-pivoted binary predictions for SentenceLDA and Multicorpus SentenceLDA

Tables 3.15–3.17 show the F_1 scores for Sent-LDA-S, Sent-LDA-W, and MC-LDA respectively for the label-pivoted binary prediction task.

Each row corresponds to a particular grouping and shows the configuration of the parameters

α and β that resulted in the best macro-F_1 performance on the test set for that group. Any ties were broken using the micro-F_1 score. Recall that $\alpha \in \{50, 100\}$ and $\beta \in \{50, 100, 500\}$. When α is empty (denoted using a vertical line) this means the best performance resulted when we did not use an informative prior for the group distributions over labels but instead used a symmetric prior over the labels. We bold the grouping that had the overall highest macro-F_1 score (again breaking ties with the micro-F_1 performance). We term this grouping (and the corresponding value of α and β) the *best configuration*.

All of the F_1, macro-F_1, and micro-F_1 scores reported have been averaged over the 10 samples taken from the Gibbs chain during training. We report the standard deviation of the macro-F_1 and micro-F_1 scores for the best configuration in the respective paragraphs discussing each model.

Group	α	β	AIM	OWN	CONT	BASE	MISC	Macro-F_1	Micro-F_1
global	50	100	0.16	0.37	0.08	0.02	0.70	0.27	0.54
section	100	100	0.16	0.39	0.08	0.02	0.70	0.27	0.54
position	100	100	0.16	0.40	0.07	0.02	0.71	0.27	0.55
sentence	**100**	**100**	**0.26**	**0.36**	**0.18**	**0.00**	**0.67**	**0.30**	**0.53**
sec-pos	$--$	100	0.15	0.38	0.08	0.02	0.70	0.27	0.54

Table 3.15: F_1 scores for Sent-LDA-S

Group	α	β	AIM	OWN	CONT	BASE	MISC	Macro-F_1	Micro-F_1
global	50	100	0.29	0.37	0.06	0.00	0.61	0.27	0.49
section	100	500	0.22	0.39	0.10	0.03	0.62	0.27	0.50
position	100	500	0.28	0.46	0.06	0.00	0.67	0.29	0.55
sentence	50	100	0.29	0.40	0.05	0.00	0.66	0.28	0.53
sec-pos	**50**	**100**	**0.30**	**0.56**	**0.09**	**0.00**	**0.66**	**0.32**	**0.57**

Table 3.16: F_1 scores for Sent-LDA-W

The best configuration for Sent-LDA-S was (*sentence*, $\alpha = 100, \beta = 100$) which had a macro-$F_1$ score of $0.30 \pm 5 \times 10^{-4}$ and a micro-F_1 score of $0.53 \pm 2 \times 10^{-4}$. Sent-LDA-S consistently performed better for $\beta = 100$ (a weaker informative prior over words) than for $\beta = 500$ (a stronger informative prior over words). Note that Sent-LDA-S has the lowest macro-F_1 and

Group	α	β	AIM	OWN	CONT	BASE	MISC	Macro-F_1	Micro-F_1
global	50	500	0.35	0.38	0.12	0.05	0.69	0.32	0.55
section	50	100	0.22	0.41	0.05	0.00	0.65	0.27	0.52
position	50	50	0.25	0.49	0.07	0.00	0.71	0.31	0.58
sentence	**50**	**50**	**0.45**	**0.40**	**0.11**	**0.00**	**0.71**	**0.33**	**0.57**
sec-pos	50	100	0.25	0.50	0.13	0.00	0.74	0.32	0.60

Table 3.17: F_1 scores for MC-LDA

micro-F_1 score out of all three models despite having the highest F_1 score for CONTRAST (0.18).

The best configuration for Sent-LDA-W was (*section-position*, $\alpha = 50, \beta = 100$) which had a macro-F_1 score of $0.32 \pm 6.4 \times 10^{-3}$ and a micro-F_1 score of $0.57 \pm 2.8 \times 10^{-3}$. It is clear that the added flexibility of allowing each word to have its own label assignment variable improves performance for the labels AIM and OWN. Note that the best configuration of Sent-LDA-W had a higher macro-F_1 score and micro-F_1 score than the best configuration of Sent-LDA-S.

The best configuration for MC-LDA was (*sentence*, $\alpha = 50, \beta = 50$) which had a macro-F_1 score of $0.33 \pm 1.6 \times 10^{-2}$ and a micro-F_1 score of $0.57 \pm 1.2 \times 10^{-2}$. Interestingly, for almost all groupings MC-LDA performed best with $\alpha = 50$. MC-LDA achieves the same or higher macro-F_1 score for all groupings. Figure 3.4 compares the macro-F_1 scores for the best configuration of all three models Sent-LDA-S, Sent-LDA-W, and MC-LDA across all 5 groupings.

3.8.3 Document-pivoted binary predictions for SentenceLDA and Multicorpus SentenceLDA

Tables 3.18–3.20 show the accuracy for Sent-LDA-S, Sent-LDA-W, and MC-LDA respectively for the document-pivoted binary prediction task. Recall that the accuracy is the proportion of documents whose predicted label was the true label. We use an asterisk (*) to indicate that

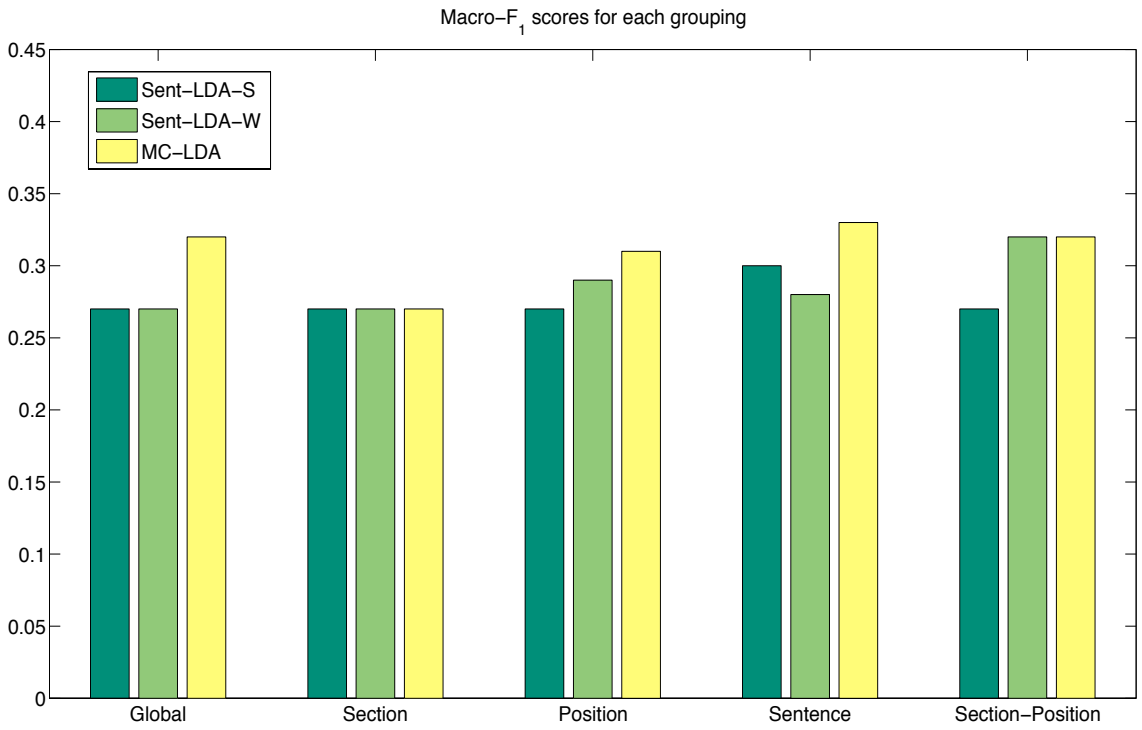

Figure 3.4: Macro-F_1 scores for the best configuration of all three models Sent-LDA-S, Sent-LDA-W, and MC-LDA for all 5 groupings

the accuracy was constant across the different values for α and/or β. We report the mean and standard deviation of the accuracy across the 10 training samples. The configuration with the highest accuracy is shown in bold.

Again, we see that MC-LDA has a higher accuracy than both Sent-LDA-S and Sent-LDA-W. MC-LDA correctly classified 62% of the sentences in the test set as opposed to 60% of the sentences correctly classified by Sent-LDA-W and 47% of sentences correctly classified by Sent-LDA-S. The best configuration of MC-LDA was no longer with the grouping *sentence* but with the grouping *section-position*. The opposite occurred with Sent-LDA-W: the best configuration was no longer with a grouping of *section-position* but *sentence*.

Group	α	β	Accuracy
global	100	500	$0.27 \pm 1.1 \times 10^{-3}$
section	50	100	$0.33 \pm 6.7 \times 10^{-4}$
position	50	500	$0.27 \pm 9.0 \times 10^{-4}$
sentence	$*$	$*$	$\mathbf{0.47} \pm 0.00$
sec-pos	$*$	500	$0.28 \pm 1.3 \times 10^{-3}$

Table 3.18: Accuracy for Sent-LDA-S

Group	α	β	Accuracy
global	100	500	$0.41 \pm 5.0 \times 10^{-3}$
section	100	500	$0.40 \pm 5.0 \times 10^{-3}$
position	50	100	$0.41 \pm 3.9 \times 10^{-3}$
sentence	$*$	$*$	$\mathbf{0.60} \pm 0.00$
sec-pos	$*$	500	$0.28 \pm 1.3 \times 10^{-3}$

Table 3.19: Accuracy for Sent-LDA-W

Group	α	β	Accuracy
global	$*$	500	$0.61 \pm 7.9 \times 10^{-4}$
section	50	100	$0.56 \pm 8.7 \times 10^{-3}$
position	100	100	$0.61 \pm 8.3 \times 10^{-4}$
sentence	50	50	$0.61 \pm 1.6 \times 10^{-2}$
sec-pos	**50**	**100**	$\mathbf{0.62} \pm 9.0 \times 10^{-3}$

Table 3.20: Accuracy for MC-LDA

3.9 Results with other classifiers

In this section, we compare the performance of *the best configuration* of Sent-LDA-S, Sent-LDA-W, and MC-LDA to 5 other supervised and semi-supervised classifiers for the three tasks: label-pivoted binary prediction, label-pivoted rankings, and document-pivoted binary predictions.

3.9.1 Label-pivoted binary predictions

Classifier	AIM	OWN	CONT	BASE	MISC	Macro-F_1	Micro-F_1
Sent-LDA-S	0.26	0.36	**0.18**	0.00	0.67	0.30	0.53
Sent-LDA-W	0.30	**0.56**	0.09	0.00	0.66	0.32	**0.57**
MC-LDA	**0.45**	0.40	0.11	0.00	0.71	**0.33**	**0.57**
DMR	0.11	0.22	0.00	0.00	0.64	0.20	0.46
TransSVM	0.00	0.52	0.00	0.00	0.56	0.22	0.48
SVM	0.00	0.08	0.05	**0.21**	**0.78**	0.23	0.50
NB-unif	0.04	0.42	0.05	0.00	0.57	0.21	0.46
NB-inform	0.04	0.43	0.05	0.00	0.57	0.22	0.47
Teufel et al.	0.11	0.83	0.17	0.07	(0.22, 0.44)	–	–

Table 3.21: F_1 scores for Sent-LDA, MC-LDA, and 5 other supervised and semi-supervised classifiers for the label-pivoted binary prediction task

Table 3.21 shows the F_1 scores for the best configurations of Sent-LDA-S, Sent-LDA-W, and MC-LDA along with the F_1 scores for the 5 baseline classifiers for the label-pivoted binary prediction task[10].

DMR had the lowest macro-F_1 score of all the baseline classifiers failing to find strong signal in the features provided. The TransSVM predicted only the most frequent labels OWN and MISC. Interestingly, the SVM had the highest F_1 score for the label BASE (0.21) out of any of the classifiers. The NB-inform and NB-unif classifiers had similar performance with

[10]Both DMR and NB-inform use the informative prior matrices A and B and, as such, require tuning of the scaling parameters α and β. We report the F_1 performance for the configuration of α and β that resulted in the highest macro-F_1 score (where ties were broken using the micro-F_1 score).

NB-inform having a slightly higher F_1 score for the label AIM.

We show in bold the classifier with the highest F_1 score for each label. Overall, we see that the probabilistic models introduced in this chapter (Sent-LDA and MC-LDA) have the highest F_1 score for three out of the five labels. In particular, MC-LDA has the highest macro-F_1 score and micro-F_1 score out of any of the classifiers.

The last row of Table 3.21 shows the reported F_1 scores from Teufel et al. (see [69] Table 8 on page 432) who used a naive Bayes classifier trained on a collection of 80 labeled linguistics articles. We show results for the baseline algorithm trained only on word features[11]. The F_1 scores of Teufel et al. are not directly comparable to our own since Teufel et al. predicted on the full text of the articles and focused on articles from computational linguistics[12]. However, these scores serve to highlight the difficulty of the sentence classification task.

3.9.2 Document-pivoted binary predictions

Classifier	**Accuracy**
Sent-LDA-S	0.47 ± 0.00
Sent-LDA-W	0.60 ± 0.00
MC-LDA	$0.62 \pm 9.0 \times 10^{-3}$
DMR	0.02
TransSVM	0.17
SVM	0.04
NB-unif	0.49
NB-inform	0.53

Table 3.22: Accuracy for Sent-LDA, MC-LDA, and 5 other supervised and semi-supervised baseline classifiers for the document-pivoted binary prediction task

Table 3.22 shows the accuracy for the *best configuration* of Sent-LDA-S, Sent-LDA-W, and

[11]We show the F_1 scores for the labels TEXTUAL and OTHER together in the column for the label MISCELLANOUS since we collapsed these labels together to create the label MISCELLANEOUS

[12]Because Teufel et al. performed classification on the entire text of the article, their distribution over labels was quite different. As reported in [69], the breakdown of sentences in their labeled data set was 2% AIM, 67% OWN, 5% CONTRAST, 2% BASE, and 24% MISC.

MC-LDA along with the 5 baseline classifiers. Once again DMR is the lowest performing classifier with an accuracy of only 0.02. The SVM also has a low accuracy of 0.04. NB-unif and NB-inform have the highest accuracy of the baseline classifiers and a higher accuracy than Sent-LDA-S. Of all the classifiers, Sent-LDA-W and MC-LDA have the highest accuracy of 0.60 and 0.62 respectively.

3.9.3 Label-pivoted rankings

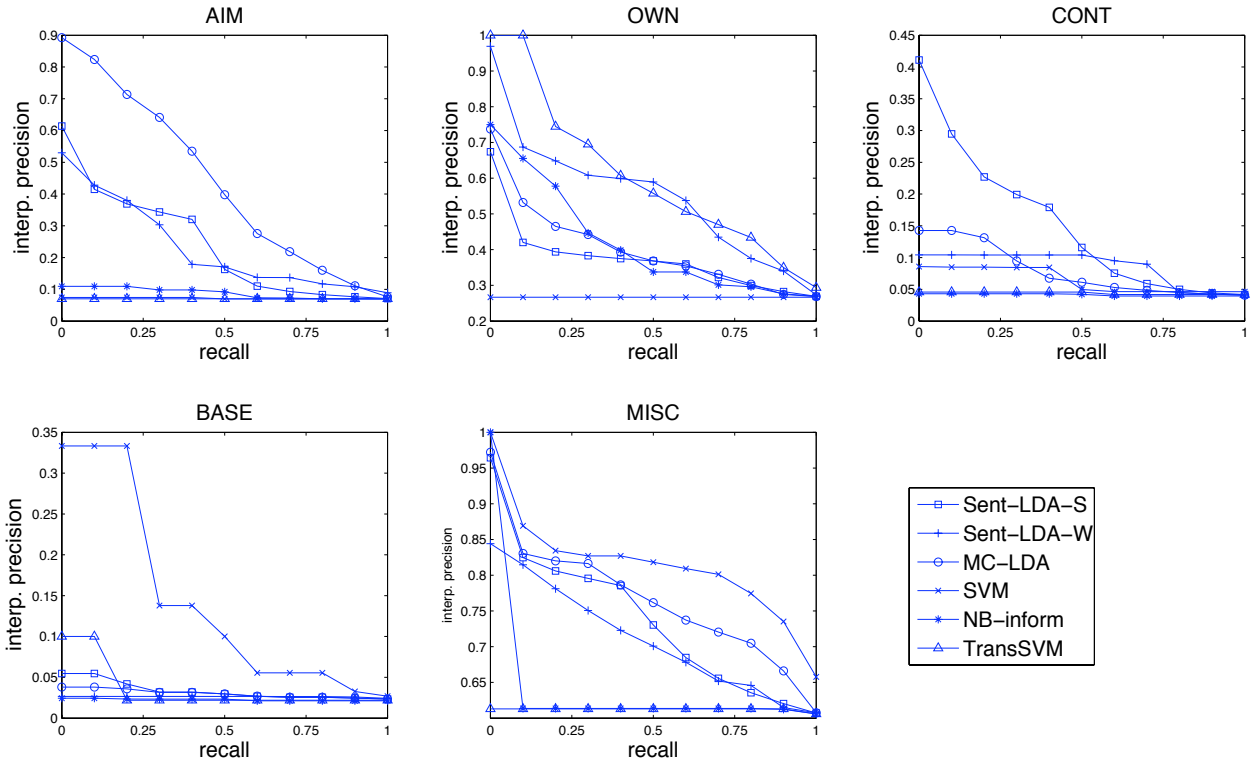

Figure 3.5: Interpolated precision for fixed recall for all labels in the annotation scheme

Figure 3.5 shows the interpolated precision computed for fixed recall values for each of the labels in the annotation scheme. We do not include NB-unif or DMR in the plots for clarity's sake since neither classifier was the highest performing classifier for any of the labels.

For AIM, MC-LDA outperforms all other classifiers. For the label OWN, Sent-LDA-W and TransSVM have the highest precision. For CONT, Sent-LDA-S has the highest precision for

recall less than 0.6 after which Sent-LDA-W has the highest precision (until a recall of 0.8). Similar to the binary prediction task, SVM has the highest precision for BASE. Finally, for the label MISC, SVM has the highest precision followed by MC-LDA, Sent-LDA-S and Sent-LDA-W. Overall, Sent-LDA-S, Sent-LDA-W, and MC-LDA either outperform or are competitive with the baseline classifiers except for the label BASE.

3.9.4 Illustrated examples

Finally, we show illustrated examples of the predictions of Sent-LDA-S, Sent-LDA-W, MC-LDA, and NB-inform on articles from the test set in Appendix E. We show three articles: one from each corpus. One interesting observation from these predictions is that many of the classifiers predict the label CONTRAST for sentences that contain overall negative statements as opposed to only those sentences that contain negative or critical statements *with respect to past work*. As an example, consider the sentence

> Most such trials have yielded higher failure rates in the treatment group than in controls receiving conventional therapy.

Note that this sentence states a negative finding of medical trials as opposed to a critical statement by the author about past research. This sentence was incorrectly, but understandably, labeled as CONTRAST by S-LDA-S and MC-LDA.

3.10 Summary and contributions

In this chapter, we presented two new statistical models for sentence classification in scientific articles: SentenceLDA and Multicorpus SentenceLDA. SentenceLDA adapts and extends latent Dirichlet allocation to perform sentence classification. One of the key characteristics of

SentenceLDA is its generalization of a "document" to groups of sentences that exhibit similar distributions over sentence functions. Multicorpus SentenceLDA extends SentenceLDA by incorporating a second word-generating mechanism to explain the presence of domain-dependent words in a sentence. Multicorpus SentenceLDA utilizes articles from different scientific domains to cluster domain-dependent words into corpus-specific topics and sorts the remaining domain-independent words into those that are truly indicative of a sentence's function as opposed to those that are simply noise.

We also created a data set of sentences from scientific articles spanning three different domains: computational biology, machine learning, and psychology. The sentences in the abstract and introduction of the articles have been labeled with sentence functions derived from the Argumentative Zones annotation scheme [69]. We discussed a method for mining this data to create informative priors and have included in Appendix C a complete list of indicator words for each label in the annotation scheme.

We used this labeled data to evaluate the performance of SentenceLDA and Multicorpus SentenceLDA along with 5 other supervised and semi-supervised baseline classifiers: Dirichlet multinomial regression [44], support vector machines, two naive Bayes classifiers, and a transductive SVM [13]. We compared the performance of each classifier on three tasks: label-pivoted binary predictions, label-pivoted rankings, and document-pivoted binary predictions [62, 40]. We showed that the probabilistic models introduced in this chapter, particularly Multicorpus SentenceLDA, either outperform or are competitive with the baseline classifiers.

In summary:

- We presented two new statistical models, SentenceLDA and Multicorpus SentenceLDA, for sentence classification in scientific articles.
  - SentenceLDA adapts and extends latent Dirichlet allocation to perform sentence

classification. One of the key characteristics of SentenceLDA is its generalization of a "document" to "groups" of sentences that exhibit similar distributions over sentence functions.

  - Multicorpus SentenceLDA extends SentenceLDA by incorporating a second mechanism to explain the presence of domain-dependent words in a sentence.

- We created a data set of sentences from scientific articles that span three different scientific domains. The sentences in the abstract and introduction of each article have been labeled with sentence functions derived from the Argumentative Zones annotation scheme [69].

- We discussed a method for mining this labeled data to create informative priors and have included in Appendix C a complete list of indicator words for each label in our annotation scheme.

- We used this labeled data to evaluate the performance of SentenceLDA and Multicorpus SentenceLDA. We also compared the performance of SentenceLDA and Multicorpus SentenceLDA to five other supervised and semi-supervised classifiers: Dirichlet multinomial regression [44], support vector machines, two naive Bayes classifiers, and a transductive SVM [13].

- We analyzed the performance of each classifier for a variety of tasks including label-pivoted binary predictions, label-pivoted rankings, and document-pivoted binary predictions. Both SentenceLDA and Multicorpus SentenceLDA are competitive with, or outperform, the baseline classifiers.

3.11 Future directions

One area for further development is a method for better constructing the hyper-parameter matrices A and B for Multicorpus SentenceLDA (MC-LDA). In particular, the distribution over words (for each label) and over labels (for each group) inferred by using **all** of the words in the sentences from the validity set is different than the distribution over words and over labels inferred if we used **only** the domain-independent words. A better procedure for MC-LDA would be to first identify and remove domain-dependent words[13] and then to estimate the hyper-parameter matrices A and B. Recall that MC-LDA performed best when the scaling parameter for A was $\alpha = 50$ (a weaker prior) as opposed to $\alpha = 100$ (a stronger prior). We hypothesize that this is due to the fact that the distribution over labels if we only consider domain-independent words is more uniform than the distribution over labels if we consider all words. Similarly, when constructing B we distributed a fixed prior count of 1 to all non-informative words because we found experimentally that the alternative method produced a prior that was too general for MC-LDA and resulted in poorer performance. However, instead of a fixed prior count of 1, using a grid search over the value of the fixed prior count could potentially improve the performance of MC-LDA.

We hypothesize that Multicorpus SentenceLDA would perform better than SentenceLDA when used to predict sentence function for sentences from different domains, e.g. computational linguistics, economics, or chemistry. The distribution over words (for each label) learned by MC-LDA contains very few domain-dependent words. In contrast, the distribution over words (for each label) learned by Sent-LDA-S and Sent-LDA-W contain a larger number of domain-dependent words (see Tables 3.11, 3.12, and 3.13) such as "protein", "stochastic", "participants", and "disease". It would be interesting to compare the performance of the trained classifiers presented in this chapter (including the supervised and semi-supervised

[13]This would require a pre-processing step to identify words that are domain-dependent with high probability

classifers) for predicting sentence function in other domains. For Sent-LDA-S, Sent-LDA-W, and MC-LDA a "trained classifier" would consist of point estimates of the distributions over words for each label ϕ_l and the distribution over labels for each group θ_g analogous to the testing process discussed in Section 3.6.2. The only potential drawback of MC-LDA is that it would require at least two different domains in order to achieve the best performance.

Another interesting extension to the work presented in this chapter is the use of phrases as features to predict sentence function in addition to word features. Many of the indicator words we extracted from the labeled sentences in the validity set originated from repeated phrases. For example, the indicator words "this" and "section" for the label OWN often appeared together in the labeled data as the phrase, "in this section." Similarly, the indicator words "show" and "how" for the label AIM often appeared together in the labeled data as the phrase "we show how." In Table 3.5, the words "et" and "al" appear in the top 5 most frequent words for the label BASE in the validity set. In our current models, a sentence only needs to have the word "this" to have high probability under the label OWN. However, with phrases we could enforce that a sentence must contain both "this" and "section" (or "show" and "how") to have high probability under a label. This could presumably improve the performance of our models by reducing the false positive detection rate.

Chapter 4

Summarizing document collections using concept graphs

4.1 Introduction

In this chapter we present a generative probabilistic model for learning concept graphs from text. We define a *concept graph* as a rooted, directed graph where the nodes represent concepts – i.e. semantically-related collections of words – and the edges represent relationships between concepts. Concept graphs are useful for summarizing document collections and providing a visualization of the semantic content and structure of large document sets - a task that is difficult to accomplish using only keyword search. An example of a concept graph is Wikipedia's category graph[1]. Figure 4.1 shows a small portion of the Wikipedia category graph rooted at the category MACHINE_LEARNING[2]. From the graph we can quickly infer that the collection of machine learning articles in Wikipedia focuses primarily on evolutionary algorithms and Markov models with less emphasis on other aspects of machine learning

[1] http://en.wikipedia.org/wiki/Category:Main_topic_classifications
[2] As of May 5, 2009

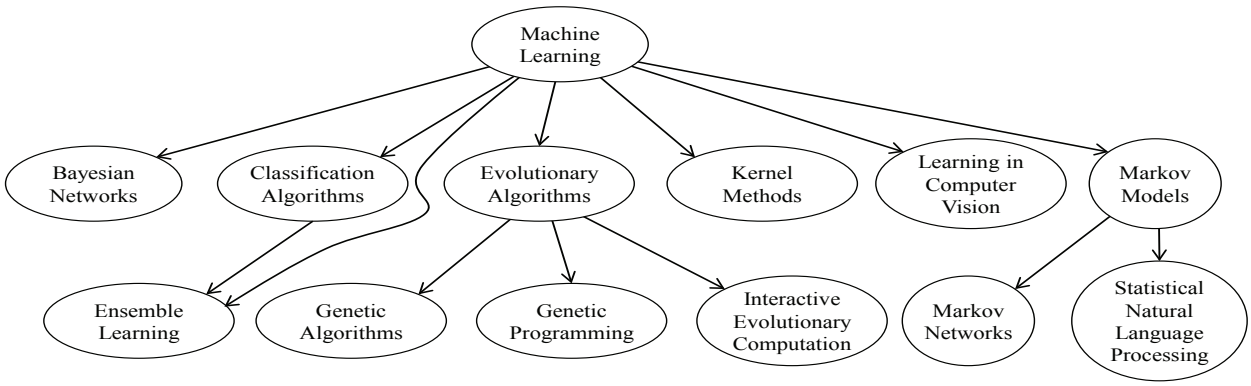

Figure 4.1: A portion of the Wikipedia category subgraph rooted at the node MACHINE_LEARNING

such as Bayesian networks and kernel methods.

The problem we address in this chapter is that of learning a concept graph given a collection of documents where (optionally) we may have concept labels for the documents and an initial graph structure. In the latter scenario, the task is to identify additional concepts in the corpus that are not reflected in the graph or additional relationships between concepts in the corpus (via the co-occurrence of concepts in documents) that are not reflected in the graph. This is particularly suited for document collections like Wikipedia where the set of articles is changing at such a fast rate that an automatic method for updating the concept graph may be preferable to manual editing or re-learning the hierarchy from scratch.

We first introduce the stick-breaking distribution and show how it can be used as a prior over graph structures. We then introduce our generative model and explain how it can be adapted for the case where we have an initial graph structure. We derive collapsed Gibbs' sampling equations for our model and present a series of experiments on simulated and real text data. We compare the performance of our model with hierarchical latent Dirichlet allocation [3] and the hierarchical Pachinko allocation model [43, 34]. We conclude with a discussion of the merits and limitations of our approach.

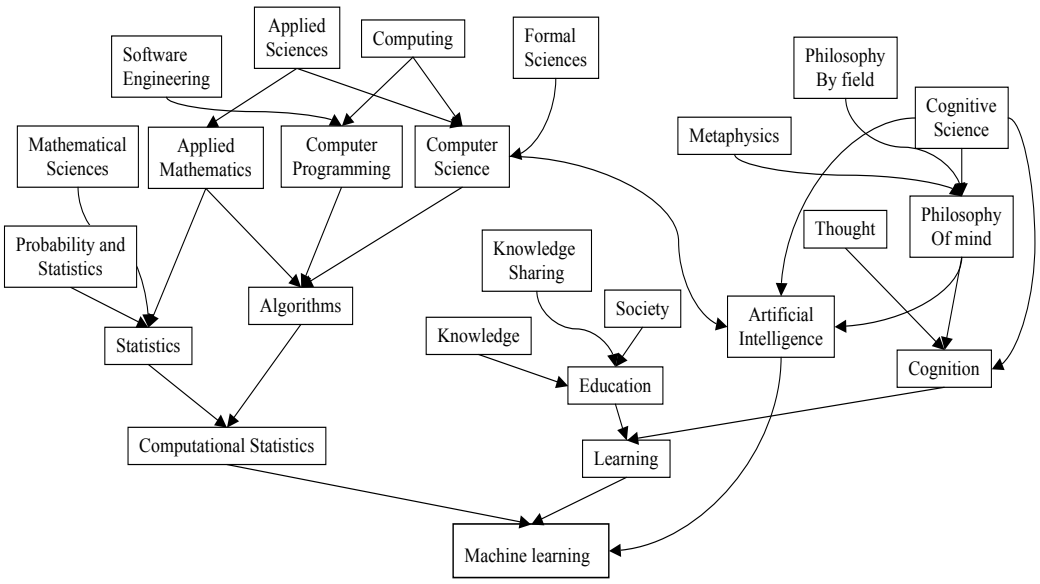

Figure 4.2: A portion of the Wikipedia category supergraph for the node MACHINE_LEARNING

4.2 Related work

The foundation of our approach is latent Dirichlet allocation (LDA) [5]. LDA is a probabilistic model for automatically identifying topics within a document collection where a topic is a probability distribution over words. The standard LDA model does not include any notion of relationships, or dependence, between topics. See Section 2.2 for a more-detailed introduction to LDA.

In contrast, methods such as the hierarchical topic model (hLDA) [3] learn a set of topics in the form of a tree structure. The restriction to tree structures however is not well suited for large document collections like Wikipedia. Figure 4.2 gives an example of the highly non-tree like nature of the Wikipedia category graph.

The hierarchical Pachinko allocation model (hPAM) [43] is able to learn a set of topics arranged in a fixed-sized graph with a nonparametric version introduced in [34].

The model we propose in this chapter is a simpler alternative to hPAM and nonparametric hPAM that can achieve the same flexibility (i.e. learning arbitrary directed acyclic graphs over a possibly infinite number of nodes) within a simpler probabilistic framework. In addition, our model provides a formal mechanism for utilizing labeled data and existing concept graph structures.

Other methods for creating concept graphs include the use of techniques such as hierarchical clustering, pattern mining and formal concept analysis to construct ontologies from document collections [18, 6, 12]. Our approach differs in that we utilize a probabilistic framework which enables us (for example) to make inferences about concepts and documents.

Our primary novel contribution is the introduction of a flexible probabilistic framework for learning general graph structures from text that is capable of utilizing both unlabeled documents as well as labeled documents and prior knowledge in the form of existing graph structures.

4.3 Stick-breaking distributions

The probabilistic model presented in this chapter relies heavily on the stick-breaking distribution. Thus, we begin with the definition of a stick-breaking distribution. For more information on stick-breaking distributions, see [47].

Stick-breaking distributions, which we denote as $\mathcal{P}(\cdot)$, are discrete probability distributions of the form:

$$\mathcal{P}(\cdot) = \sum_{j=1}^{\infty} \pi_j \delta_{x_j}(\cdot) \quad \text{where} \quad \sum_{j=1}^{\infty} \pi_j = 1,\ 0 \leq \pi_j \leq 1,$$

and $\delta_{x_j}(\cdot)$ is the delta function centered at x_j. The $\{x_j\}$ are sampled independently from

a base distribution H (where H is often assumed to be continuous). The stick-breaking weights π_j have the form

$$\pi_1 = v_1, \quad \pi_j = v_j \prod_{k=1}^{j-1}(1 - v_k) \quad \text{for } j = 2, 3, \ldots, \infty \tag{4.1}$$

where the v_j are independent Beta(α_j, β_j) random variables.

Stick-breaking distributions derive their name from the analogy of repeatedly breaking the remainder of a unit-length stick at a randomly chosen breakpoint. Given a stick of unit length, $v_1 \sim \text{Beta}(\alpha_1, \beta_1)$ is the proportion of the stick that is broken off. The remainder of the stick has length $(1 - v_1)$. In the next iteration, $v_2 \sim \text{Beta}(\alpha_2, \beta_2)$ is the proportion of the stick broken off. The length of the portion that is broken off is $v_2(1 - v_1)$ whereas the length of the remainder of the stick is $(1 - v_2)(1 - v_1)$. We repeat this process: in the jth iteration, v_j is sampled from Beta(α_j, β_j) and the length of the portion of the stick broken off is $v_j \prod_{k=1}^{j-1}(1-v_k)$ whereas the length of the remainder of the stick is $(1-v_j)\prod_{k=1}^{j-1}(1-v_k)$.

The probability of sampling a particular cluster from $\mathcal{P}(\cdot)$ given the sequences $\{x_j\}$ (drawn from H) and $\{v_j\}$ (drawn from a Beta distribution) is *not* equal to the probability of sampling the same cluster given a permutation of the sequences $\{x_{\sigma(j)}\}$ and $\{v_{\sigma(j)}\}$ (where σ is a permutation of the integers). This can be seen in Equation 4.1 where the probability of sampling x_j depends upon the value of the $j-1$ preceding Beta random variables $\{v_1, v_2, \ldots, v_{j-1}\}$. If we fix x_j and permute every other element, then the probability of sampling x_j changes: it is now determined by the Beta random variables $\{v_{\sigma(1)}, v_{\sigma(2)}, \ldots, v_{\sigma(j-1)}\}$. Thus, if we wish to learn a stick-breaking distribution from data, we must take this into account, and learn not only the value of the Beta random variables but their ordering as well.

4.4 Prior over graphs

In this section, we construct a prior distribution over graph structures using stick-breaking distributions. We construct this prior by specifying a distribution at each node (denoted as $\mathcal{P}_t$ for node t) that governs the probability of transitioning from the given node to another node in the graph. Graphically, this is equivalent to specifying a distribution over the outgoing edges at a given node.

We have two requirements when defining $\mathcal{P}_t$. First, making a new transition – i.e. creating a new edge – must have non-zero probability. In Figure 4.1 it is clear that from MACHINE_LEARNING we should be able to transition to any of its children. However, we may discover evidence for transitioning directly to a leaf node such as STAT_NATURAL_LANG_PROC (e.g. if we observe new articles related to statistical natural language processing that do not use Markov models). In this case, there must be non-zero probability for creating a new edge from MACHINE_LEARNING to STAT_NATURAL_LANG_PROC.

Second, creating a new node and transitioning to this new node must also have non-zero probability. For example, we may observe articles related to a new topic Bioinformatics. In this case, we want to add a new node to the graph (BIOINFORMATICS) and assign some probability of transitioning to it from other nodes.

With these two requirements we can now provide a formal definition for $\mathcal{P}_t$. We begin with an initial graph structure G_0 with $t = 1 \dots T$ nodes. For each node t we define a *feasible set* $\mathcal{F}_t$ as the collection of nodes to which t can transition. The feasible set may contain the children of node t or possible child nodes of node t (as discussed above). In general, $\mathcal{F}_t$ is some subset of the nodes in G_0. We add a special node called the exit node to $\mathcal{F}_t$. If we sample the exit node then we exit from the graph instead of transitioning forward. We define $\mathcal{P}_t$ as a stick-breaking distribution over the finite set of nodes $\mathcal{F}_t$ where the remaining probability mass is assigned to an infinite set of new nodes (nodes that exist but have not

yet been observed). The exact form of $\mathcal{P}_t$ is shown below.

$$\mathcal{P}_t(\cdot) = \sum_{j=1}^{|\mathcal{F}_t|} \pi_{tj} \delta_{f_{tj}}(\cdot) \; + \sum_{j=|\mathcal{F}_t|+1}^{\infty} \pi_{tj} \delta_{x_{tj}}(\cdot)$$

The first $|\mathcal{F}_t|$ elements of the stick-breaking distribution are the feasible nodes $f_{tj} \in \mathcal{F}_t$. The remaining elements are unidentifiable nodes that have yet to be observed (denoted as x_{tj} for simplicity). Note that if G_0 is empty, i.e. $T = 0$, then we create a single node called the "root" node and all probability mass for the root node's transition distribution is placed on the unobserved nodes $\{x_{tj}\}$.

This is not yet a working definition unless we explicitly state which nodes are in the set $\mathcal{F}_t$. Our model does not in general assume any specific form for $\mathcal{F}_t$. Instead, the user is free to define it as they like. In our experiments, we assign each node to a unique depth and then define $\mathcal{F}_t$ as any node at the next lower depth. If there is an existing graph structure G_0, we assign each node to a depth by topologically sorting the nodes, and grouping nodes based on an equivalence relation where a is equivalent to b if a and b can be swapped in the ordering without violating the topological property.

The choice of $\mathcal{F}_t$ determines the type of graph structures that can be learned. For our choice of $\mathcal{F}_t$, edges that traverse multiple depths are not allowed and edges between nodes in the same depth are not allowed. This prevents cycles from forming and allows inference to be performed in a timely manner. More generally, one could extend the definition of $\mathcal{F}_t$ to include any node that is at a strictly lower depth.

Finally, we must learn the order of the feasible nodes in $\mathcal{F}_t$. Note that despite the order, the elements of $\mathcal{F}_t$ always occur before the infinite set of new nodes in the stick-breaking permutation. We use a Metropolis-Hastings sampler proposed by [54] to learn the permutation of feasible nodes with the highest likelihood given the data.

1. For node $t \in \{1, \ldots, \infty\}$
   i. Sample stick-break weights $\{v_{tj}\}|\alpha, \beta \sim \text{Beta}(\alpha, \beta)$
   ii. Sample word distribution $\phi_t|\eta \sim \text{Dirichlet}(\eta)$
2. For document $d \in \{1, 2, \ldots D\}$
   i. Sample path $p_d \sim \{\mathcal{P}_t\}_{t=1}^{\infty}$
   ii. Sample a distribution over levels in the path $\tau_d|a, b \sim \text{Beta}(a,b)$
   iii. For word $i \in \{1, 2, \ldots, N_d\}$

   Sample level $l_{d,i} \sim \text{TruncatedDiscrete}(\tau_d)$

   Generate word $x_{d,i}|\{p_d, l_{d,i}, \Phi\} \sim \text{Multinomial}(\phi_{p_d[l_{di}]})$

Figure 4.3: Generative process for GraphLDA

4.5 Generative process

Figure 4.3 shows the generative process for our proposed model, which we refer to as GraphLDA. As discussed earlier, each node t is associated with a stick-breaking distribution $\mathcal{P}_t$ that governs the probability of transitioning to a node in $\mathcal{F}_t$. In addition, we associate with each node a probability distribution ϕ_t over words in the fashion of a topic model. The only exceptions are the "exit nodes" which have no corresponding topic and thus cannot generate words.

A two-stage process is used to generate a document d. First, starting at the root node, a path through the graph is sampled from the stick-breaking distributions $\{\mathcal{P}_t\}_{t=1}^{\infty}$ until an exit node is drawn. We denote this path by $p_d = (p_{d1}, p_{d2}, \ldots, p_{d\lambda_d})$. We denote the length of the path by λ_d. Note that the first node p_{d1} is always the root node and the last node $p_{d\lambda_d}$ is always an exit node. Except for the root node, the ith node in the path p_{di} is sampled from the stick-breaking distribution of the $i-1$st node in the path, $\mathcal{P}_{p_{d,i-1}}(\cdot)$.

We also associate with each document a discrete distribution over the levels, i.e. nodes, in

D	Number of documents
V	Number of words in the vocabulary
$\mathcal{F}_t$	The set of feasible nodes for node t
$\mathcal{P}_t$	The stick-breaking distribution over $\mathcal{F}_t$ for node t
$\{v_{tj}\}$	Beta random variables that parameterize $\mathcal{P}_t$
ϕ_t	Probability distribution over words for node t
$p_d = (p_{d1}, \ldots, p_{d\lambda_d})$	Path through the graph for document d
λ_d	The length of the path p_d
τ_d	Discrete distribution over levels in path for document d
x_{di}	The ith word in document d
l_{di}	The level in the path p_d associated with x_{di}
$p_d[l_{di}]$	The node at level l_{di} in the path
$N_{(x,y)}$	The number of paths that traverse the edge (x, y) in the graph
$N_{(x,>y)}$	The number of paths that go from x to a node with a strictly higher position than y in x's stick-breaking permutation
$N_{(x,\geq y)}$	$N_{(x,y)} + N_{(x,>y)}$

Table 4.1: Notation for GraphLDA

the path p_d. We use this distribution to disperse the words in the document across the nodes in the path. Instead of using a multinomial distribution over levels, we choose instead a parametric smooth form in order to constrain the distributions to have the same functional form across documents (in contrast to the relatively unconstrained multinomial), but to also allow the parameters of the distribution to be document-specific. We use a geometric distribution parameterized by probability of success $\tau_d \in (0, 1]$ and we place a Beta(a, b) prior over the parameters τ_d, for $a, b \in \mathbb{R}_{>0}$.

Finally, for each word x_{di} a level in the path $l_{di} \in \{1, 2, \ldots, \lambda_d - 1\}$ is sampled from the truncated distribution parameterized by τ_d. Note that l_{di} cannot equal λ_d since words cannot be assigned to an exit node. A word is then generated by the topic at level l_{di} of the path

p_d. We use indexing notation $p_d[l_{di}]$ to represent this node. If the word x_{di} has level $l_{di} = 1$ then the word is generated by the topic at the *last* node on the path (not including the exit node) and successive levels correspond to earlier nodes in the path. Thus if $l_{di} = \lambda_d - 1$, then x_{di} is generated by the root node. In the case of labeled documents, this matches our intuition that a majority of words in the document should be assigned to the concept label itself assuming the document labels are not noisy[3]. Table 4.1 summarizes this, and future, notation.

4.6 Inference and parameter estimation

The above generative model gives rise to the following joint distribution:

$$\begin{aligned} &p(\mathbf{x}, \mathbf{l}, \mathbf{p}, \boldsymbol{\tau}, \mathbf{v}, \boldsymbol{\phi} | \alpha, \beta, a, b, \eta) \\ &= \prod_{di} p(x_{di} | p_d, l_{di}, \boldsymbol{\phi}) p(l_{di} | \tau_d) \prod_d p(\tau_d | a, b) p(p_d | \mathbf{v}) \prod_t p(\mathbf{v}_t | \alpha, \beta) p(\phi_t | \eta) \end{aligned} \tag{4.2}$$

We marginalize out the topic distributions $\boldsymbol{\phi}$ and the stick-breaking weights $\mathbf{v}$. We use a collapsed Gibbs sampler [23] to infer the paths $\mathbf{p}$, the level assignments $\mathbf{l}$, and the level distribution parameters $\boldsymbol{\tau}$. We use Metropolis-Hastings [54] to learn the permutation of feasible nodes for each node.

Of the five hyper-parameters in this model, inference is sensitive to the value of β and η so we place an exponential prior on both and use a Metropolis-Hastings sampler to learn the best setting.

Note that we always condition on the hyper-parameters, $\{\eta, a, b, \alpha, \beta\}$ even though this is

[3] and stop words have been removed

not explicitly shown in equations and figures for clarity's sake.

4.6.1 Sampling paths

For each document, we must sample a path p_d conditioned on all other paths $\mathbf{p}_{-d}$, the level variables $\mathbf{l}$, and the word tokens $\mathbf{x}$. We only consider paths whose length is greater than or equal to the maximum level of the words in the document.

$$p(p_d|\mathbf{x}, \mathbf{l}, \mathbf{p}_{-d}, \boldsymbol{\tau}) \propto p(\mathbf{x}_d|\mathbf{x}_{-d}, \mathbf{l}, \mathbf{p}) \ \cdot \ p(p_d|\mathbf{p}_{-d}) \tag{4.3}$$

The first factor in Equation 4.3 is the probability of all words in the document given the path p_d. We compute this probability by marginalizing over the topic distributions ϕ_t:

$$p(\mathbf{x}_d|\mathbf{x}_{-d}, \mathbf{l}, \mathbf{p}) = \prod_{l=1}^{\lambda_d - 1} \left(\prod_{v=1}^{V} \frac{\Gamma(\eta + N_{p_d[l],v})}{\Gamma(\eta + N^{-d}_{p_d[l],v})} \right) \ * \ \frac{\Gamma(V\eta + \sum_v N^{-d}_{p_d[l],v})}{\Gamma(V\eta + \sum_v N_{p_d[l],v})}$$

where $N_{p_d[l],v}$ stands for the number of times in the corpus word type v has been assigned to node $p_d[l]$. The superscript $-d$ means we do not count the words in document d.

The second factor in Equation 4.3 is the conditional probability of the path p_d given all other paths $\mathbf{p}_{-d}$. We present the sampling equation under the assumption that there is a maximum number of nodes M allowed at each depth, and then let M to go to infinity.

We first consider the probability of sampling a single edge in the path from a node x to one of its feasible nodes $\{y_1, y_2, \ldots, y_M\}$ where the node y_1 has the first position in the stick-breaking permutation, y_2 has the second position, y_3 the third, and so on.

We denote the number of paths that have gone from x to y_i as $N_{(x,y_i)}$. We denote the number of paths that have gone from x to a node with a strictly higher position in the stick-breaking distribution than y_i as $N_{(x,>y_i)}$. That is, $N_{(x,>y_i)} = \sum_{k=i+1}^{M} N_{(x,y_k)}$. Extending this notation

we denote the sum $N_{(x,y_i)} + N_{(x,>y_i)}$ as $N_{(x,\geq y_i)}$. Then, the probability of selecting node y_i is given by:

$$p(x \to y_i \mid \mathbf{p}_{-d}) = \frac{\alpha + N_{(x,y_i)}}{\alpha + \beta + N_{(x,\geq y_i)}} \prod_{r=1}^{i-1} \frac{\beta + N_{(x,>y_r)}}{\alpha + \beta + N_{(x,\geq y_r)}} \qquad \text{for } i = 1 \ldots M \tag{4.4}$$

The probability of selecting y_i is the probability of *not* selecting any earlier node (which corresponds to the product over r) times the probability of selecting y_i. This equation has a form similar to the prior probability of selecting a cluster shown in Equation 4.1.

If y_m is the last node with a nonzero count $N_{(x,y_m)}$ and $m << M$, it is convenient to compute the probability of transitioning to y_i, for $i \leq m$, and the probability of transitioning to a node higher than y_m. Since y_m is the last node with nonzero edge count, $N_{(x,y_i)}$ and $N_{(x,\geq y_i)}$ for $i > m$ equals zero. Thus, the probability of transitioning to a node higher than y_m is given by

$$\sum_{k=m+1}^{M} p(x \to y_k | \mathbf{p}_{-d}) = \Delta \left[1 - \frac{\beta}{\alpha + \beta}^{M-m} \right] \tag{4.5}$$

where $\Delta = \prod_{r=1}^{m} \frac{\beta + N_{(x,>y_r)}}{\alpha + \beta + N_{(x,\geq y_r)}}$. The derivation of Equation 4.4 and Equation 4.5 is given in Appendix A. As M goes to infinity, the probability of sampling a node higher than y_m becomes,

$$\sum_{k=m+1}^{\infty} p(x \to y_k | \mathbf{p}_{-d}) = \Delta \tag{4.6}$$

Now that we have computed the probability of a single edge, we can compute the probability

of an entire path p_d:

$$p(p_d|\mathbf{p}_{-d}) = \prod_{j=1}^{\lambda_d - 1} p(p_{dj} \rightarrow p_{d,j+1}|\mathbf{p}_{-d})$$

Sampling a new path for document d requires enumerating all paths in the graph. To achieve this, we perform a depth-first traversal of the graph (starting at the root node), keeping track of the paths and path probabilities found. The running time is $O(E)$ for a single document, where E is the number of edges in the graph. Since, in our experiments, the graph is fully connected from one depth to the next, the running time for a single document becomes $O(DM^2)$ where D is the depth of the graph, and M is the average number of nodes per depth.

4.6.2 Sampling levels

For x_{di}, the ith word in the dth document, we must sample a level $l_{di} \in \{1, 2, \ldots, \lambda_d - 1\}$ conditioned on all other levels $\mathbf{l}_{-di}$, the document paths $\mathbf{p}$, the level parameters $\boldsymbol{\tau}$, and the word tokens $\mathbf{x}$.

$$p(l_{di}|\mathbf{x}, \mathbf{l}_{-di}, \mathbf{p}, \boldsymbol{\tau}) = p(x_{di}|\mathbf{x}_{-di}, \mathbf{l}, \mathbf{p}) \cdot p(l_{di}|\mathbf{l}_{-di}, \boldsymbol{\tau}) \tag{4.7}$$

The first factor in Equation 4.7 is the probability of word type x_{di} given the topic at node $p_d[l_{di}]$. This can be computed by marginalizing over the topic distribution at node $p_d[l_{di}]$.

$$p(x_{di}|\mathbf{x}_{-di}, \mathbf{l}, \mathbf{p}) = \frac{\eta + N^{-di}_{p_d[l_{di}],x_{di}}}{V\eta + \sum_v N^{-di}_{p_d[l_{di}],v}}$$

where again $N^{-di}_{p_d[l_{di}],v}$ is the number of times word type v has been assigned to node $p_d[l_{di}]$ not counting the current word x_{di}.

The second factor in Equation 4.7 is the probability of the level l_{di} given the geometric distribution over levels parameterized by τ_d.

$$\begin{aligned} p(l_{di}|\mathbf{l}_{-di}, \boldsymbol{\tau}) &= \frac{(1-\tau_d)^{l_{di}-1}\ \tau_d}{1-(1-\tau_d)^{\lambda_d-1}} \\ &\propto (1-\tau_d)^{l_{di}-1}\ \tau_d \end{aligned}$$

Since levels are drawn from a *truncated* Geometric distribution, the probability of the level is normalized by the cumulative distribution function (cdf) of the geometric distribution evaluated at $\lambda_d - 1$.

4.6.3 Sampling τ variables

Finally, we must sample the level distribution τ_d conditioned on the rest of the level parameters $\boldsymbol{\tau}_{-d}$, the level variables $\mathbf{l}$, the document paths $\mathbf{p}$, and the word tokens $\mathbf{x}$.

$$\begin{aligned} p(\tau_d|\mathbf{x}, \mathbf{l}, \mathbf{p}, \boldsymbol{\tau}_{-d}) &= p(\mathbf{l}_d|\tau_d) \cdot p(\tau_d|a,b) \\ &= \left(\prod_{i=1}^{N_d} \frac{(1-\tau_d)^{l_{di}-1}\ \tau_d}{(1-(1-\tau_d)^{\lambda_d-1})}\right) * \left(\frac{\tau_d^{a-1}\ (1-\tau_d)^{b-1}}{\mathbf{B}(a,b)}\right) \\ &\propto \frac{(1-\tau_d)^{\sum_i l_{di}-N_d+b-1} \cdot \tau_d^{N_d+a-1}}{(1-(1-\tau_d)^{\lambda_d-1})^{N_d}} \end{aligned} \tag{4.8}$$

Due to the normalization constant $(1-(1-\tau_d)^{\lambda_d+1})$, Equation 4.8 is not a recognizable probability distribution and we must use rejection sampling. Since the first term in Equation 4.8 is always less than or equal to 1, the sampling distribution is dominated by a Beta(a,b) distribution. According to the rejection sampling algorithm, we sample a candidate value for

τ_d from Beta(a, b) and either accept with probability $\prod_{i=1}^{N_d} \frac{(1-\tau_d)^{l_{di}}\ \tau_d}{(1-(1-\tau_d)^{\lambda_d})}$ or reject and sample again.

4.6.4 Metropolis Hastings for stick-breaking permutations

In addition to the Gibbs sampling, we employ a Metropolis Hastings sampler presented in [54] to mix over stick-breaking permutations. Consider a node x with feasible nodes $\{y_1, y_2, \ldots, y_M\}$. We sample two feasible nodes y_i and y_j from a uniform distribution[4]. Assume y_i comes before y_j in the stick-breaking distribution. Then the probability of swapping the position of nodes y_i and y_j is given by

$$\min\left\{1,\ \prod_{k=0}^{N_{(x,y_i)}-1} \frac{\alpha+\beta+N^*_{(x,>y_i)}+k}{\alpha+\beta+N_{(x,>y_j)}+k} \cdot \prod_{k=0}^{N_{(x,y_j)}-1} \frac{\alpha+\beta+N_{(x,>y_j)}+k}{\alpha+\beta+N^*_{(x,>y_i)}+k}\right\}$$

where $N^*_{(x,>y_i)} = N_{(x,>y_i)} - N_{(x,y_j)}$. See [54] for a full derivation. After every new path assignment, we propose one swap for each node in the graph.

4.7 Experiments

In this section, we present experiments performed on both simulated and real text data. We compare the performance of GraphLDA against hierarchical latent Dirichlet allocation (hLDA) and the hierarchical Pachinko allocation model (hPAM).

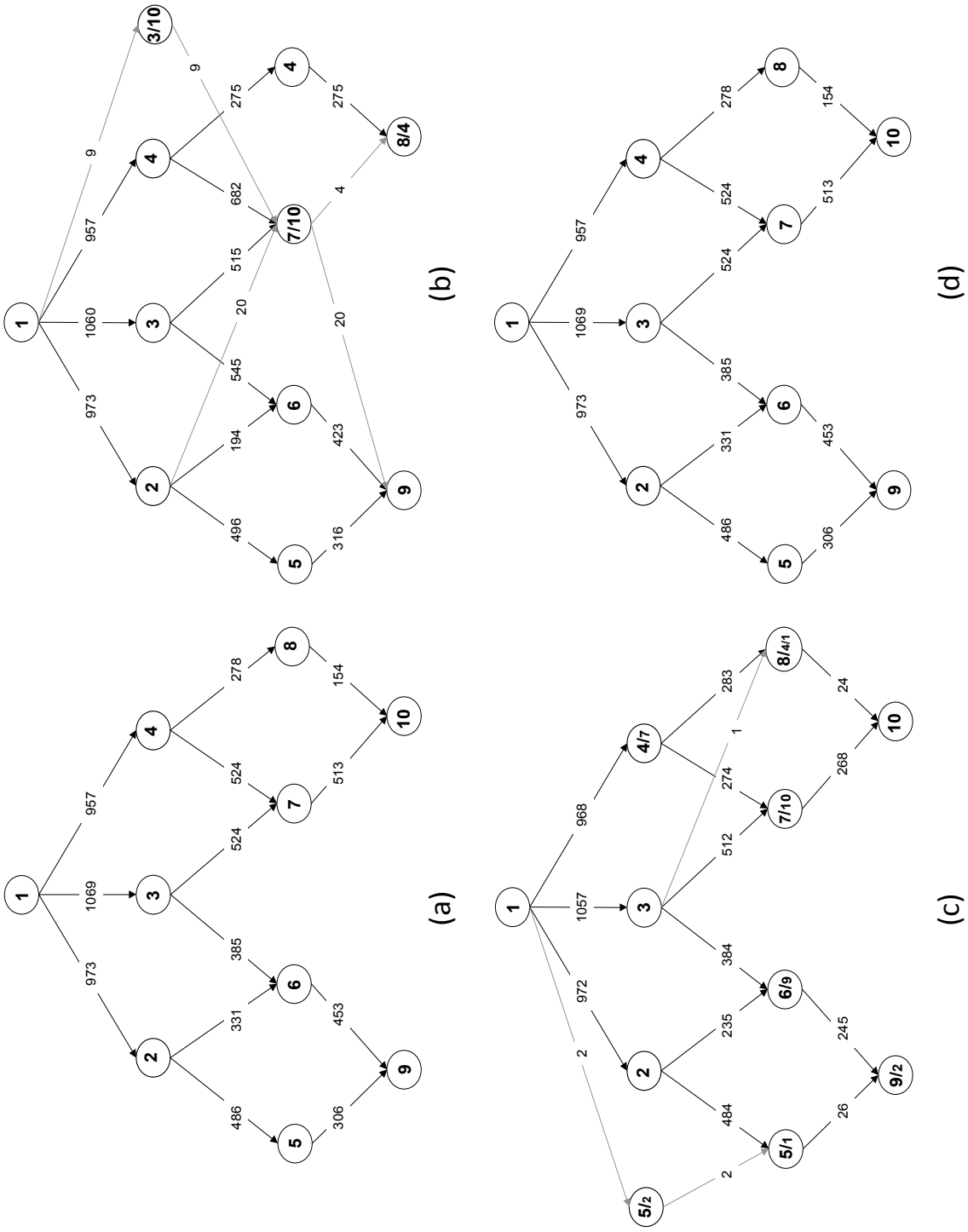

Figure 4.4: Learning graph structures from simulated data: (a) shows the original simulated graph (b) the learned graph structure with 0 labeled documents (c) the learned graph structure with 250 labeled documents (d) the learned graph structure with all 4000 labeled documents.

4.7.1 Simulated text data

In this section, we illustrate how the performance of GraphLDA improves as the fraction of labeled data increases. Figure 4.4(a) shows a simulated concept graph with 10 nodes drawn according to the generative process in Figure 4.3. Topic distributions were drawn from a symmetric Dirichlet prior with scalar parameter $\eta = .025$, the stick-breaking weights $\mathbf{v}$ were drawn from a Beta$(10, 10)$ distribution, and the level distribution parameters $\boldsymbol{\tau}$ were drawn from a Beta$(2, 5)$ distribution. The vocabulary size was $1,000$ words and we generated $4,000$ documents with 250 words each. Each edge in the graph is labeled with the number of paths that traverse it.

The stick-breaking weights for each node $\{\mathbf{v}_t\}$ were drawn from a Beta$(10, 10)$ distribution. Thus, the expected value $\mathrm{E}[v_{tj}]$ is 0.5 and the variance $\mathrm{Var}(v_{tj})$ is 0.5. The expected stick-breaking probabilities π_{tj} (which are computed from the Beta random variables according to Equation 4.1) are $\mathrm{E}[\pi_{tj}] = 0.5^j$. This means that the probability of sampling the first node according to the stick-breaking permutation is 0.5, the second node is 0.25, the third is 0.125, and so on.

Figure 4.4 (b) shows the learned graph structure with 0 labeled documents and $4,000$ unlabeled documents. The Gibbs sampler was initialized to only a root node. We label each edge with the number of paths that traverse the edge. We label each node based upon the similarity of the learned topic at the node to the topics of the original graph structure. With no labeled data, the sampler is unable to recover the relationship between concepts 8 and 10 (due to the relatively small number of documents that contain words from both concepts).

Figure 4.4 (c) shows the learned graph structure with 250 labeled documents and $3,750$ unlabeled documents. The Gibbs sampler was initialized with the correct placement of

[4] In [54] feasible nodes are sampled from the prior probability distribution. However for small values of α and β this can result in extremely slow mixing.

nodes to levels. The sampler does not observe the edge structure of the graph nor the correct number of nodes at each level (i.e. the sampler may add additional nodes). With 250 labeled documents, the sampler is able to learn the correct placement of both nodes 8 and 10 although the topic still contain some noise. With $4,000$ labeled documents (Figure 4.4 (d)) the original graph structure is learned back perfectly.

4.7.2 Comparison with baseline models

In this section, we compare the performance of GraphLDA to hPAM and hLDA on a set of 518 machine-learning articles taken from Wikipedia. The input to each model is only the article text. All models are restricted to learning a three-level hierarchical structure. For both GraphLDA and hPAM, the number of nodes at each level was set to 25. For GraphLDA, the parameters were fixed at $\alpha = 1$, $a = 1$ and $b = 1$. The parameters β and η were initialized to 1 and .001 respectively and optimized using a Metropolis Hastings sampler. We used the MALLET toolkit implementation of hPAM[5] and hLDA [42]. For hPAM, we used different settings for the topic hyper-parameter $\eta = (.001, .01, .1)$. For hLDA we set $\eta = .1$ and experimented with $\gamma = (.1, 1, 10)$ where γ is the smoothing parameter for the Chinese restaurant process and $\alpha = (.1, 1, 10)$ where α is the smoothing over levels in the graph.

All models were run for $9,000$ iterations to ensure burn-in and samples were taken every 100 iterations thereafter, for a total of $10,000$ iterations. The performance of each model was evaluated on a hold-out set consisting of 20% of the articles using both empirical likelihood and the left-to-right evaluation algorithm (see Sections 4.1 and 4.5 of [74]) which are measures of generalization to unseen data. For both GraphLDA and hLDA we use the distribution over paths that was learned during training to compute the per-word log likelihood. For hPAM we compute the MLE estimate of the Dirichlet hyperparameters for both the distribution

[5]MALLET implements the exit node version of hPAM

over super-topics and the distributions over sub-topics from the training documents.

Table 4.2 shows the per-word log-likelihood for each model averaged over the ten samples. GraphLDA is competitive when computing the empirical log likelihood. We speculate that GraphLDA's lower performance in terms of left-to-right log-likelihood is due to our choice of the geometric distribution over levels (and our choice to position the geometric distribution at the last node of the path).

Model	**Parameters**	**Empirical LL**	**Left-to-Right LL**
GraphLDA	MH opt.	-7.10 ± .003	-7.13 ± .009
hPAM	$\eta = .1$	-7.36 ± .013	-6.11 ± .007
	$\eta = .01$	-7.33 ± .012	-6.47 ± .012
	$\eta = .001$	-7.38 ± .006	-6.71 ± .013
hLDA	$\gamma = .1,\ \alpha = .1$	-7.10 ± .004	-6.82 ± .007
	$\gamma = .1,\ \alpha = 1$	-7.09 ± .003	-6.86 ± .006
	$\gamma = .1,\ \alpha = 10$	-7.08 ± .003	-6.90 ± .008
	$\gamma = 1,\ \alpha = .1$	-7.08 ± .003	-6.83 ± .007
	$\gamma = 1,\ \alpha = 1$	-7.08 ± .002	-6.86 ± .006
	$\gamma = 1,\ \alpha = 10$	-7.06 ± .003	-6.88 ± .008
	$\gamma = 10,\ \alpha = .1$	-7.07 ± .004	-6.81 ± .006
	$\gamma = 10,\ \alpha = 1$	-7.07 ± .003	-6.83± .005
	$\gamma = 10,\ \alpha = 10$	-7.06 ± .003	-6.88 ± .010

Table 4.2: Per-word log likelihood of test documents

4.7.3 Wikipedia articles with a graph structure

In our final experiment we illustrate how GraphLDA can be used to update an existing category graph. We use the aforementioned 518 machine-learning Wikipedia articles, along with their category labels, to learn topic distributions for each node in Figure 4.1. The sampler is initialized with the correct placement of nodes and each document is initialized to a random path from the root to its category label. After $2,000$ iterations, we fix the path assignments for the Wikipedia articles and introduce a new set of documents. We use a

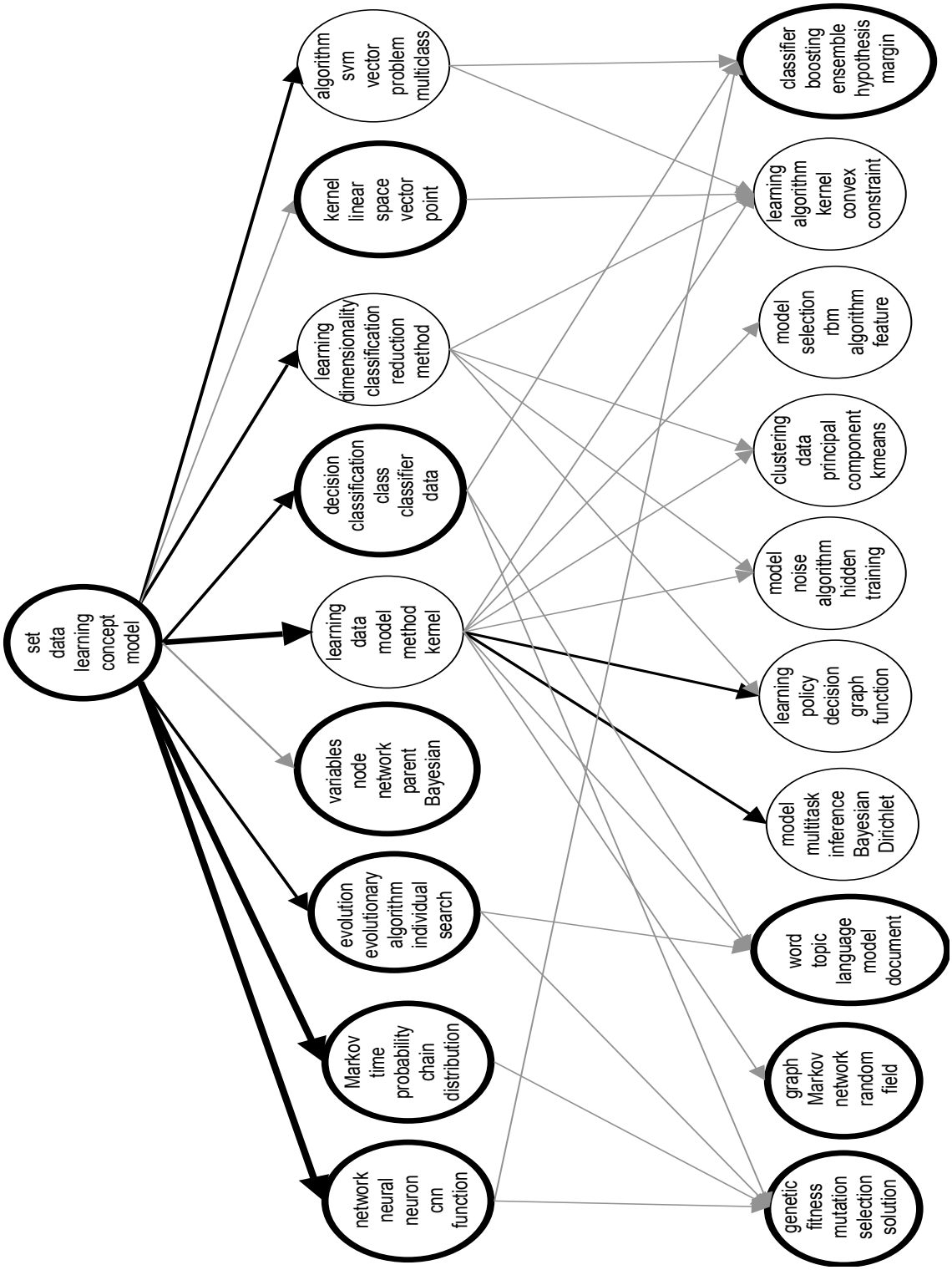

Figure 4.5: Wikipedia graph structure with additional machine learning abstracts. The edge widths correspond to the probability of the edge in the graph

collection of 400 machine learning abstracts from the International Conference on Machine Learning (ICML). We sample paths for the new collection of documents keeping the paths from the Wikipedia articles fixed. The sampler was allowed to add new nodes to each level to explain any new concepts that occurred in the ICML text set. Figure 4.5 illustrates a portion of the final graph structure. The nodes in bold are the original nodes from the Wikipedia category graph. The edge widths correspond to the probability of the edge in the graph.

GraphLDA adds a number of nodes to the existing concept graph that correspond to well-defined topics in machine learning, e.g. clustering (clustering, data, principal components, k-means), support vector machines (algorithm, svm, vector, problem, multiclass), dimensionality reduction (learning, dimensionality, classification, reduction, method), and kernel functions (learning, algorithm, kernel, convex, constraint).

In addition, GraphLDA is able to identify a number of meaningful relationships between concepts. Table 4.3 lists some of the learned relationships (i.e. edges) between nodes in the second level of the graph and leaf nodes.

Parent node	Child node
support vector machines	ensembles and boosting
support vector machines	kernels, constraints, convex optimization
linear algebra	kernels, constraints, convex optimization
dimensionality reduction	clustering, k-means, PCA
classification	ensembles and boosting
evolutionary algorithms	fitness, mutation, selection
neural networks	ensembles and boosting

Table 4.3: Examples of relationships (edges) learned by GraphLDA.

Note that because ensemble learning (classifier, boosting, ensemble, hypothesis, margin) was in the original category graph as a leaf node, the newly added concept support vector machines (algorithm, svm, vector, problem, multiclass) was placed in the second level – this

satisfies the geometric property where those words not directly related to ensemble learning (e.g. words related to specific supervised classifiers such as neural networks and SVMs) must be placed higher in the category graph. However, kernel methods (kernel, linear, space, vector, point) was also in the original category graph in the second level. Since kernel methods and support vector machines are related but edges are not allowed between nodes in the same level, a new leaf node is added on the topic of kernels and constrained optimization (learning, algorithm, kernel, convex, constraint). As a consequence, the only words retained at the node kernel methods were the more general words related to linear algebra (kernel, linear, space, vector, point).

Similarly, note the newly added concept regarding dimensionality reduction that contains more general terms such as "learning", "dimensionality", "classification", "reduction" whereas its child node contains more specific terms "k-means", "principal", and "component"

The results show that GraphLDA is capable of augmenting an existing concept graph with new concepts as well as learning meaningful relationships between concepts.

4.8 Summary of contributions

We present a flexible non-parametric prior for rooted, directed, acyclic graphs with a possibly infinite number of nodes. We constructed this prior by specifying a stick-breaking distribution at each node that governs the probability of transitioning from the given node to another node in the graph.

We combined this non-parametric prior over graph structures with latent Dirichlet allocation to create a new probabilistic model called GraphLDA for learning concept graphs from text. A concept graph is a rooted, directed graph where the nodes represent concepts – i.e. semantically-related collections of words – and the edges represent relationships between

concepts. Concept graphs provide a useful summary and visualization of the semantic content and structure of document sets.

We showed how GraphLDA can be used to learn a concept graph from a collection of documents or can be used to update an existing graph structure in the presence of new labeled documents (identifying new concepts, and new relationships between concepts, that do not already occur in the graph). We derived and presented an inference procedure for GraphLDA based on collapsed Gibbs sampling and a Metropolis Hastings algorithm that mixes over the permutation of feasible nodes in the stick-breaking distribution $\mathcal{P}_t$.

We illustrated the performance of GraphLDA on a set of simulated documents where we increase the proportion of labeled documents used for training. We then compared the performance of GraphLDA to the hierarchical Pachinko allocation model (hPAM) and hierarchical latent Dirichlet allocation (hLDA). We used both the empirical likelihood algorithm and the left-to-right algorithm [74] to compute the per-word log likelihood on a hold-out set of articles. GraphLDA is competitive when computing the empirical log likelihood but has lower performance in terms of the left-to-right algorithm which we speculate is due to our choice of a geometric distribution over levels (positioned at the last node in a document's path). Although this choice is supported by the intuition that a majority of words in a document should be explained by the node corresponding to the label of the document, a more flexible approach may result in better performance. Finally, we illustrate an application of GraphLDA to Wikipedia's category graph. We show how GraphLDA can be used to update a portion of the Wikipedia category graph rooted at the node MACHINE LEARNING given a collection of machine learning abstracts.

In summary:

- We presented a flexible non-parametric prior for rooted, directed, acyclic graphs with a possibly infinite number of nodes.

- We combineed this prior over graphs with latent Dirichlet allocation to create a new generative model called GraphLDA for learning concept graphs from text.
- We showed how GraphLDA could be used to learn a concept graph from a collection of documents or to update an existing graph structure in the presence of new labeled documents.
- We illustrateed the performance of GraphLDA on a set of simulated documents where we increase the proportion of labeled documents used for training.
- We compared the performance of GraphLDA to the hierarchical Pachinko allocation model (hPAM) and hierarchical latent Dirichlet allocation (hLDA) using both the empirical likelihood algorithm and the left-to-right algorithm [74].
- We illustrateed an application of GraphLDA to Wikipedia's category graph. We showed how GraphLDA can be used to update a portion of the Wikipedia category graph rooted at the node MACHINE LEARNING given a collection of machine learning abstracts.

4.9 Future directions

One important extension to the work presented in this chapter is scalability to large graphs which is likely to be an important issue in practice. Computing the probability of every path during sampling, where the number of paths is a product over the number of nodes at each level, is a computational bottleneck in the current inference algorithm and will not scale. Approximate inference methods that can address this issue should be quite useful in this context. Other interesting extensions include allowing the model to handle multiple paths per document (thus capturing the multi-labeled nature of text) as well as experimentation with other distributions over levels beyond the geometric distribution used in this work.

Chapter 5

An analysis of the multinomial Dirichlet mixture model for text classification

In this final chapter, we analyze the accuracy of the multinomial Dirichlet mixture model when used for text classification. Figure 5.1 (a) shows the plate notation for the statistical model we term the **multinomial Dirichlet mixture model**. In this model, there are K classes. The latent variable y indicates the class membership of the document x. The prior probability of a class k is parameterized by the probability vector θ, i.e. $p(y = k) = \theta_k$, and the class likelihood function, i.e. the conditional probability function $p(x|y = k)$, is a multinomial distribution parameterized by the probability vector ϕ_k (and the integer L which corresponds to the number of words in the document x). We place a Dirichlet prior over the class multinomial parameters ϕ_k with hyper-parameter η[1]. Figure 5.1 (b) and (c) shows the relationship of this statistical model to other more commonly used statistical models of text. In Figure 5.1 (b) the class probability vector θ is unknown and sampled from a Dirichlet

[1] η is either a vector of strictly-positive real numbers or a scalar value. η is also independent of the class

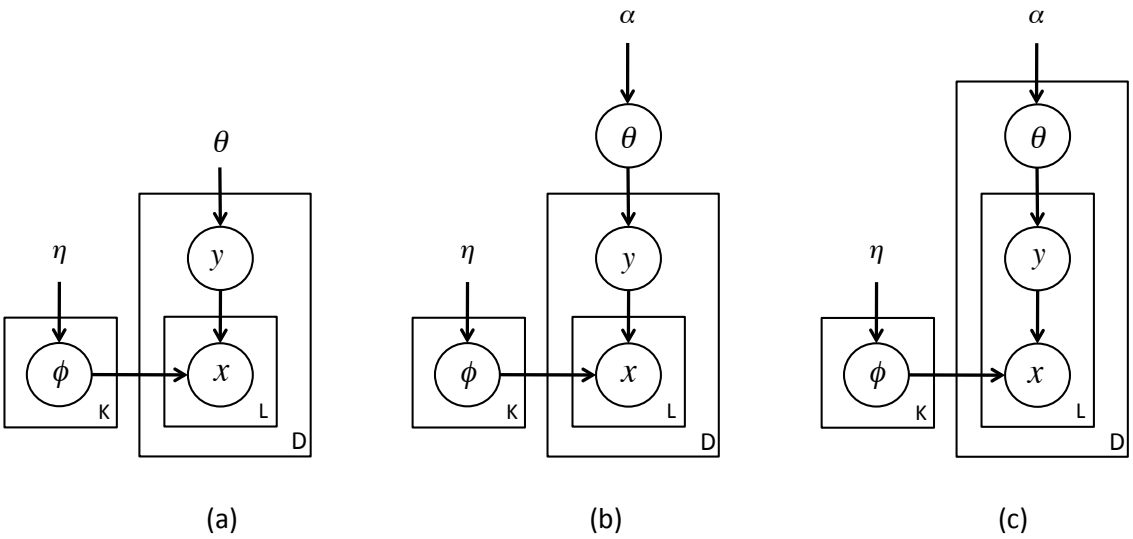

Figure 5.1: (a) plate notation for the generative model investigated in this chapter (b) plate notation for a multinomial Dirichlet mixture model with a Dirichlet prior over the class probabilities θ_d (c) plate notation for latent Dirichlet allocation.

prior with hyper-parameter α. In Figure 5.1 (c) each document has its own distribution over classes and each word in the document has its own class indicator variable. This latter model is known as latent Dirichlet allocation [5]. As can be seen from the plate notation, the statistical model we analyze in this chapter is the starting point of many of the more complex statistical models used for probabilistic text classification [40, 55, 61].

We motivate the work in this chapter using the following scenario. Let x be a document. Our task is to classify x as belonging to one of K classes. To make the analysis simpler, we will assume here and for the remainder of the chapter that $K = 2$. We first make the assumption that x arises from a multinomial Dirichlet mixture model as defined in Figure 5.1 (a). Based on this assumption and other information that is available to us – e.g. we might also observe the class multinomial parameters – we compute the posterior distribution over y and assign x to the class with the highest posterior probability. What is the probability that we have misclassified x and how do the model parameters control this probability? In particular, what is the relationship between parameters such as the document length, vocabulary size, and similarity of the class multinomial parameters and the probability of misclassifying x?

These questions are the main focus of our work. We consider two different scenarios in which different information is available to us. In the first scenario, discussed in Section 5.2, we observe the class multinomial parameters. In this case, we use the likelihood ratio test to classify the document x. In the second scenario, discussed in Section 5.3, we observe only a set of representative documents from each class. In this case, we use the *marginal* likelihood ratio to classify x. In both scenarios, we want to analyze the accuracy of the classification rule by computing the probability of a misclassification and determining the relationship between this probability and certain model parameters of interest.

Before moving on, we make clear some of the assumptions of this work. First, the statistical model we examine in this chapter (the multinomial Dirichlet mixture model as given by Figure 5.1 (a)) is a *unigram language model* where the words in a document are assumed to be conditionally independent given the class multinomial parameters. This leads to a "bag-of-words" model where documents are represented as unordered sets of words. Second, we are not concerned with obtaining an estimate of the class multinomial parameters. In the first scenario, we assume that these parameters are known. In the second scenario, we take a fully Bayesian approach and compute the marginal likelihood ratio by integrating over the unknown parameters. It would be interesting in future work to consider a scenario in which we first compute an estimate of the class multinomial parameters and then use these estimates to predict using the likelihood ratio test. In this case, we would be interested in determining the amount of additional error introduced by our estimation process. Third, in all the work presented in this chapter we assume that the class probabilities given by θ are fixed known. Finally, we use text as a motivating example. We refer to "words", "documents", "document lengths", and "vocabulary sizes". However, the results presented in this chapter are more general and apply in any situation where data is modeled as multinomial.

5.1 Related work

The multinomial Dirichlet mixture model studied in this chapter is equivalent to a naive Bayes classifier with a Dirichlet prior over the class multinomial parameters. A Bayesian treatment of the class multinomials provides a framework in which to characterize the relationship between the error rate of the classifier and the similarity of the class multinomials as measured by the Jeffrey's divergence or by the Dirichlet hyperparameter itself. Furthermore, in Section 5.3 where we assume that we do not observe the class multinomial parameters (but observe only a set of representative documents) this Bayesian treatment allows us to marginalize over the unknown parameters and derive a classifier based on the marginal likelihood ratio.

The error rate of the naive Bayes classifier

There is a large body of work that investigates the relationship between the error rate of the naive Bayes classifer and the amount of dependence in the features. [14, 58, 33, 32, 59, 77]. Rish et al. [58] present an empirical analysis using Monte Carlo simulations. They show that the naive Bayes classifier is optimal for complete independence or complete dependence of the features. Kuncheva et al. [33, 32] analyze the error rate of the naive Bayes classifier when there are only two classes and two binary features.

The main difference between our work and this body of research is that we are interested in determining the relationship between the error rate of the multinomial Dirichlet mixture model and certain model parameters of interest (e.g. the document length, the vocabulary size, the hyper-parameter η). We are interested in answering questions such as, "How fast does the error rate increase when the similarity between the class multinomial parameters increases?" We are not (for example) investigating the relationship between the error rate and the amount of dependence between the words in a document (although this would be

interesting to investigate as well).

Likelihood functions for statistical text classification

Two common choices for the likelihood function $p(x|y = k)$ for text classification are the multinomial distribution and the multivariate Bernoulli distribution (see chapter 13 in [40] for a presentation of both models). McCallum et al. [41] present a comparison of the multinomial and multivariate Bernoulli distributions for text classification. They find that the multinomial distribution often outperforms the multivariate Bernoulli. Eyheramendy et al. [15] compare the multivariate Bernoulli, multinomial, Poisson, and negative binomial models for text classification and found that the multinomial model generally performed best.

In this work, we assume a multinomial likelihood although again it would be interesting to reproduce this work with a different likelihood assumption.

The Bayes error rate

The work presented in Section 5.2 deals with the Bayes error rate (which we derive for the likelihood ratio test). The *average error* of a classifier[2] is the probability of misclassifying a random instance x. The *Bayes error* is the lowest achievable average error given by the integral,

$$\begin{aligned} p_\epsilon &= E[p_\epsilon(x)] \\ &= \int p_\epsilon(x)p(x)\ d(x) \end{aligned}$$

[2]also known as the *total error* or the *true error* in the statistical pattern recognition literature

where $p_\epsilon(x)$ is the minimum achievable error rate at x defined as $1 - \max_k p(y = k|x)$. The Bayes error is the average error of a classifier that has perfect knowledge of the class probabilities $p(y = k)$ and the likelihood functions $p(x|y = k)$. See [19] and [76] for a thorough treatment.

The computation of the Bayes error rate is often intractable. The Bayes error rate can be directly computed in simple cases such as Gaussian class likelihoods with equal covariances. However, in general, the Bayes error rate must be approximated. Methods of approximation include non-parametric kernel density estimates [20] and ensembles of classifiers [71]. Fukunaga [19] provides a summary of upper bounds for the Bayes error rate including the Chernoff bound (pg. 97), the Bhattacharyya bound (pg. 99), and the asymptotic nearest neighbor error (pg. 102).

In our case, we are fortunate enough to have a central limit theorem for multinomial sums that we use to provide a Normal approximation to the Bayes error rate of the likelihood ratio test (in Section 5.2) and the average error rate of the marginal likelihood ratio classifier (in Section 5.3).

The closest work to our own is an analysis of the error rate of a naive Bayes classifier with a multinomial likelihood presented by Van Dyke et al [72]. Van Dyke et al. also appeal to the same central limit theorem to approximate the Bayes error rate of the naive Bayes classifier which is used in turn as a feature selection algorithm by noting that the mean and variance of the Normal approximation are additive over the features. They propose a method for computing the error rate over subsets of features.

Our work differs from Van Dyke et al. in a number of ways. First, as mentioned above, the statistical model we analyze in the first section of this chapter is equivalent to the naive Bayes classifier analyzed by Van Dyke et al. with the addition of a Dirichlet prior over the class multinomial parameters. This Bayesian treatment provides a framework in which

to characterize the similarity between the class multinomials. Second, the motivation of this work is markedly different from the motivation of Van Dyke et al. whose main goal was feature selection. Our motivation is to provide insight into a statistical model used for classification by characterizing how the error rate of the classifier varies as a function of the model parameters. As a consequence, we present a series of Monte Carlo simulations that cover a wide range of values for the model parameters with an emphasis on values that are commonly encountered in text classification – e.g. we look at document lengths ranging from 15 words (the length of a tweet) to 1200 words (the length of a news article). We also compare the error rate from classifying real text documents with the error rate obtained from classifying simulated data generated by a multinomial model. In contrast, Van Dyke et al. restrict their attention to a small simulated example with multinomial parameters that were artificially constructed and show how the Bayes error changes as more features are used[3]. Finally, we continue on in Section 5.3 to derive and approximate the average error rate in the case where the multinomial parameters are not known.

5.2 Scenario 1: known multinomial parameters

In this first scenario, we assume that the class multinomial parameters ϕ_k are known where $K = 2$. Let x be a document and y its (latent) class assignment. We classify x by computing the posterior probability distribution over y conditioned on x and the class parameters ϕ_1 and ϕ_2 and assigning x to the class with the highest posterior probability. It is trivial to show that this classification rule is just the likelihood ratio test in which the likelihood ratio – the ratio of the likelihood of x given $y = 1$ to the likelihood of x given $y = 2$ – is compared to the prior odds.

We analyze the accuracy of this classification rule by computing the probability that the

[3]They present only one simulated example that would be analogous to a vocabulary size of 500 words, $10,000$ Monte Carlo simulations, and a document length of 10 words

document x is misclassified. This is called the probability of error. If we average the probability of error over the set of documents x then we obtain the *average error*. It is well-known that the likelihood ratio test is optimal in that it achieves the lowest possible average error which is called the *Bayes error rate* (see [19] pg. 53 or [76] pg. 7).

Our goal in this section is to understand how the Bayes error rate changes as a function of the model parameters, e.g. the length of the document x or the size of the vocabulary. In addition, we apply our work to real text data and analyze to what extent text data deviates from the multinomial assumption.

5.2.1 Notation and the generative model

Let W be a positive integer that represents a number of possible outcomes. In our case, W is the number of words in our vocabulary. The class multinomial parameters are given by $\phi_k = (\phi_{k1}, \ldots, \phi_{kW})$ where $\sum_i \phi_{ki} = 1$ and $0 < \phi_{ki} < 1$ for $k \in \{1, 2\}$. To refer to both ϕ_1 and ϕ_2 we will use the notation $\phi_{1:2}$.

We represent documents using the vector notation $x = (x_1, \ldots, x_W)$ where x_i is the number of times the ith word in the vocabulary occurs in the document x. We denote the length of the document by $L = \sum_i x_i$. Finally, let $y \in \{1, 2\}$ denote the true (but unknown) class assignment of the document x.

Then the generative model for x is as follows:

We call this the multinomial Dirichlet mixture model. The hyper-parameter η is a strictly positive real-valued number. The Dirichlet distribution is traditionally parameterized by a vector of such positive reals. However, we use a *symmetric* Dirichlet distribution where Dirichlet(η) is shorthand for Dirichlet($[\eta, \ldots, \eta]$)[4]. The scalar parameter η is sometimes

[4]The vector $[\eta, \ldots, \eta]$ has dimension W

$$\eta \in \mathbb{R}_{>0},\ \theta \in (0,1)$$
$$\phi_i \sim \text{Dirichlet}(\eta) \text{ for } i \in \{1,2\}$$
$$y \sim \text{Discrete}(\theta, 1-\theta)$$
$$x \sim \text{Multinomial}(\phi_y, L)$$

Table 5.1: Generative model

called the *concentration parameter*[5]: when $\eta < 1$, the probability mass of the Dirichlet distribution is concentrated on sparse probability vectors that give low probability to most outcomes and high probability to only a few outcomes. When $\eta = 1$, the probability mass is spread uniformly over all probability vectors. When $\eta > 1$, the probability mass is concentrated on those probability vectors with nearly uniform (i.e. equal) probability over all outcomes.

5.2.2 Classification rule

We begin by deriving a classification rule for the document x. To classify x, we compute the posterior distribution of y conditioned on the observed data x and the parameters $\phi_{1:2}$. This posterior distribution is given by,

$$\begin{aligned} q_k(x) &= p(y = k \mid x, \phi_{1:2}, \eta, \theta) \\ &= \frac{p(x|\phi_{1:2}, y = k)\ p(y = k|\theta)}{p(x|\phi_{1:2}, \theta)} \end{aligned} \tag{5.1}$$

Note that the denominator does not play a role in classifying x since it is constant with respect to y. Given the posterior distribution over y, our classification rule is as follows:

[5]Others define the sum $W\eta$ as the concentration parameter

Classification rule: If $q_1(x) > q_2(x)$ then we classify x as belonging to ϕ_1. If $q_1(x) < q_2(x)$ then we classify x as belonging to ϕ_2. If $q_1(x) = q_2(x)$ we make a random decision.

Since it will often be more convenient to work in log space, we rewrite the inequality $q_1(x) > q_2(x)$ in terms of the log function, and restate the classification rule.

$$\begin{aligned} q_1(x) &> q_2(x) && \text{iff} \\ p(x|\phi_{1:2}, y=1)p(y=1|\theta) &> p(x|\phi_{1:2}, y=2)p(y=2|\theta) && \text{iff} \\ \frac{p(x|\phi_{1:2}, y=1)}{p(x|\phi_{1:2}, y=2)} &> \frac{p(y=2|\theta)}{p(y=1|\theta)} && \text{iff} \\ \log p(x|\phi_{1:2}, y=1) - \log p(x|\phi_{1:2}, y=2) &> \log p(y=2|\theta) - \log p(y=1|\theta) \end{aligned} \tag{5.2}$$

The left-hand side of the final inequality is the log of the likelihood ratio. We denote this quantity as $\ell(x)$. Recalling that $y \sim \text{Discrete}(\theta, 1-\theta)$, we rewrite the right-hand side of the final inequality as $\log \frac{1-\theta}{\theta}$. We now restate the classification rule:

Classification rule: If $\ell(x) > \log \frac{1-\theta}{\theta}$ then we classify x as belonging to ϕ_1. If $\ell(x) < \log \frac{1-\theta}{\theta}$ then we classify x as belonging to ϕ_2. If $\ell(x) = \log \frac{1-\theta}{\theta}$, we make a random decision.

This classification rule is known as the likelihood ratio test, or to be more precise, the log likelihood ratio test.

5.2.3 The Bayes error

Having derived a classification rule, we now analyze it by computing the probability of incorrectly classifying a document x. We follow the derivation presented in Fukunaga [19].

Suppose that for a particular document x, $q_1(x) = 0.4$ and $q_2(x) = 0.6$. We classify x as belonging to class 2, but there is a 40% chance that x actually belonged to class 1. For this document, the probability of an error is 0.4. In general, the probability of error for a document x is the minimum of $q_1(x)$ and $q_2(x)$. If we average the probability of error over all documents x, then we obtain the average error. The likelihood ratio test is optimal in that it achieves the lowest possible average error which is called the Bayes error rate. We denote the Bayes error rate as p_ϵ. Note that the Bayes error rate is a function of ϕ_1, ϕ_2, and the document length L.

We can compute the Bayes error rate by averaging over the set of all documents with length L:

$$\begin{aligned}
p_\epsilon &= E_{x|\phi_{1:2}}\big[\, \min\{q_1(x), q_2(x)\} \,\big] \\
&= \int p(x|\phi_{1:2}) \cdot \min\big\{q_1(x), q_2(x)\big\}\, dx \\
&= \int p(x|\phi_{1:2}) \cdot \min\left\{ \frac{p(x|\phi_{1:2}, y=1)\theta}{p(x|\phi_{1:2})}, \frac{p(x|\phi_{1:2}, y=2)(1-\theta)}{p(x|\phi_{1:2})} \right\} dx \\
&= \theta \int_{\Omega_2} p(x|\phi_{1:2}, y=1)\, dx \quad + \quad (1-\theta) \int_{\Omega_1} p(x|\phi_{1:2}, y=2)\, dx
\end{aligned}$$

where Ω_1 is the set of documents for which $p(x|\phi_{1:2}, y=2) < p(x|\phi_{1:2}, y=1)$ (i.e. the set of documents that are classified as belonging to class 1) and Ω_2 is the set of documents for which $p(x|\phi_{1:2}, y=2) > p(x|\phi_{1:2}, y=1)$ (i.e. the set of documents that are classified as belonging to class 2).

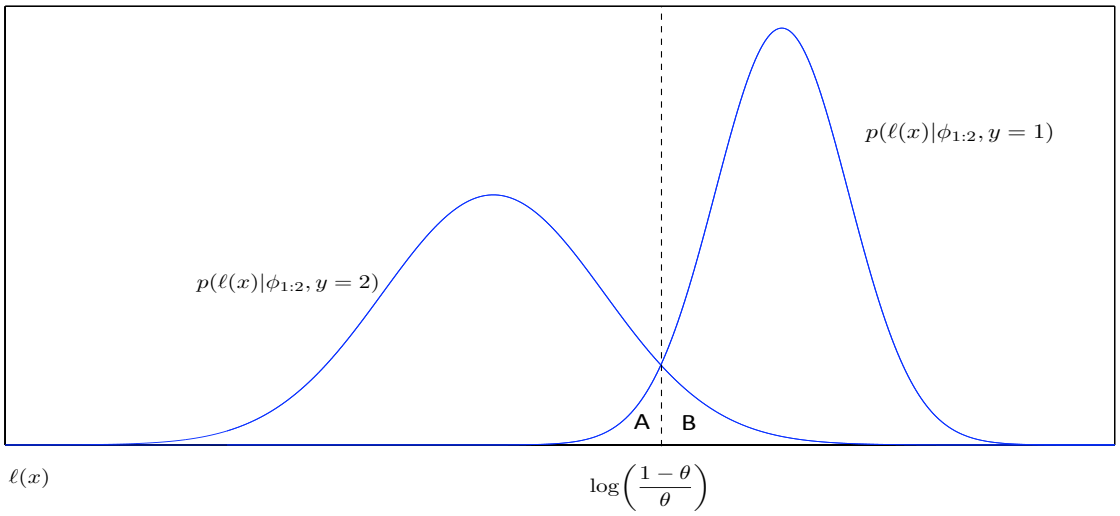

Figure 5.2: Hypothetical densities for the conditional distribution of $\ell(x)$ conditioned on y. The dotted line is the decision boundary.

As pointed out in Fukunaga [19], p_ϵ is often hard to evaluate since it requires integrating over the sets Ω_1 and Ω_2. However, we can move from integrating over Ω_1 and Ω_2 to integrating over the one-dimensional density of the log of the likelihood ratio $\ell(x)$.

Figure 5.2 shows hypothetical densities for $\ell(x)$ conditioned on y and the parameters $\phi_{1:2}$. The decision boundary is at $\log(\frac{1-\theta}{\theta})$. When $\ell(x)$ is less than the decision boundary, x is classified as belonging to class 2. When $\ell(x)$ is greater than the decision boundary, x is classified as belonging to class 1. The Bayes error rate is given by the sum of the areas A and B. The area A is the error from classifying x as belonging to class 2 when in reality it belongs to class 1 and the area B is the error from classifying x as belonging to class 1 when in reality it belongs to class 2. Using this intuition, we can express the Bayes error rate as an integral over $\ell(x)$ as follows,

$$p_\epsilon = \theta \int_{-\infty}^{\log \frac{1-\theta}{\theta}} p(\ell(x)|\phi_{1:2}, y = 1)\, d\ell(x) + (1-\theta) \int_{\log \frac{1-\theta}{\theta}}^{\infty} p(\ell(x)|\phi_{1:2}, y = 2)\, d\ell(x) \quad (5.3)$$

Equation 5.3 states that in the region from negative infinity to the decision boundary, we integrate over the density of $\ell(x)$ conditioned on $y = 1$, and from the decision boundary to positive infinity, we integrate over the density of $\ell(x)$ conditioned on $y = 2$. These integrals are weighted by the prior probability of $y = 1$ and $y = 2$ respectively.

We now have a more tractable expression for p_ϵ in terms of two integrals over the real line. However, Equation 5.3 still requires knowing the conditional density of $\ell(x)$. It may appear that we have traded one problem for another, however we continue on in the remainder of this section to explore methods for approximating this density.

5.2.4 The log of the likelihood ratio

We quickly pause to derive the explicit form of the log of the likelihood ratio $\ell(x)$ for our generative model where x given y and $\phi_{1:2}$ is sampled from a multinomial distribution:

$$\begin{aligned}
\ell(x) &= \log p(x|\phi_{1:2}, y = 1) - \log p(x|\phi_{1:2}, y = 2) \\
&= \sum_{i=1}^{W} x_i \ \log(\phi_{1i}) - \sum_{i=1}^{W} x_i \ \log(\phi_{2i}) \\
&= \sum_{i=1}^{W} x_i \ \Big(\log(\phi_{1i}) - \log(\phi_{2i}) \Big)
\end{aligned} \tag{5.4}$$

If $x_i = 0$, then the corresponding term in the summation is defined to be 0.

5.2.5 Monte Carlo estimates

One way to approximate the conditional density of the log likelihood ratio $\ell(x)$ is by computing a Monte Carlo estimate. We first sample two multinomial distributions ϕ_1 and ϕ_2 from a

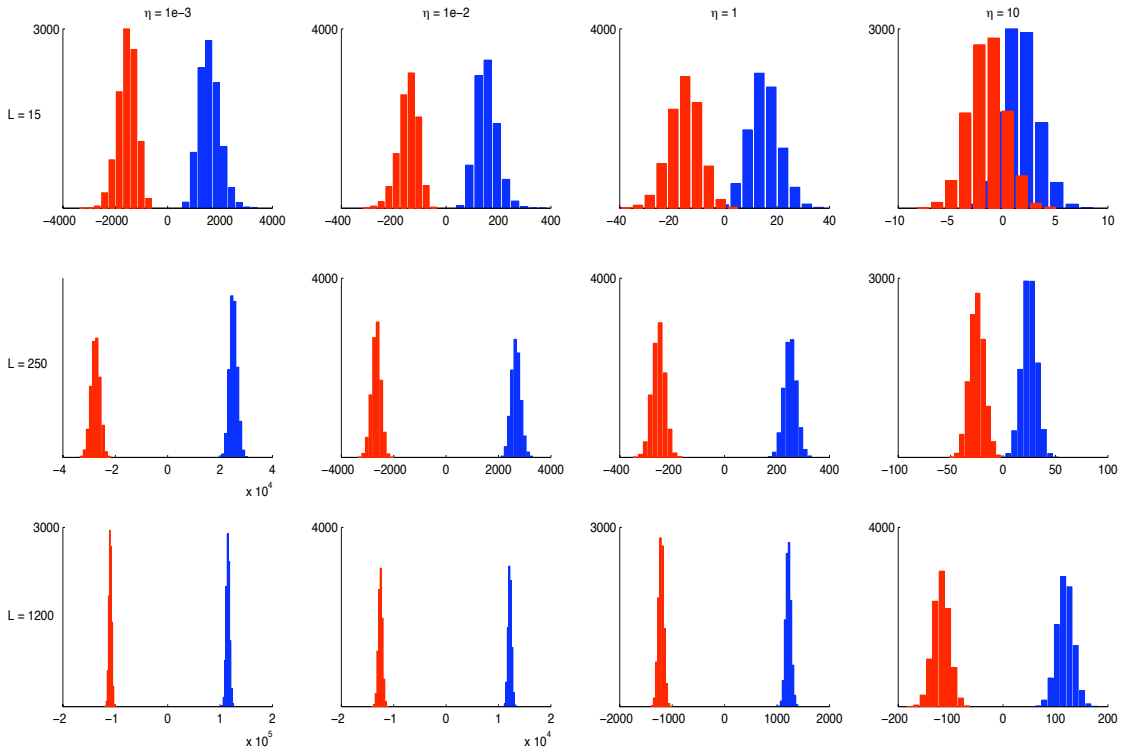

Figure 5.3: Monte Carlo estimate of $\ell(x)$

symmetric Dirichlet prior with hyper-parameter η. Next, we sample $S = 10,000$ documents of length L from ϕ_1 and compute $\ell(x)$ for each sample. We do the same for ϕ_2.

Figure 5.3 shows the results of this simulation. The three rows correspond to document lengths of $L = 15$, $L = 250$ and $L = 1200$ – the average length of a tweet, a scientific abstract, and a news article respectively[6]. We selected these document lengths since we are interested in analyzing situations that correspond to real-world situations. The four columns correspond to a symmetric Dirichlet prior with hyper-parameter $\eta = 0.01$, $\eta = .1$, $\eta = 1$, and $\eta = 10$. We set the prior probability of ϕ_1 to be $\theta = 0.5$, and we use a vocabulary size of $W = 10,000$ words[7]. The decision boundary is at $\log \frac{1-\theta}{\theta} = 0$.

The blue histogram in each plot shows the distribution of $\ell(x)$ for the documents sampled

[6]These are approximations to the length of a tweet, abstract, and news articles. News articles in particular vary based upon the newspaper and the content of the article.

[7]We will see later that the vocabulary size does not have a strong effect on the Bayes error rate.

from ϕ_1. The red histogram in each plot shows the distribution of $\ell(x)$ for the documents sampled from ϕ_2. The proportion of the blue histogram that is less than the decision boundary (at 0) weighted by the prior probability of class 1 (0.5) plus the proportion of the red histogram that is greater than the decision boundary weighted by the prior probability of class 2 (also 0.5) is an estimate of the Bayes error rate.

The most error occurs in the upper-right plot where the document length is 15 words and the Dirichlet hyper-parameter is 10. Recall that as the hyper-parameter approaches infinity, the Dirichlet probability mass becomes concentrated on those multinomials that are close to uniform and are thus hard to distinguish from one another. As the number of words in the document increases, or the hyper-parameter decreases, the histograms move away from the decision boundary. Interestingly, with as few as 250 words (the second row), the error seems to be negligible.

5.2.6 Central limit theorem and moments

Recall the form of the log of the likelihood ratio $\ell(x)$:

$$\ell(x) = \sum_{i=1}^{W} x_i \left(\log(\phi_{1i}) - \log(\phi_{2i}) \right) \tag{5.5}$$

Although $\ell(x)$ is a summation of dependent random variables – dependent since the x_i are constrained to sum to L the length of the document – the histograms appear to be approximately Normal. In fact, Morris [46] presents a central limit theorem which states that, under certain conditions, as the number of non-zero terms in the summation goes to infinity, the conditional distribution of $\ell(x)$ is asymptotically Normal (Lemma 2.2 from [46]). We state here the conditions of the lemma:

Conditions of the fundamental lemma for asymptotic normality of multinomial sums:

Let x be multinomially distributed with parameters ϕ, $\sum_{i=1}^{W} \phi_i = 1$, and integer L. We assume that $\phi_i > 0$ for all i. Let $\{z_i : 1 \leq i \leq W\}$ be independent Poisson random variables where z_i has mean $\lambda_i = L\phi_i$ and $\sum_i z_i = L$. Let $\{g_i : 1 \leq i \leq W\}$ be a set of functions with domain the nonnegative integers. Define:

$$\sigma_i^2 = \mathrm{Var}\big(g_i(z_i)\big), \qquad s^2 = \sum_{i=1}^{W} \sigma_i^2$$

If the following conditions hold:

1. $E\big[g_i(z_i)\big] = 0$ for all i
2. $\mathrm{Cov}\big(\sum_i g_i(z_i), \sum_i z_i\big) = \sum_i \mathrm{Cov}(g_i(z_i), z_i) = 0$
3. As $W \to \infty$, $L \to \infty$
4. As $W \to \infty$, $\max_{1 \leq i \leq W} \phi_i = o(1)$
5. As $W \to \infty$, $\frac{1}{s^2} \max_{1 \leq i \leq W} \sigma_i^2 = o(1)$
6. As $W \to \infty$, $\frac{1}{s} \sum_i g_i(z_i) \to \mathrm{Normal}(0, 1)$

then

$$\tfrac{1}{s} \textstyle\sum_{i=1}^{W} g_i(x_i) \to \mathrm{Normal}(0, 1) \text{ as } W \to \infty$$

We show how to apply this central limit theorem to $\ell(x)$. We can rewrite the log of the likelihood ratio as $\ell(x) = \sum_{i=1}^{W} f_i(x_i)$ where $f_i(x_i) = x_i(\log(\phi_{1i}) - \log(\phi_{2i}))$. The functions $\{f_i\}$ have domain the nonnegative integers. However conditions (1), (2), and (6) do not hold true. Instead, we define a new set of functions given by Morris,

$$g_i(x) = f_i(x) - E\big[f_i(z_i)\big] - \gamma(x - \lambda_i) \quad \text{where} \quad \gamma = \frac{1}{L}\sum_{i=1}^{W} \text{Cov}(f_i(z_i), z_i)$$

The function $g_i(\cdot)$ has been engineered to have certain desirable properties. First, when we apply $g_i(\cdot)$ to the Poisson random variables, the expected value is zero – i.e. $E\big[g_i(z_i)\big] = 0$. It is less obvious, but still true, that the covariance $\sum_i \text{Cov}(g_i(z_i), z_i) = 0$. Thus, conditions (1) and (2) are satisfied. Furthermore, by the construction of $g_i(\cdot)$, the variance $\text{Var}\big(\frac{1}{s}\sum_i g_i(z_i)\big) = 1$, and we also have condition (6).

Condition (3) states that as the vocabulary grows, the length of the document grows. Since the vocabulary is often created by aggregating the unique words found in the corpus of text documents, this is not an unreasonable assumption. Conditions (4) and (5) govern how the probability mass is spread over the words as the vocabulary size grows. They ensure that the probability mass does not stay concentrated on a small subset of words despite the ever increasing size of the vocabulary. This is also a reasonable assumption for our application.

Given Morris' lemma, we can now approximate the conditional density of $\ell(x)$:

$$\begin{aligned}
&p(\ell(x)|y = k, \phi_{1:2}) \longrightarrow \text{Normal}(\mu_k, s_k^2) \quad \text{as } W \longrightarrow \infty \quad \text{where} \\
&\qquad \mu_k = E[f_i(z_i)] = L\sum_{i=1}^{W} \phi_{ki}\Big(\log(\phi_{1i}) - \log(\phi_{2i})\Big) \quad \text{and} \\
&\qquad s_k^2 = \sum_{i=1}^{W} \text{Var}(g_i(z_i)) \\
&\qquad\quad = \sum_{i=1}^{W} \text{Var}(f_i(z_i)) - L\gamma^2 \\
&\qquad\quad = L\left[\sum_{i=1}^{W} \phi_{ki}\ \log^2\left(\frac{\phi_{1i}}{\phi_{2i}}\right) - \left(\sum_{i=1}^{W} \phi_{ki}\ \log\left(\frac{\phi_{1i}}{\phi_{2i}}\right)\right)^2\right]
\end{aligned} \tag{5.6}$$

Note that the mean μ_k and variance s_k^2 are functions of ϕ_k where k is determined by the value of y.

If we examine the mean and variance of the Normal approximation, we make the satisfying observation that the mean μ_k and the variance s_k^2 are both functions of f-divergences. An f-divergence is a function, $D_f(p||q)$, that takes as input two probability distributions p and q and returns a non-negative value indicating the dissimilarity of the distributions. In other words, an f-divergence is a measure of how difficult it is to discriminate between two distributions. It is satisfying then to find that the Bayes error rate – also a measure of the difficulty of distinguishing between ϕ_1 and ϕ_2 – can be approximated using divergences. In fact, we could have anticipated this outcome from the form of $\ell(x)$. See Pardo [52] for a fuller treatment of divergences and their relation to statistical inference.

The mean μ_k can be recognized as the length times the Kullback-Leibler divergence:

$$\mu_k = \begin{cases} L\ D_{KL}(\phi_1||\phi_2) & : k = 1 \\ -L\ D_{KL}(\phi_2||\phi_1) & : k = 2 \end{cases}$$

Given two discrete distributions p and q, the Kullback-Leibler divergence is defined as:

$$\begin{aligned} D_{KL}(p||q) &= \sum_i p_i \log\left(\frac{p_i}{q_i}\right) \\ &= E_p\left[\log\left(\frac{p}{q}\right)\right] \end{aligned}$$

We also recognize that the difference between the means $(\mu_1 - \mu_2) = L\big(D_{KL}(\phi_1||\phi_2) + D_{KL}(\phi_2||\phi_1)\big)$ is the document length L times the Jeffrey's divergence. Thus the Jeffrey's divergence provides a back-of-the-envelope approximation to the Bayes error rate: as the

document length L increases by one, the difference between the means increases by the Jeffrey's divergence.

The variance s_k^2 can also be recognized as a function of the Kullback-Leibler divergence and the exponential divergence:

$$s_k^2 = \begin{cases} L\ (D_e(\phi_1||\phi_2) - D_{KL}(\phi_1||\phi_2)) & : k = 1 \\ L\ (D_e(\phi_2||\phi_1) - D_{KL}(\phi_2||\phi_1)) & : k = 2 \end{cases}$$

Here D_e is the exponential divergence given by,

$$D_e(p||q) = \sum_i p_i \log^2\left(\frac{p_i}{q_i}\right)$$

Recall that our goal is to find an approximation for the conditional density of $\ell(x)$ so that we can approximate the Bayes error rate. We now plug the Normal approximation provided by the central limit theorem into Equation 5.3:

$$\begin{aligned} p_\epsilon &= \theta \int_{-\infty}^{\log\frac{1-\theta}{\theta}} p(\ell(x)|\phi_{1:2}, y = 1)\ d\ell(x) + (1-\theta) \int_{\log\frac{1-\theta}{\theta}}^{\infty} p(\ell(x)|\phi_{1:2}, y = 2)\ d\ell(x) \\ &\approx \theta\ \Phi\left(\frac{\log\frac{1-\theta}{\theta} - \mu_1}{\mathrm{s}_1}\right) + (1-\theta)\ \Phi\left(\frac{\mu_2 - \log\frac{1-\theta}{\theta}}{\mathrm{s}_2}\right) \end{aligned} \tag{5.7}$$

where $\Phi(\cdot)$ is the cumulative distribution function (cdf) of the standard Normal distribution and s_k is the standard deviation. Equation 5.7 gives us a direct way to compute p_ϵ as a function of ϕ_1, ϕ_2, and the document length L.

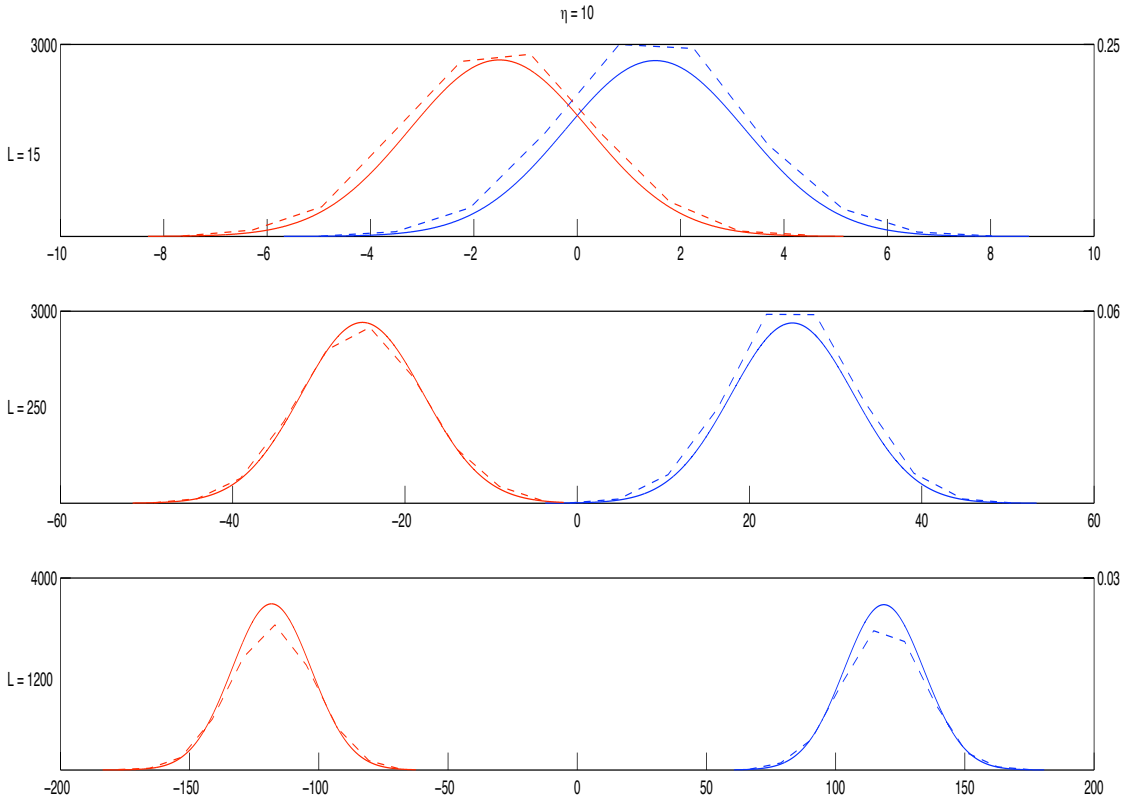

Figure 5.4: Monte Carlo estimate of $\ell(x)$ along with the Normal approximation

Figure 5.4 shows the Monte Carlo estimate of $\ell(x)$ for document lengths $L = 15$, $L = 250$, and $L = 1200$ and symmetric Dirichlet hyper-parameter $\eta = 10$ plotted as a dashed line. This corresponds to the last column of Figure 5.3. The Normal approximation is shown with a solid line. Using Equation 5.7, we approximate the Bayes error rate to be $p_\epsilon = 0.19$, $p_\epsilon = 0.0002$, and $p_\epsilon = 0.00$ respectively.

From Figure 5.4, we see that the Normal approximations are not perfect. To investigate the degree to which the Normal distributions deviate from the conditional density of $\ell(x)$, we consider the hardest case:[8] when the document length is small, $L = 15$, and the symmetric Dirichlet hyper-parameter is large, $\eta = 10$. Given these parameters, we sample 100 pairs of multinomial distributions. From each pair of multinomials, we sample $S = 1,000,000$

[8]Recall that as the document length increases, or the hyper-parameter decreases, the Bayes error rate approaches zero. In this case, the fit of the Normal approximation becomes less and less important

documents. We use such a large number of samples S since, ideally, we want to compare the approximation to the true Bayes error rate. Lacking the true Bayes error rate, we make do with a Monte Carlo estimate with a large number of samples. We classify the S documents using our classification rule and compute the empirical error rate: the number of incorrectly classified documents divided by the total number of documents S. The absolute difference between the empirical error rate and the approximation given by Equation 5.7, averaged across the 100 pairs, was $3.27\text{x}10^{-4}$ with a standard deviation of $2.77\text{x}10^{-4}$. Thus, we see that the central limit theorem gives a reasonably good approximation for the Bayes error rate.

5.2.7 Analysis of the Bayes error rate

In this section, we empirically investigate the relationship between the Bayes error rate – given by the approximation in Equation 5.7 – and certain quantities of interest[9]. We want to answer questions such as, "How would the Bayes error rate change if I doubled the number of words in the document?", or "How fast does the Bayes error rate increase when η increases from $\eta = 0.1$ to $\eta = 1$?".

In particular, we look at the relationship between the Bayes error rate and the Jeffrey's divergence $D_J(\phi_1||\phi_2)$, the Dirichlet hyper-parameter η, the document length L, and the vocabulary size W. We include the Jeffrey's divergence in our list since, through the Jeffrey's divergence, we can investigate the error rate for non-symmetric Dirichlet hyper-parameters. Recall that the Jeffrey's divergence is the sum of the Kullback-Leibler divergences $D_{KL}(\phi_1||\phi_2)$ and $D_{KL}(\phi_2||\phi_1)$. We abbreviate the notation for the Jeffrey's divergence as D_J.

The pseudocode used to generate the plots in this section are shown in Appendix F. We use $S = 2,500$, $\tau = 1\text{e} - 8$, and $\theta = 0.5$ for all Monte Carlo simulations (again, refer to the

[9]In the remainder of this section, we use "Bayes error rate" to mean the approximation given in Equation 5.7

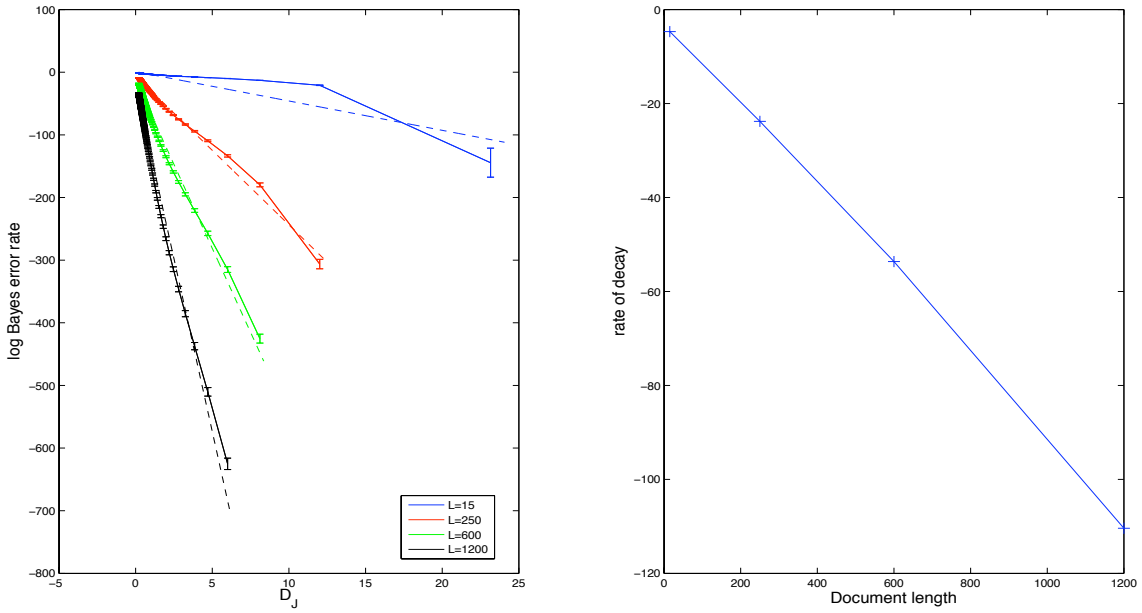

Figure 5.5: The first plot shows the log Bayes error rate p_ϵ as a function of the Jeffrey's divergence $D_J(\phi_1||\phi_2)$. Document lengths range from $L = 15, 250, 600, 1200$. The dotted lines show the best-fit linear approximations. This plot suggests that the Bayes error rate exponentially decreases as the Jeffrey's divergence increases. The second plot shows the rate of decay of the exponential function versus the document length.

pseudocode). Using $\theta = 0.5$ means that in our simulations both class 1 and class 2 are a priori equally likely.

The Jeffrey's Divergence

We begin with the Jeffrey's Divergence between ϕ_1 and ϕ_2: how does the Bayes error rate p_ϵ change as the Jeffrey's divergence D_J grows?

The first plot in Figure 5.5 shows the log Bayes error rate as a function of the Jeffrey's divergence. The pseudocode used to generate the plot is shown in Algorithm 2 in Appendix F. We set $W = 25,000$ and consider document lengths $L = 15, 250, 600, 1200$ which correspond to the blue, red, green and black lines respectively.

We see an approximately linear relationship between $\log(p_\epsilon)$ and D_J which suggests that

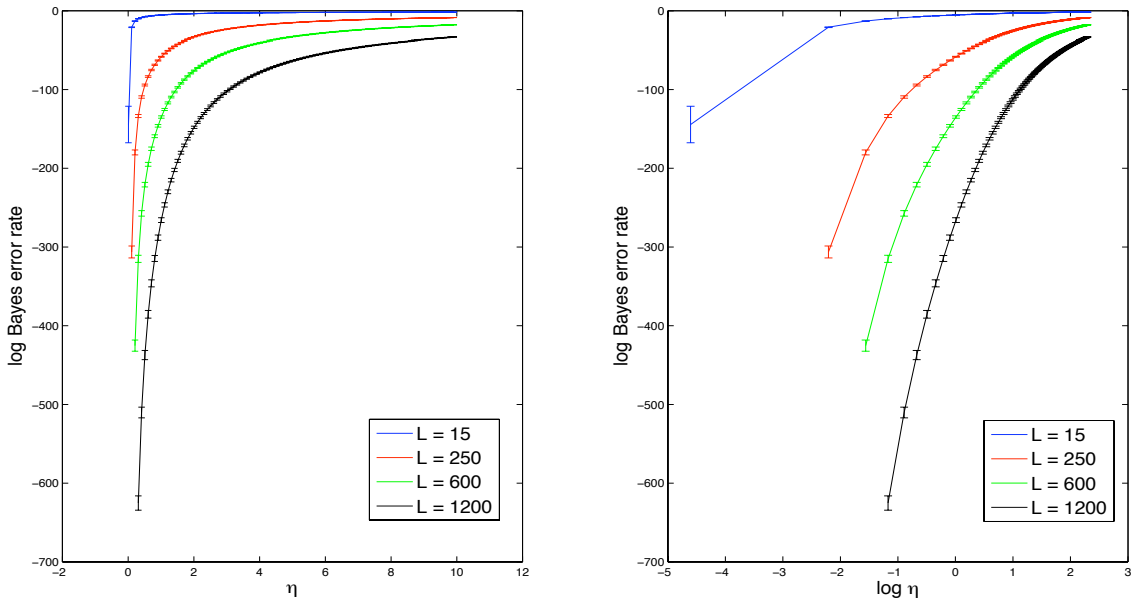

Figure 5.6: Bayes error rate as a function of the Dirichlet hyper-parameter η for document lengths $L \in \{15, 250, 600, 1200\}$ and vocabulary size $W = 25,000$. The first plot is a semi-log plot where $\log(p_\epsilon)$ is plotted against η. The second plot is a log-log plot.

the relationship between the Bayes error rate and the Jeffrey's divergence is approximately exponential. That is, the Bayes error rate exponentially decays as the Jeffrey's divergence increases. We have plotted with a dashed line the least squares regression line. The slope of the regression line gives an estimate of the rate of decay: -4.7 (for $L = 15$), -23.8 (for $L = 250$), -53.6 (for $L = 600$) and -110.4 (for $L = 1200$). The second plot in Figure 5.5 shows these rates of decay plotted against the document length L. We see that the rate of decay is a linear function of the document length L.

Dirichlet hyper-parameter η

Figure 5.6 shows the results of the Monte Carlo simulations for η. The first plot is a semi-log plot. The second plot is a log-log plot. It is not immediately clear what the relationship is between the Bayes error rate and η. From the plots, we know it is neither exponential nor does it follow a power-law. What we do know however is the functional form relating the

Jeffrey's divergence D_J and the Bayes error rate p_ϵ. If we can uncover the functional form relating η and the Jeffrey's divergence D_J, then we can compose the two functions together. Figure 5.7 illustrates this transitive relationship between η, D_J, and p_ϵ.

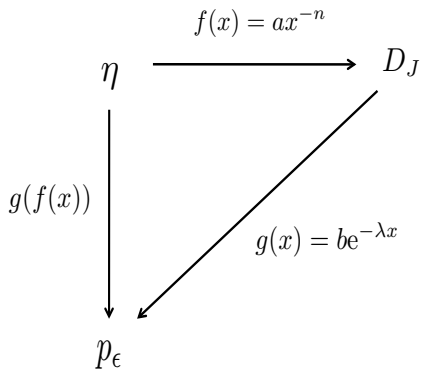

Figure 5.7: The relationship between the Dirichlet hyper-parameter η, the Jeffrey's divergence D_J, and the Bayes error rate p_ϵ.

Figure 5.8 provides strong evidence that the relationship between η and the Jeffrey's divergence D_J follows a power-law distribution. First, the log-log plot is almost linear (the least squares regression line is shown in red). As further evidence, if a random variable has a power-law distribution, then the log of the random variable follows an Exponential distribution. Thus, the log of the Jeffrey's divergence should be exponentially distributed. To test this hypothesis, we plot the quantiles of the log of the Jeffrey's divergence against the quantiles of an Exponential(1) distribution. This quantile-quantile plot is shown in the second plot in Figure 5.8. Note the strong linear relationship.

Composing the exponential function relating D_J and p_ϵ with the power-law function relating η and D_J, we see that the relationship between η and p_ϵ is given by the exponential function, $f(\eta) = ae^{-\lambda g(\eta)}$ where $g(\eta) = b\eta^{-n}$ follows a power-law distribution. This is an exponential function whose rate of decay is not constant. When η is small, $g(\eta)$ is large and thus $f(\eta)$ is small. Conversely, when η is large, $g(\eta)$ is small and thus $f(\eta)$ is large. This is consistent with Figure 5.9 which shows η versus p_ϵ for $L = 15, 250, 600, 1200$.

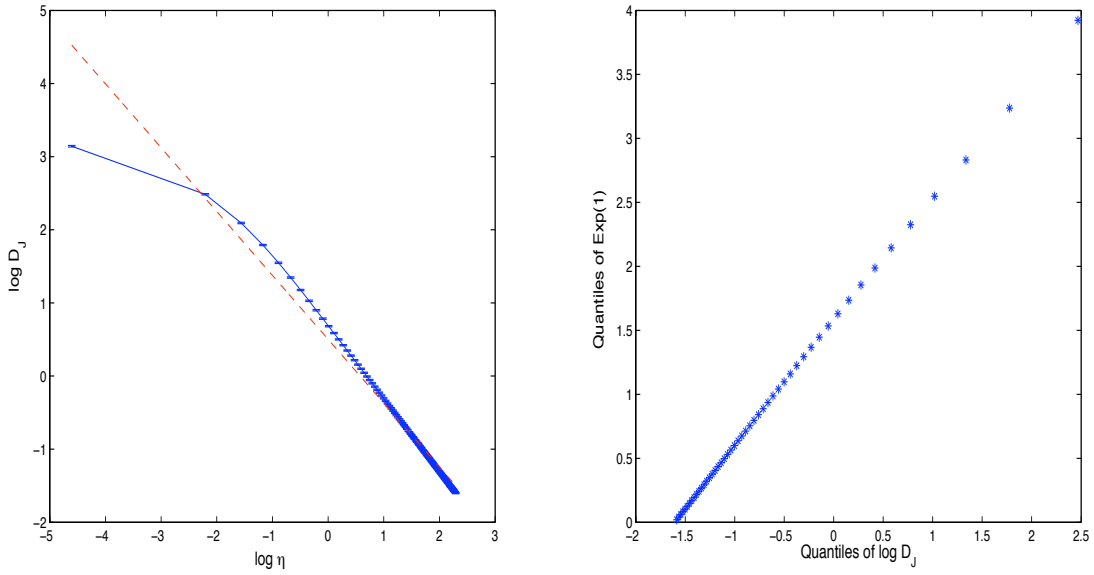

Figure 5.8: The first plot shows $\log D_J$ versus $\log(\eta)$ along with the best-fit linear approximation (dashed). The second plot shows the quantiles of $\log D_J$ plotted against the quantiles of an Exponential(1) distribution. Both plots suggest a power-law relationship between the Dirichlet hyper-parameter η and the Jeffrey's divergence D_J.

Document length L

Figure 5.10 shows the log of the Bayes error rate, $\log(p_\epsilon)$, plotted against the document length. Algorithm 3 in Appendix F shows the pseudocode used to generate this plot. The blue, red, and green lines correspond to $\eta = 0.1, 1.0, 10$ respectively.

It seems from the strong linear nature of the plot that as the document length increases, the Bayes error rate exponentially decays, where the rate of decay approaches zero as η approaches infinity. We can verify this assumption by taking the derivative of p_ϵ given in Equation 5.7 with respect to the document length L. Recall that $\theta = 0.5$ and thus $\log \frac{1-\theta}{\theta}$ is zero.

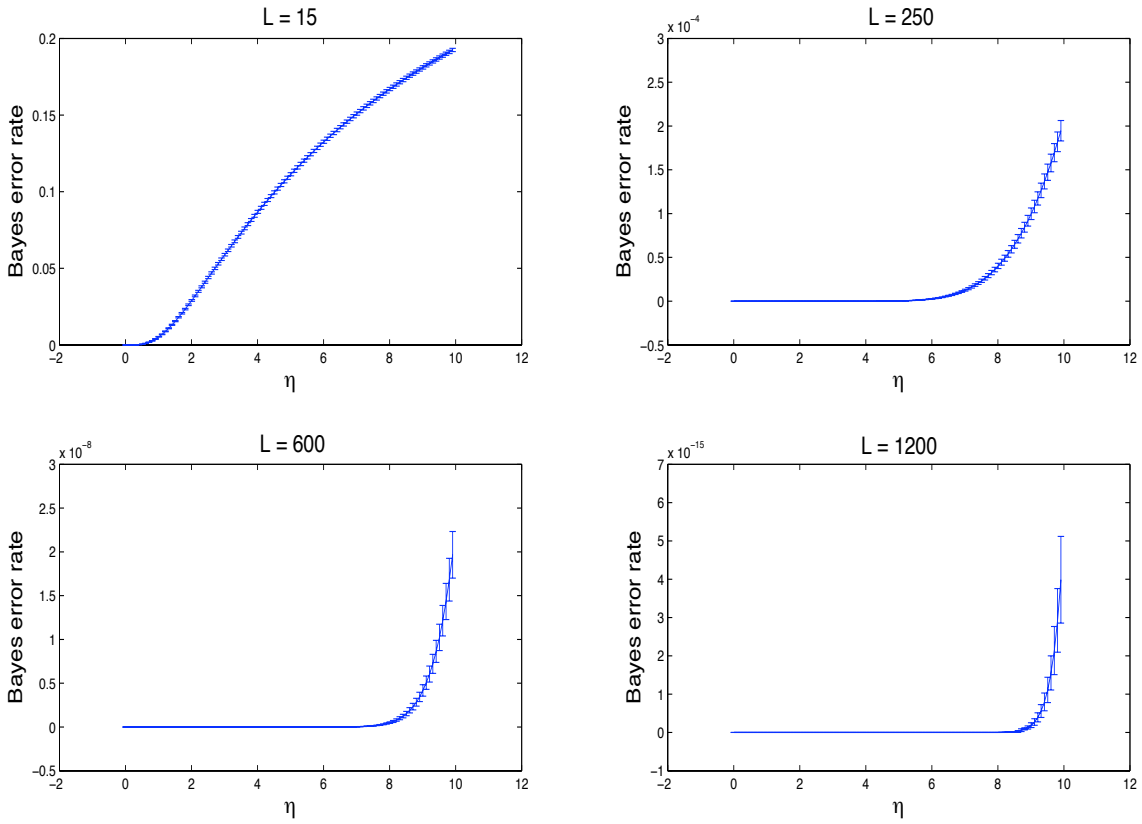

Figure 5.9: The Bayes error rate p_ϵ plotted against the Dirichlet hyper-parameter η for $L = 15, 250, 600, 1200$

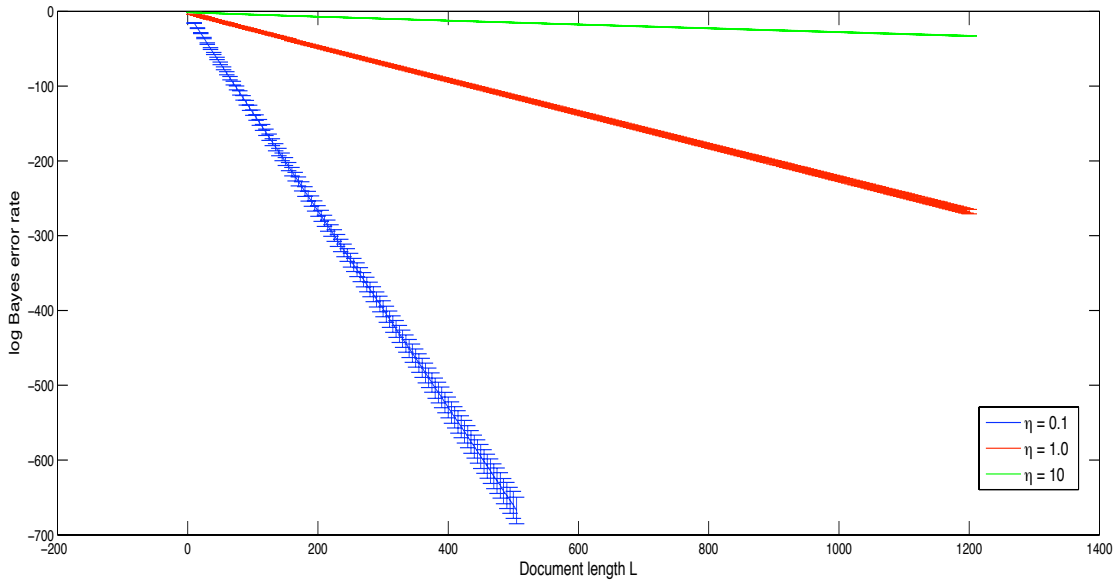

Figure 5.10: Semi-log plot of $\log(p_\epsilon)$ against the document length L

$$\frac{\partial p_\epsilon}{\partial L} = \frac{1}{2}\left(\frac{\partial \Phi\big(z_1\big)}{\partial z_1} \cdot \frac{\partial z_1}{\partial L} + \frac{\partial \Phi\big(z_2\big)}{\partial z_2} \cdot \frac{\partial z_2}{\partial L} \right) \qquad \text{where}$$

$$\begin{aligned} z_1 &= \frac{-\mu_1}{s_1} \\ &= \frac{-LD_{KL}(\phi_1||\phi_2)}{\sqrt{L}\sqrt{D_e(\phi_1||\phi_2) - D_{KL}(\phi_1||\phi_2)^2}} \\ &= \frac{-D_{KL}(\phi_1||\phi_2)}{\sqrt{D_e(\phi_1||\phi_2) - D_{KL}(\phi_1||\phi_2)^2}} \cdot \sqrt{L} \\ &\equiv a_1\sqrt{L} \qquad\qquad \text{and} \end{aligned}$$

$$\begin{aligned} z_2 &= \frac{\mu_2}{s_2} \\ &= \frac{-D_{KL}(\phi_2||\phi_1)}{\sqrt{D_e(\phi_2||\phi_1) - D_{KL}(\phi_2||\phi_1)^2}} \cdot \sqrt{L} \\ &\equiv a_2\sqrt{L} \end{aligned}$$

The derivative of the standard Normal cdf $\Phi(\cdot)$ is the standard Normal probability density function (pdf). Thus we have,

$$\begin{aligned} \frac{\partial p_\epsilon}{\partial L} &= \frac{1}{2}\left(\frac{\partial \Phi\big(z_1\big)}{\partial z_1} \cdot \frac{\partial z_1}{\partial L} + \frac{\partial \Phi\big(z_2\big)}{\partial z_2} \cdot \frac{\partial z_2}{\partial L} \right) \\ &= \frac{1}{2}\left(\frac{1}{\sqrt{2\pi}} \exp^{-\frac{1}{2}z_1^2} \cdot \frac{a_1}{2\sqrt{L}} + \frac{1}{\sqrt{2\pi}} \exp^{-\frac{1}{2}z_2^2} \cdot \frac{a_2}{2\sqrt{L}} \right) \\ &= \frac{1}{\sqrt{2\pi L}}\left(a_1 \exp^{-\frac{1}{2}a_1^2 L} + a_2 \exp^{-\frac{1}{2}a_2^2 L} \right) \\ &\approx \frac{2a}{\sqrt{2\pi L}} \exp^{-\frac{1}{2}a^2 L} \end{aligned} \tag{5.8}$$

where the last line is true for multinomials sampled from a symmetric Dirichlet prior. The divergence of ϕ_1 from ϕ_2 and the divergence of ϕ_2 from ϕ_1 for two multinomial distributions sampled from a symmetric Dirichlet distribution are approximately equal. This can be seen in the Monte Carlo simulations shown in Figure 5.3 and Figure 5.4. From the Monte Carlo simulations, we see that the red and blue histograms are approximately equidistant from 0 (the decision boundary) and have approximately the same variance. The Normal approximation tells us that this distance, i.e. the distance from the decision boundary, is proportional to the KL-divergence and the variance is proportional to the exponential divergence minus the squared KL-divergence.

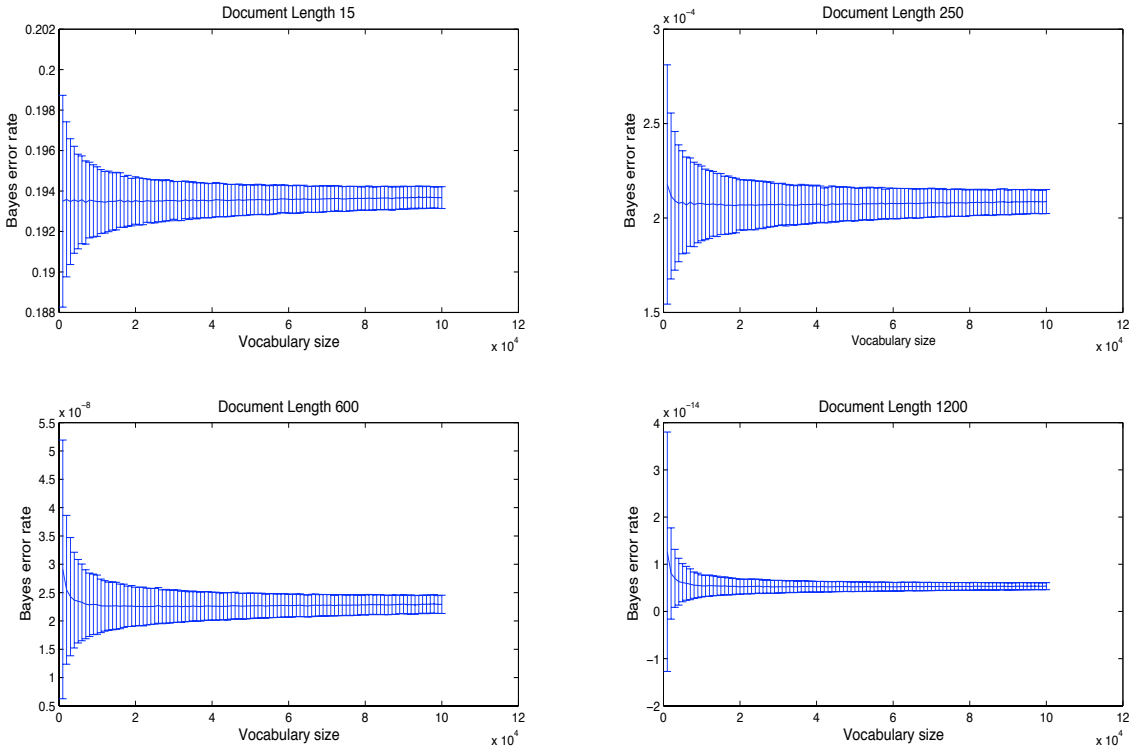

Figure 5.11: Plot of $\log(p_\epsilon)$ against the vocabulary size W for $\eta = 10$.

Thus, for symmetric η we see that as the document length L increases, the Bayes error rate decreases exponentially. Also note that as η increases the constant a (which affects the rate of decay) approaches zero. Equation 5.8 nicely captures the relationship between the document length L, the hyper-parameter η, and the rate of change of the Bayes error rate.

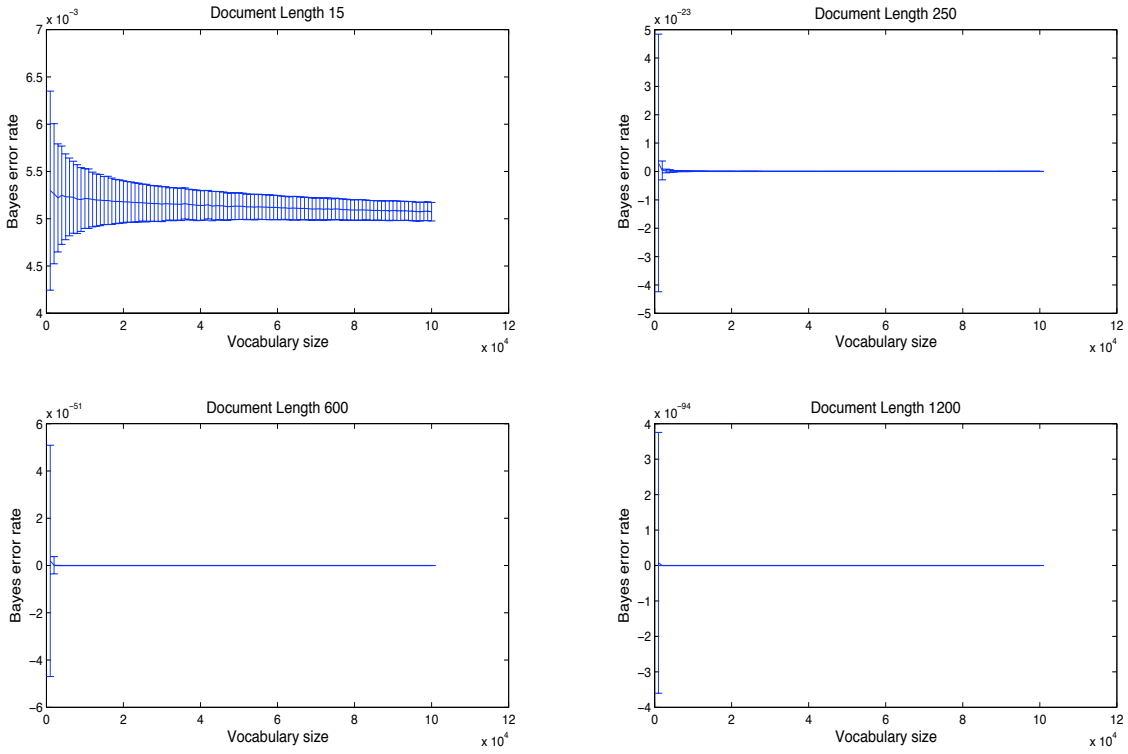

Figure 5.12: Plot of $\log(p_\epsilon)$ against the vocabulary size W for $\eta = 1.0$.

Vocabulary size W

Figure 5.11 shows the Bayes error rate as the vocabulary size ranges from $1,000$ words to $100,000$ words for a Dirichlet hyper-parameter of $\eta = 10$. Each plot represents a different document length. For a document length of 15 words, the Bayes error rate is nearly constant with a mean (across vocabulary sizes) of 0.19 and a standard deviation of 6.6×10^{-5}. For document lengths greater than 15, there is an increase of the Bayes error rate for vocabulary sizes less than $5,000$ words. For a vocabulary size larger than $5,000$ words we again see an almost constant Bayes error rate.

Figure 5.12 shows the Bayes error rate for Dirichlet hyperparameter $\eta = 1$. The Bayes error rate is effectively zero for all plots except the first, where the document length is 15 words. Figure 5.13 shows the Bayes error rate for $\eta = 0.1$. Again, the Bayes error rate is effectively zero for all plots except for a document length of 15 words and a vocabulary size of less than

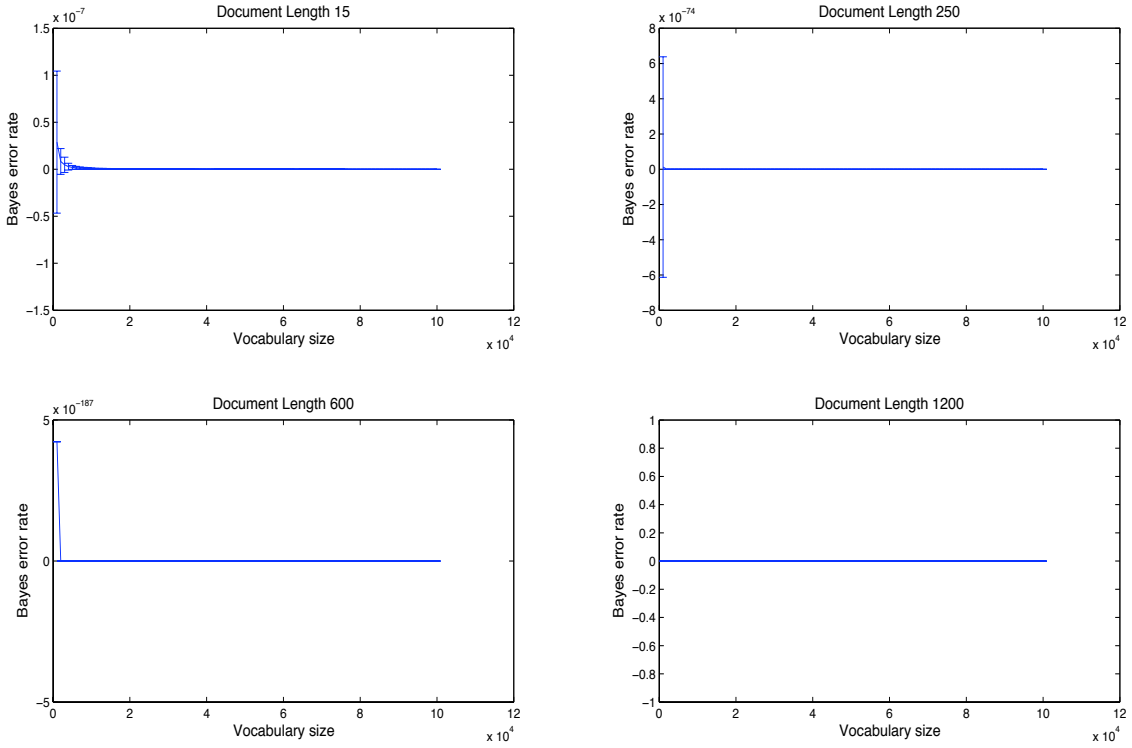

Figure 5.13: Plot of $\log(p_\epsilon)$ against the vocabulary size W for $\eta = 0.1$.

$5,000$ words.

It is interesting to note that one obtains a similar Bayes error rate for $\eta = 1$ with a document length of 15, and $\eta = 10$ with a document length between 150 and 200 words. In other words, increasing η by a factor of 10 (thus making it harder to distinguish between two multinomials) requires an increase of the document length by a factor of 10 to achieve roughly the same error rate.

Algorithm 4 in Appendix F shows the pseudocode used to generate these figures.

5.2.8 A note on real text data

Suppose we have an infinite collection of documents $D = \{(x_d, y_d) : y_d \in \{1, 2\}, d = 1, 2, \ldots\}$ where the length of each document is L. We make the modeling assumption that the doc-

uments in D arise from a multinomial Dirichlet mixture model[10]. We use D to compute the maximum a posteriori (MAP) estimate of θ and $\phi_{1:2}$. We then use the MAP estimates to classify each document using the log likelihood ratio test and compute the proportion of documents that were misclassified. We denote this error rate as $T(D; \phi_1, \phi_2)$.

Now suppose that we use the MAP estimates of θ and $\phi_{1:2}$ to generate an infinite collection of random variables $D^{\text{REP}} = \{(x_i, y_i) : y_i \in \{1, 2\}, i = 1, 2, \ldots\}$ which we then classify using the likelihood ratio test. We compute the proportion of multinomial random variables x_i that were misclassified. We denote this error rate as $T(D^{\text{REP}}; \phi_1, \phi_2)$. Note that in this case $T(D^{\text{REP}}; \phi_1, \phi_2)$ is the Bayes error.

Is it necessarily true that $T(D; \phi_1, \phi_2)$ is equal to $T(D^{\text{REP}}; \phi_1, \phi_2)$? The answer to this question is no. The error rate $T(D; \phi_1, \phi_2)$ is a function of both the error due to the overlap of the class likelihood functions (which is the Bayes error $T(D^{\text{REP}}; \phi_1, \phi_2)$) and the error introduced by our modeling assumptions [11].

Thus, we can compare the error rate achieved by the likelihood ratio test on real text data to the error rate achieved by the likelihood ratio test on data known to have come from the posited statistical model (i.e. the multinomial variables x_i) to gain insight into the extent to which actual text data deviates from the commonly used multinomial Dirichlet model. Note that we are performing a goodness-of-fit test, measuring how well the multinomial Dirichlet mixture model fits actual text data.

We apply this procedure to the Topic Detection and Tracking (TDT) corpus which contains $6,498$ news articles written between April and September of 2003. We chose this corpus because the articles were labeled according to the event discussed (as opposed to topical content) and because almost all articles are singly-labeled. There are a total of 246 events

[10] We also make the assumption that the class indicator variables y_d are noiseless

[11] Since we assumed the labels y_d were perfect and that we had an infinite amount of "training data" from which to estimate the parameters, there is no error that arises from insufficient or noisy data

in the TDT corpus.

The likelihood ratio test and the TDT corpus

Let e_1 and e_2 represent two events in the TDT corpus and $D = \{D_1, D_2\}$ the set of articles with label e_1 and e_2. We use the notation ϕ_{e_1} and ϕ_{e_2} to denote the parameters of the class likelihood functions and we let θ be the proportion of articles from e_1.

Obviously, we do not have an infinite set of documents D. Thus, there is some uncertainty about the "true" value of the class parameters ϕ_{e_1} and ϕ_{e_2}. As a result, instead of using a point-estimate to summarize the posterior distribution (e.g. the MAP estimate) we take J samples from the posterior distribution $p(\phi_{e_1}, \phi_{e_2}|D, \eta)$[12].

For each posterior sample, $\{\phi_{e_1}^{(j)}, \phi_{e_2}^{(j)}\}$ we generate a set of multinomial random variables D^{REP} (where REP stands for "replicated data"). D^{REP} contains one multinomial random variable x_i for each article x_d in D. Furthermore, the sum $\sum_{i=1}^{W} x_{ij}$ is equal to the number of words in x_d. In this way, D and D^{REP} are identical except that the random variables in D^{REP} are known to have come from the posited statistical model whereas the documents D are not.

Given D and D^{REP}, we compute the error rate achieved by the likelihood ratio test on D and the error rate achieved by the likelihood ratio test on D^{REP}. This procedure is summarized in Algorithm 1.

Recall that all of the articles in D (D^{REP}) must have the same length L. To accomplish this, we randomly select L words from each article in D (D^{REP}) and use these L words to compute the likelihood ratio.

[12] Also instead of fixing the value of η, we learn a non-symmetric Dirichlet prior to better model the data in D. Thus, we alternate between Gibbs sampling ϕ_{e_1} and ϕ_{e_2} given D and $\eta = (\eta_1, \ldots, \eta_W)$ and using gradient optimization to find η that maximizes the joint probability distribution $p(\phi_{e_1}, \phi_{e_2}, \eta|D)$

Algorithm 1: Our statistic of interest is the error rate of the likelihood ratio test

Input: D, $\phi_{e_1}^{(j)}$, $\phi_{e_2}^{(j)}$, L, θ
Output: $T(D; \phi_{e_1}, \phi_{e_2})$, $T(D^{\text{REP}}; \phi_{e_1}, \phi_{e_2})$

```
Shuffle(D) ;                                   // Shuffle the words in each article
miss ← 0
total ← 0
for i ∈ {1...N} do
    if x_i has length ≥ L then
        Compute ℓ(x_i) using first L words of x_i ;        // using Equation 5.4
        if ℓ(x_i) < log (1−θ)/θ and y_i = 1 then
            miss++
        if ℓ(x_i) > log (1−θ)/θ and y_i = 2 then
            miss++
        total++
T(D^REP; φ_1, φ_2) ← miss/total

D^REP ← GenerateReplicatedData(φ_e1, φ_e2, D)
Shuffle(D^REP) ;                  // Shuffle the words in each replicated article
miss ← 0
total ← 0
for i ∈ {1...N} do
    if x_i^REP has length ≥ L then
        Compute ℓ(x_i^REP) using first L words of x_i^REP ;   // using Equation 5.4
        if ℓ(x_i^REP) < log (1−θ)/θ and y_i = 1 then
            miss++
        if ℓ(x_i^REP) > log (1−θ)/θ and y_i = 2 then
            miss++
        total++
T(D^REP; φ_1, φ_2) ← miss/total
```

Once we have $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ and $T(D^{\text{REP}}; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ for $j \in \{1, \ldots, J\}$ we can compute the probability that $T(D^{\text{REP}}; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ exceeds $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ by counting the proportion of the J samples for which $T(D^{\text{REP}}; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ is greater than $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$. Note that it is now possible for the error rate on D to be less than the error rate on D^{REP} because the error rate on D^{REP} is no longer the Bayes error but an estimate of the Bayes error computed from the J samples.

Posterior predictive checks

The procedure described above is in fact a *posterior predictive check* [22] – a Bayesian analog to frequentist goodness-of-fit tests. Posterior predictive checks provide a means of estimating a posterior p-value, i.e. the probability that data known to have come from a posited statistical model is at least as extreme as observed data with respect to some statistic of interest. The statistic of interest is a real valued function of the data. When the statistic of interest is also a function of the parameters – e.g. ϕ_{e_1} and ϕ_{e_2} – then it is called a *discrepancy*. Gelman et al. [22] present a simple procedure for computing a posterior p-value. We present this procedure here:

Given a set of posterior samples θ^j, $j = 1, \ldots, J$

1. Draw a replicated data set $y^{\text{rep},j}$ from the distribution $p(y^{\text{rep}}|\theta^j, H)$
2. Calculate $T(y; \theta^j)$ and $T(y^{\text{rep},j}; \theta^j)$

Estimate the posterior p-value by the proportion of $T(y^{\text{rep},j}; \theta^j) \geq T(y; \theta^j)$

Table 5.2: Procedure for estimating a posterior predictive p-value taken from Gelman et al. [22]

where y is the observed data, y^{rep} the replicated data, H the statistical model, and θ (the model parameters) are sampled from the posterior distribution $p(\theta|H, y)$. Note that this is the exact procedure that we developed. Thus, we are in fact performing a posterior predictive check which measures the goodness-of-fit of the multinomial Dirichlet mixture model to the TDT articles with respect to the discrepancy given by Algorithm 1.

Examples taken from the TDT corpus

The first pair of events that we examine are event $e_1 = 55072$ (*Court indicts Liberian president*) which occurred in June of 2003, and event $e_2 = 55089$ (*Liberian former president arrives in exile*) which occurred in August of 2003. The posterior samples of ϕ_{e_1} and ϕ_{e_2} had

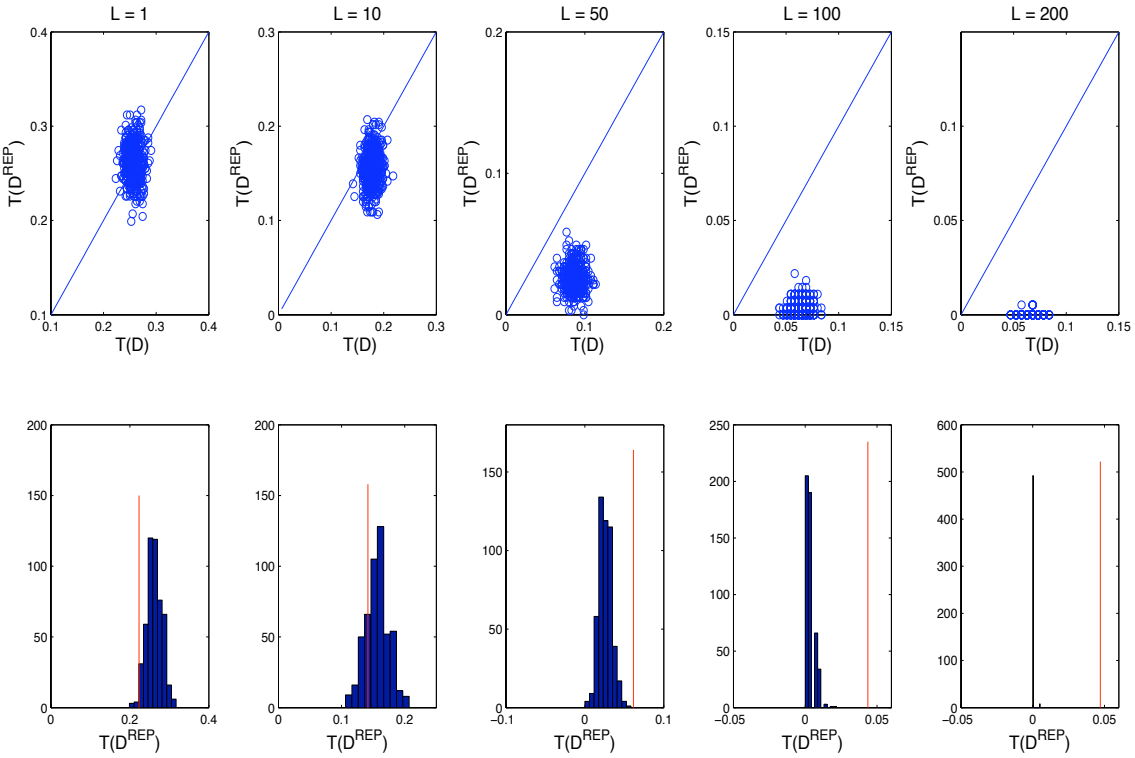

Figure 5.14: (Liberian president) The first row shows scatterplots of $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ versus $T(D^{\text{REP}}; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ for $J = 500$ posterior samples. Each column corresponds to a different document length L. The proportion of circles above the $y = x$ line is an estimate of the posterior predictive p-value. The second row shows the reference distribution, that is a histogram of the $T(D^{\text{REP}}; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ values. The red line in each plot is the minimum value of $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ over all J posterior samples.

the smallest Jeffrey's divergence in the TDT corpus $D_J = 3.101$.

The first row of Figure 5.14 shows scatterplots of $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ versus $T(D^{\text{REP}}; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ for different article lengths L using $J = 500$ posterior samples. The proportion of circles above the $y = x$ line is an estimate of the posterior predictive p-value. The second row shows a histogram of the 500 $T(D^{\text{REP}}; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ values. The red line in each plot is the **minimum** value of $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ over all 500 posterior samples.

Figure 5.15 shows a plot of the discrepancy $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ and $T(D^{\text{REP}}; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ averaged over the $J = 500$ posterior samples for L ranging from 1 to 400 words. The average discrepancy for the replicated articles is shown in blue; the average discrepancy for the TDT articles

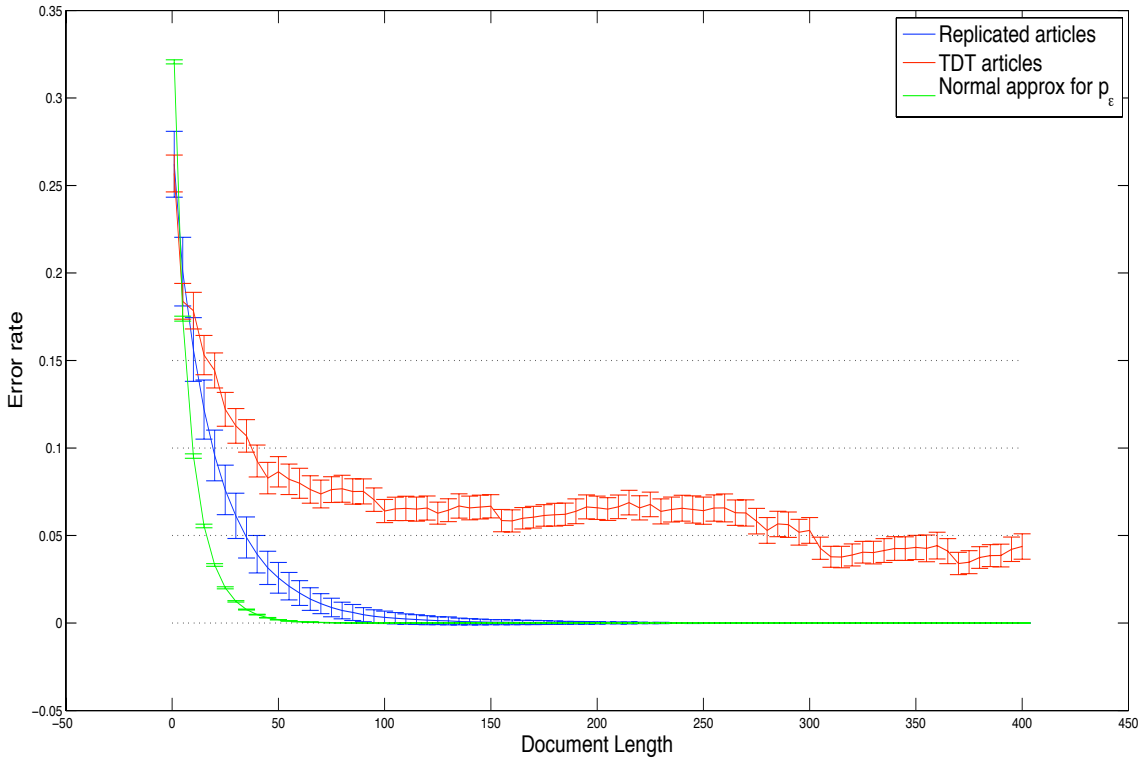

Figure 5.15: (Liberian president) The average error rate (with one standard deviation) over the J posterior samples for D (red) and D^{REP} (blue). Document length varies from 1 to 400 words.

is shown in red. We have also plotted an estimate of the Bayes error using our approximation (Equation 5.7) as a reference line. This line is shown in green.

From the scatter plots, we estimate a posterior predictive p-value of 0.28, 0.01, 0.00, 0.00, and 0.00 for $L = 1, 10, 50, 100, 200$ words respectively. A posterior predictive p-value of zero indicates that the error rate achieved on the TDT articles is highly unlikely to have arisen under a multinomial Dirichlet mixture model. In particular, documents tend to violate the independence assumption made by the multinomial Dirichlet mixture model which assumes conditional independence of the words in a document given the class multinomial parameters. Madsen et al. demonstrate empirically that this assumption does not hold true for actual text data [39]. The effect of this violation of the independence assumption becomes more noticeable as L increases.

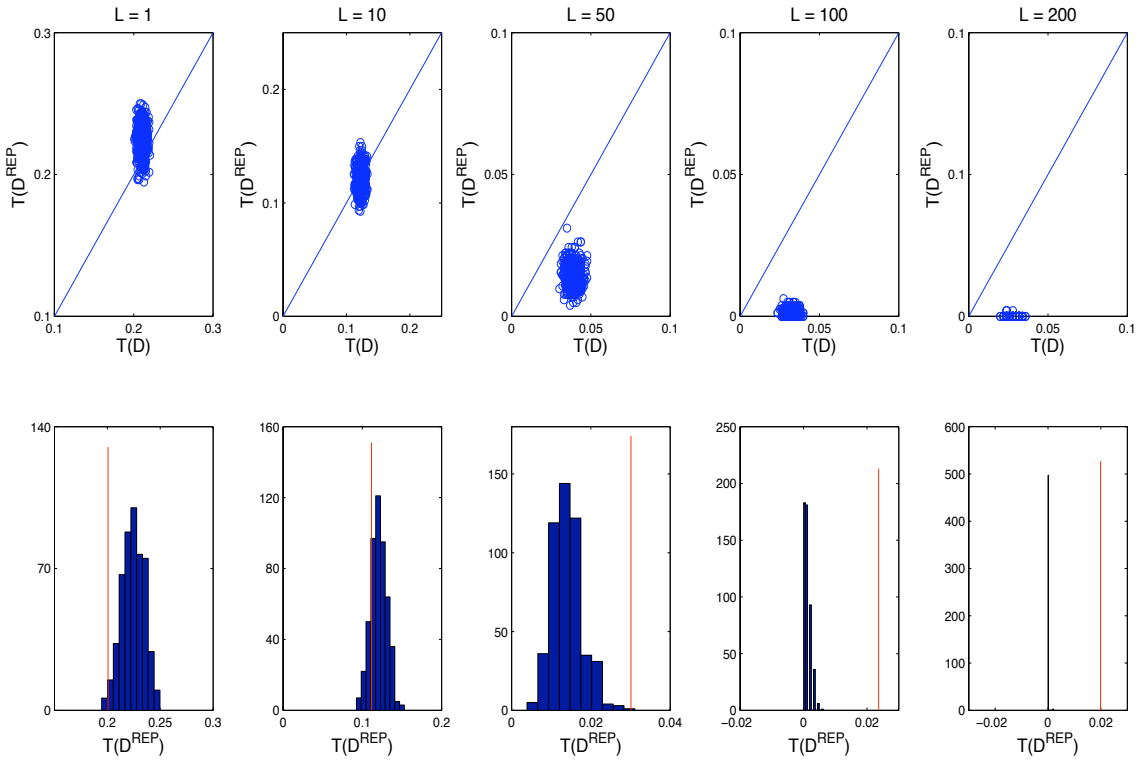

Figure 5.16: (Bombings) The first row shows scatterplots of $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ versus $T(D^{\text{REP}}; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ for $J = 500$ posterior samples. Each column corresponds to a different document length L. The proportion of circles above the $y = x$ line is an estimate of the posterior predictive p-value. The second row shows the reference distribution, that is a histogram of the $T(D^{\text{REP}}; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ values. The red line in each plot is the minimum value of $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$ over all J posterior samples.

From both of these figures, we see that the error rate achieved on the replicated articles rapidly approaches zero as L increases. This is consistent with our findings in the previous section – namely the Bayes error rate exponentially decays as the document length L increases. However, this is not the case with the error rate achieved on the actual text articles. In Figure 5.15, long after the error rate of the replicated articles reaches zero, the error rate on the actual text articles is still greater than 5%.

Figure 5.16 and Figure 5.17 show the same plots for event $e_1 = 55107$ (*Casablanca bombs*) and event $e_2 = 55106$ (*Bombings in Riyadh, Saudi Arabia*). The posterior samples for ϕ_{e_1} and ϕ_{e_2} have a larger Jeffrey's divergence of $D_J = 3.75$. We see the same trends. The error

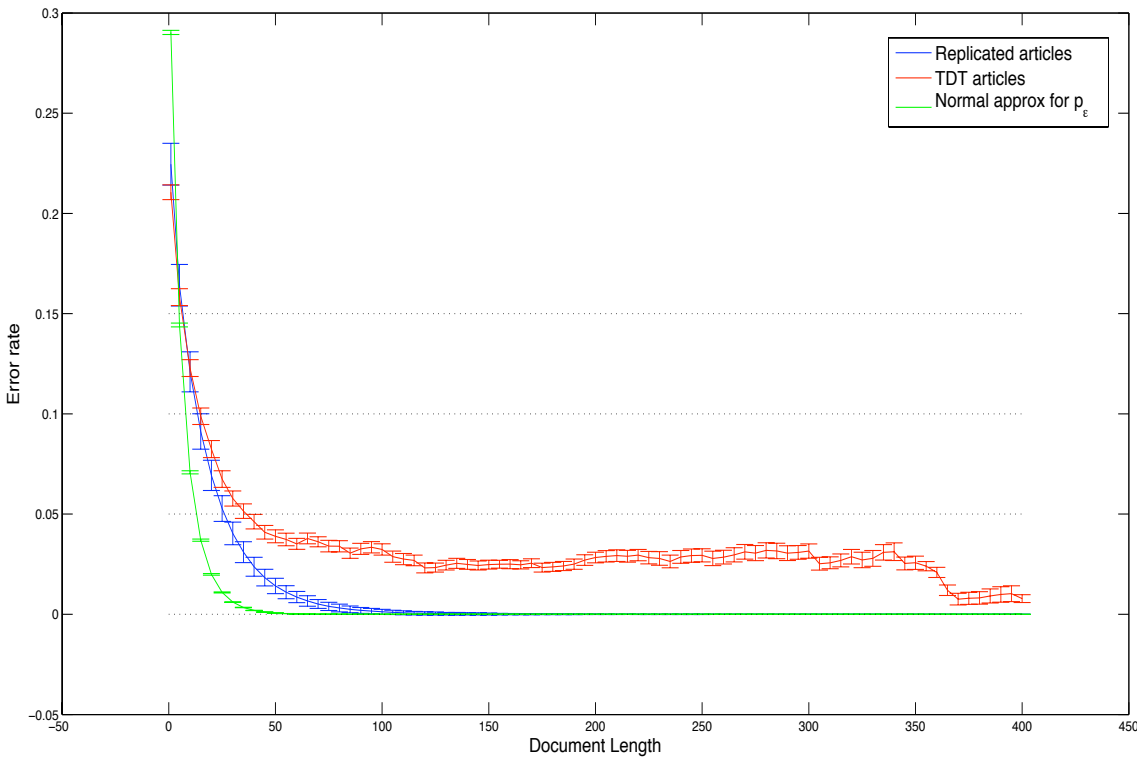

Figure 5.17: (Bombings) The average error rate (with one standard deviation) over the J posterior samples for D (red) and D^{REP} (blue). Document length varies from 1 to 400 words.

rate achieved on the replicated data approaches zero at a faster rate than the error rate achieved on the actual text data. The estimated posterior predictive p-values are again zero for $L = 50$, $L = 100$, and $L = 200$ words. One difference however is that the discrepancy between the replicated and actual data is not as great as it was for the previous pair of events. In Figure 5.17 we see that when the error rate on D^{REP} reaches zero, the error rate on D is still non-zero but less than 5%.

Finally, Figure 5.18 and Figure 5.19 show the same plots for event 55107 (*Casablanca bombs*) and 55029 (*Swedish Foreign Minister killed*) whose posterior class multinomial parameters have an even larger Jeffrey's divergence of $D_J = 6.29$. In this case, the error rate achieved on the replicated data D^{REP} and the error rate achieved on the actual text data D approaches zero at the same rate and is comparable for all values of L. Thus, we see that as the Jeffrey's divergence increases, i.e. the overlap of the class likelihood functions decreases, the effect

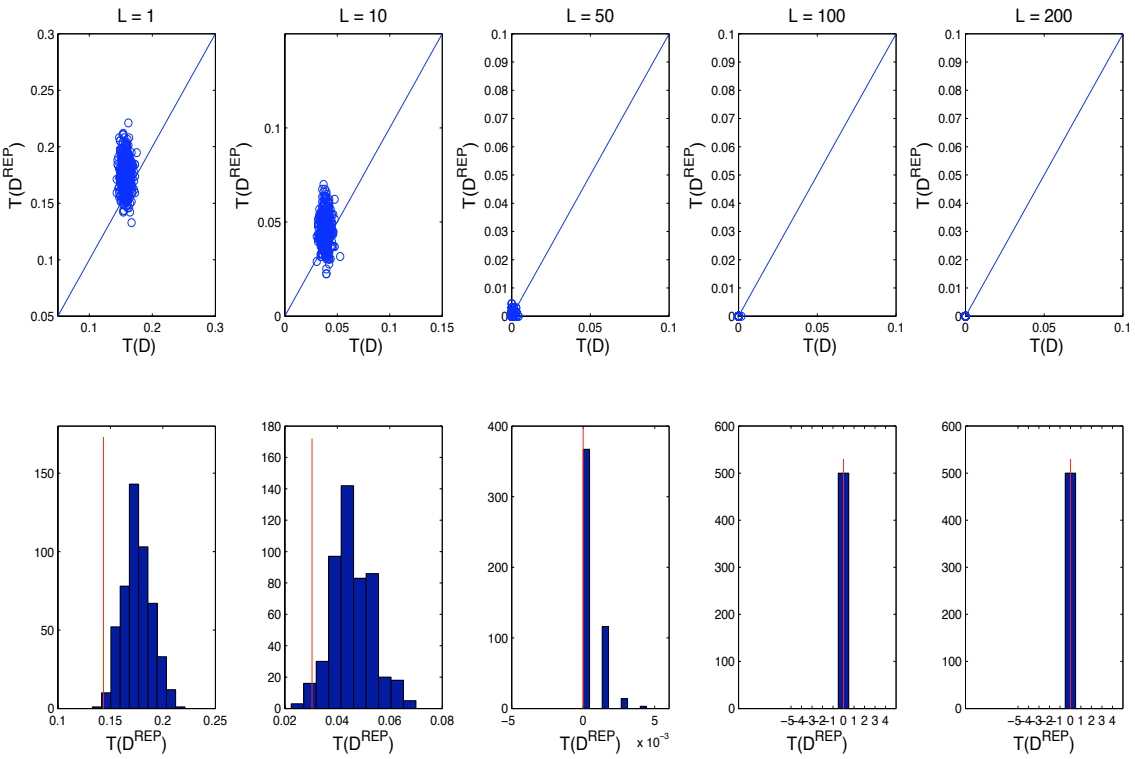

Figure 5.18: (Bombing and Swedish foreign minister) The first row shows scatterplots of $T(D; \phi_{e1}^{(j)}, \phi_{e2}^{(j)})$ versus $T(D^{\text{REP}}; \phi_{e1}^{(j)}, \phi_{e2}^{(j)})$ for $J = 500$ posterior samples. Each column corresponds to a different document length L. The proportion of circles above the $y = x$ line is an estimate of the posterior predictive p-value. The second row shows the reference distribution, that is a histogram of the $T(D^{\text{REP}}; \phi_{e1}^{(j)}, \phi_{e2}^{(j)})$ values. The red line in each plot is the minimum value of $T(D; \phi_{e1}^{(j)}, \phi_{e2}^{(j)})$ over all J posterior samples.

of the violation of the independence assumption by the actual text data becomes of less importance. If the modeling assumptions made by the multinomial Dirichlet mixture model held true for actual text data then we would have seen this same trend (i.e. the error rate achieved on D^{REP} and the error rate achieved on D approaching zero at the same rate) for all three examples. However, this only occurs in proportion to the magnitude of the Jeffrey's divergence. This fits our intuition that when two classes are highly dissimilar many classifiers can achieve a low error rate regardless of their inductive biases.

We applied this procedure to every pair of events in the TDT corpus for $L = 100$ words. That is, for every pair of events we sampled $J = 500$ posterior samples $\{\phi_{e1}^{(j)}, \phi_{e2}^{(j)}\}$ and for

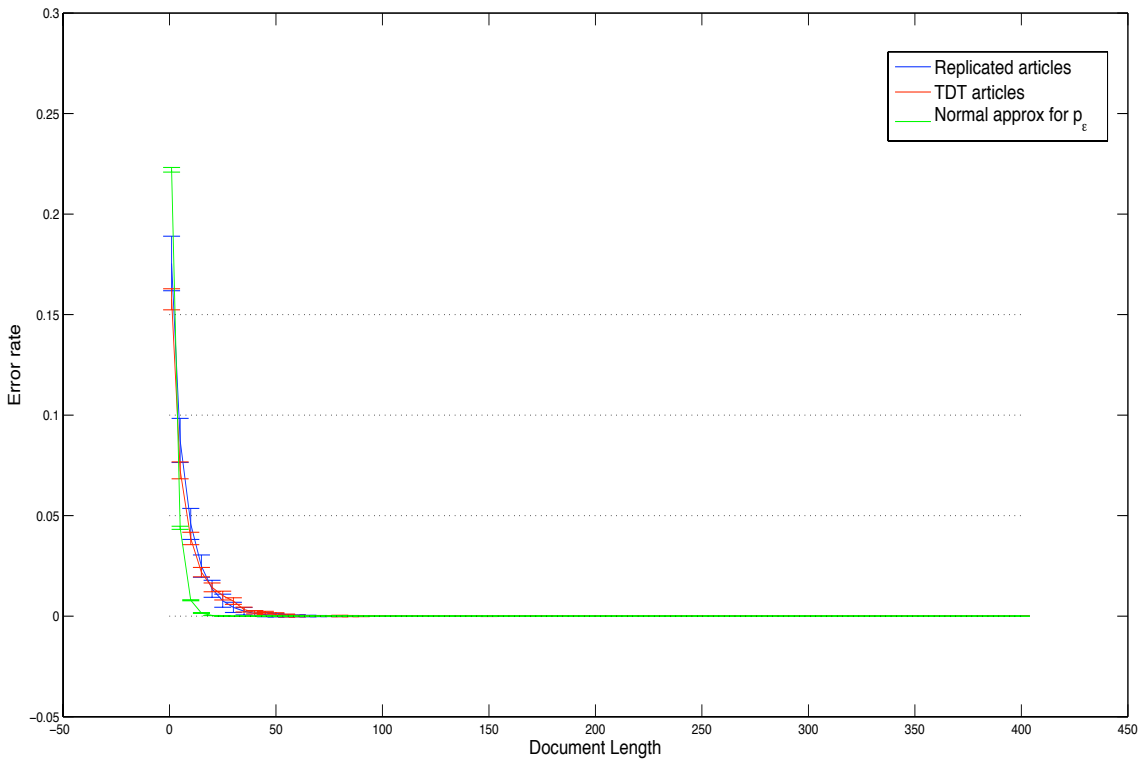

Figure 5.19: (Bombing and Swedish foreign minister) The average error rate (with one standard deviation) over the J posterior samples for D (red) and D^{REP} (blue). Document length varies from 1 to 400 words.

each posterior sample we ran Algorithm 1 with $L = 100$. Figure 5.20 shows the results.

Figure 5.20 shows the log of the error rate as a function of the Jeffrey's divergence. The error rate on the replicated data D^{REP} was zero for all pairs of events and as such is not shown in Figure 5.20. The error rate on the TDT data D was zero for all pairs of events except 10 pairs which are show in red. The error rate approximated using Equation 5.7 is shown in green. We also show the least squares regression line for both the error rate on D and the approximate error rate.

We see from our approximation of the Bayes error (shown in green) that as the Jeffrey's divergence increases the error rate decreases. This is consistent with our findings up to this point. Second, we see that the error rate on the TDT articles D shows the same relationship between the Jeffrey's divergence and the true error. Finally, we note that the error rate

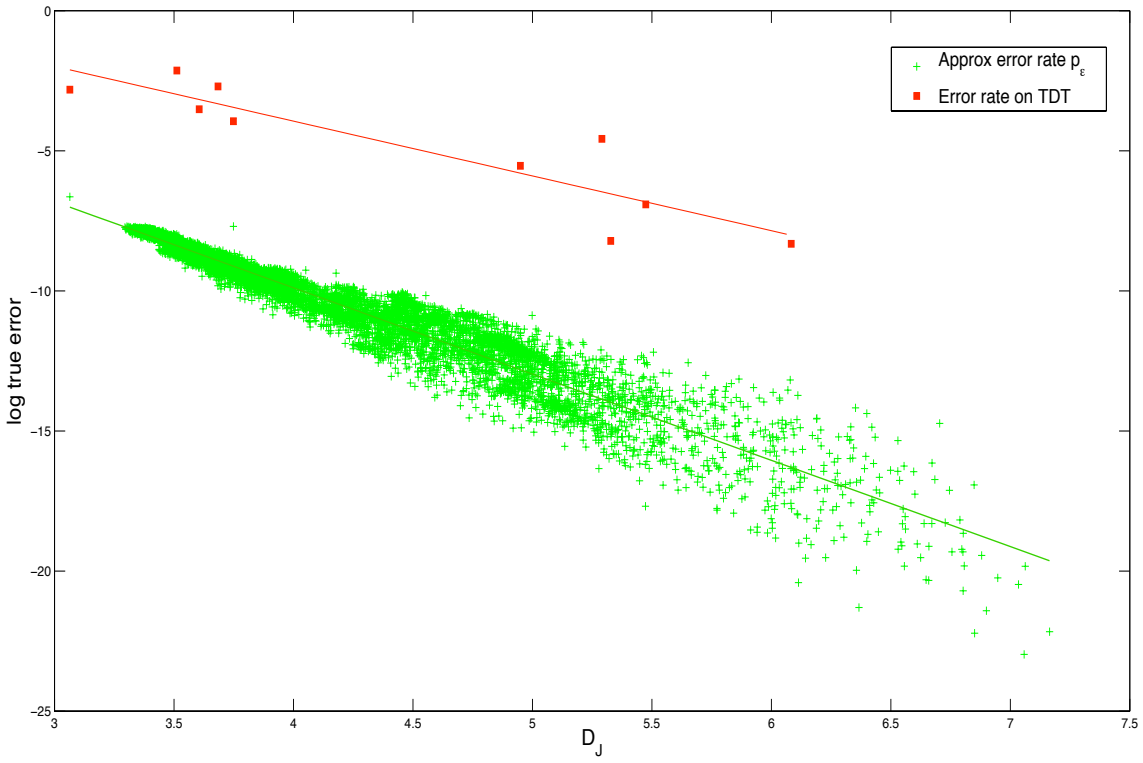

Figure 5.20: This plot shows the error rate achieved by the likelihood ratio test for **all** pairs of events in the TDT corpus using $L = 100$ words. The error rate achieved on the TDT articles was zero except for 10 pairs of events shown by the red boxes. The error rate achieved on the replicated data was zero for all pairs of events. The error rate given by the Normal approximation is shown in green.

on the TDT articles D, although showing the same exponential trend, is higher than the approximate error rate. This illustrates the extra error that arises from the lack-of-fit between the multinomial Dirichlet mixture model assumption and the TDT articles.

Finally, we may question why the error rate on the TDT data D was zero for all except 10 pairs of events. Obviously for those pairs of events with large Jeffrey's divergences (i.e. pairs of events similar to *Casablanca bombs* and *Swedish Foreign Minister killed*) the error rate was zero due to the dissimilarity between the classes. However, in general, we discovered that many of the events in the TDT corpus have mostly articles with fewer than 100 words. For those pairs of events with very few articles of 100 words or longer, the error rate was zero simply due to the lack of data. Table 5.3 shows pairs of events in the TDT corpus

D_J	e_1 ($\lvert D_{e_1}\rvert$) and e_2 ($\lvert D_{e_2}\rvert$)	$T(D)$
3.101	Liberian former president exile (85), Liberian president indicted (41)	0.06
3.486	Sweden rejects Euro (56), Sweden Minister killed (190)	0.12
3.548	British soldiers attacked in Basra(5), UN official killed in attack (171)	0.03
3.604	Morocco death sentence for bombers (0), USS Cole attack suspects escape (1)	0.0
3.686	Morocco death sentence for bombers (0), Casablanca bombs (123)	0.07
3.712	Casablanca bombs (123), Bombing in Saudi Arabia (320)	0.02
3.767	Man opens fire at Case Western (2), Explosion at Yale(1)	0.0
3.790	USS Cole attack suspects escape (1), British soldiers attacked in Basra (5)	0.0
3.794	Singapore removed from SARS list (3), SARS quarantine in Taiwan (23)	0.0
3.746	US troops fire on Mosul crowd (22), British soldiers attacked in Basra (5)	0.0
4.950	Bombing in Saudi Arabia (320), World economic forum in Jordan (46)	0.004
5.291	Spanish elections (23), Casablanca bombs (123)	0.01
5.328	UN official killed in attack (171), Palestine next prime minister (51)	0.0003
5.475	Bombing in Saudi Arabia (320) Palestine next prime minister (51)	0.001
6.083	Sweden Minister killed (190), World economic forum in Jordan(46)	0.0002

Table 5.3: Pairs of TDT events listed by increasing Jeffrey's divergence. For each event, we show in parenthesis the number of articles with length at least 100 words. We show in the last column the error rate achieved by the likelihood ratio test (for $L = 100$) on the TDT data, i.e. $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$.

ranked by their Jeffrey's divergence. For each event we show in parenthesis the number of corresponding articles with 100 words or more. The last column shows the error rate achieved by the likelihood ratio test on the TDT articles, i.e. $T(D; \phi_{e_1}^{(j)}, \phi_{e_2}^{(j)})$.

We see from Table 5.3 that many of the pairs of events with small Jeffrey's divergence had few articles greater than 100 words and as a consequence achieved an error rate of zero on the actual text data. Similarly, we see that many of the pairs of events with larger Jeffrey's divergence that had many articles greater than 100 words achieved a non-zero error rate on the actual text data.

The results from this section indicate that we can gain insight into the degree to which actual text data deviates from the assumptions of the multinomial Dirichlet mixture model by comparing the error rate achieved by the likelihood ratio test on actual text data to the error rate achieved by the likelihood ratio test on data known to have come from the posited statistical model. The disparity between these two error rates gives us insight into how

far the text data breaks the modeling assumptions – namely the conditional independence of the words given the class multinomial parameters. This violation of the independence assumption by actual text data has been noted before [57, 39].

5.3 Scenario 2: unknown multinomial parameters

When classifying text it is rare to observe the class multinomial parameters ϕ_k for $k \in \{1, 2\}$. Instead, one typically observes a set of representative documents from each class. Let $x^{(1)}$ and $x^{(2)}$ denote a set of representative documents from class 1 and class 2 respectively.

In this case, one option is to estimate the class parameters ϕ_k using the representative documents. For example, we can use the mode of the posterior distribution $p(\phi_k|x^{(1)}, x^{(2)}, \eta)$ to estimate the class multinomial parameters – this is called a maximum a posteriori (MAP) estimate. We can then classify a new document x by using the MAP estimates to compute the log likelihood ratio (and comparing this value to the log prior odds). Note that this classification rule is no longer guaranteed to achieve the Bayes error rate since we are using estimates of the true class parameters. In this approach, the estimation and the classification are broken up into two steps.

A second option is to represent our uncertainty about the class parameters by marginalizing them out from the joint distribution of our statistical model,

$$\begin{aligned} p(x, y, \phi_1, \phi_2|x^{(1)}, x^{(2)}, \eta, \theta) &= p(y|\theta) \cdot \int_{\phi_{1:2}} p(x|\phi_{1:2}, y)\ p(\phi_{1:2}|x^{(1)}, x^{(2)}, \eta)\ d\phi_{1:2} \\ &= p(y|\theta) \cdot p(x|y, x^{(1)}, x^{(2)}, \eta) \end{aligned} \tag{5.9}$$

where $p(x|y, x^{(1)}, x^{(2)}, \eta)$ is called the *marginal likelihood*. This quantity is also known as the *posterior predictive distribution*, i.e. the distribution over new data (x) given a set of observations ($x^{(1)}$ and $x^{(2)}$). Instead of using a point estimate of the class parameters (e.g. the MAP estimates) we are using the entire posterior distribution over $\phi_{1:2}$. We have also combined the "estimation" and the classification into one step. That is, to classify a new document x we take the optimal strategy from a Bayesian viewpoint and compute the posterior distribution over y given the documents x, $x^{(1)}$, and $x^{(2)}$, and the parameters η and θ. As in Section 5.2, this reduces to a comparison of the ratio of the marginal likelihoods[13] (i.e. the marginal likelihood of x under the hypothesis $y = 1$ to the marginal likelihood of x under the hypothesis $y = 2$) to the prior odds $\frac{1-\theta}{\theta}$. This is the approach we analyze in this section.

In the remainder of this section, we derive the classification rule based on the ratio of the marginal likelihoods. We present an interpretation of this classification rule that elucidates how evidence is accumulated in favor of both hypotheses $y = 1$ and $y = 2$. We then derive an expression for the average expected error rate for this classification rule[14]. We refer to this average expected error rate as the *Bayesian classifier error rate*. Note that the Bayesian classifier error rate is lower bounded by the Bayes error rate (which is attained when the classifier has perfect knowledge of the class parameters as in Section 5.2). We again present Monte Carlo simulations of the Bayesian classifier error rate and then appeal to the same central limit theorem to compute a Normal approximation. Finally, we investigate the relationship between the Bayesian classifier error rate (as given by the Normal approximation) and the model parameters.

[13]Note that the ratio of the marginal likelihoods is often termed the *Bayes factor* [30]. However, this terminology is traditionally used when comparing two structurally different models – e.g. comparing a multinomial likelihood function with a multivariate Bernoulli likelihood function – or two models with a different number of parameters. In this case, the two models differ only in the value of y. Thus, we do not use the terminology Bayes factor but instead refer to the *marginal likelihood ratio*

[14]Also referred to as the *true error* in the statistical pattern recognition literature [76].

$$\eta \in \mathbb{R}_{>0},\ \theta \in (0,1)$$
$$\phi_i \sim \text{Dirichlet}(\eta) \text{ for } i \in \{1,2\}$$
$$y^{ij} \sim \text{Discrete}(\theta, 1-\theta)$$
$$x^{ij} \sim \text{Multinomial}(\phi_{y_{ij}}, L_{ij})$$
$$y \sim \text{Discrete}(\theta, 1-\theta)$$
$$x \sim \text{Multinomial}(\phi_y, L)$$

Table 5.4: The multinomial Dirichlet mixture model for Scenario 2

5.3.1 Notation and the generative model

In this section, we have three sets of documents: the set of representative documents from class 1 denoted $x^{(1)}$, the set of representative documents from class 2 denoted $x^{(2)}$, and the document to be classified x.

Let $x^{(1)} \equiv \{x^{11}, x^{12}, \ldots, x^{1N}\}$ be a random sample of N documents generated by ϕ_1 and $x^{(2)} \equiv \{x^{21}, x^{22}, \ldots, x^{2M}\}$ a random sample of M documents generated by ϕ_2. The class indicators for the documents in $x^{(1)}$ and $x^{(2)}$ are known. That is, $y^{1j} = 1$ for $j \in \{1, \ldots, N\}$ and $y^{2j} = 2$ for $j \in \{1, \ldots, M\}$ where y^{ij} is the class indicator for document x^{ij}.

Each document x^{ij} is itself a vector of word counts that has dimension W, $x^{ij} = (x_1^{ij}, x_2^{ij}, \ldots, x_W^{ij})$ where x_w^{ij} is the number of times the wth word in the vocabulary occurs in the document x^{ij}. We use the superscripts ij to indicate the class and document respectively and the subscript w to indicate the word.

It is sufficient to know how many times the wth word in the vocabulary occurs in the *entire set* of representative documents $x^{(1)}$ or $x^{(2)}$. We denote these counts as,

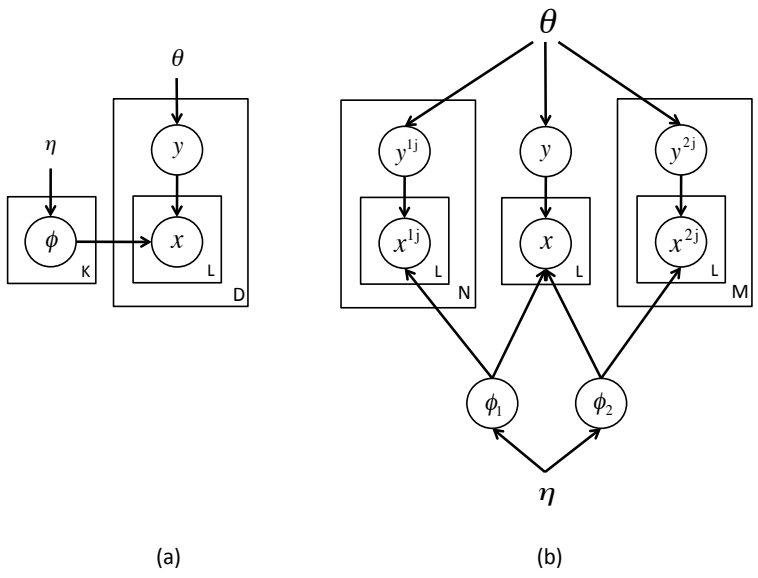

Figure 5.21: (a) The plate notation for the multinomial Dirichlet mixture model. For the scenario considered in this section, $D = M + N + 1$ (b) The same plate notation where $x^{(1)}$, $x^{(2)}$ and x are shown explicitly

$$x_w^1 \equiv \sum_{j=1}^{N} x_w^{1j} \quad \text{and}$$
$$x_w^2 \equiv \sum_{j=1}^{M} x_w^{2j}$$

Note that x_w^1 and x_w^2 have only one superscript to indicate the class. It is also sufficient to know the total number of words in $x^{(1)}$ and $x^{(2)}$ which we denote as

$$L_1 \equiv \sum_{w=1}^{W} x_w^1$$
$$L_2 \equiv \sum_{w=1}^{W} x_w^2$$

Table 5.4 shows the generative model and Figure 5.21 shows the plate notation for the multinomial Dirichlet mixture model. The generative model is still the multinomial Dirichlet mixture model. The only difference is that we know observe the class indicator variables for the sets of documents $x^{(1)}$ and $x^{(2)}$. Table 5.5 summarizes the important notation for this section. Recall that θ and η are fixed and known.

$x^{(1)}$	N representative documents generated by ϕ_1
$x^{(2)}$	M representative documents generated by ϕ_2
x_w^1	Number of times word w occurs in the set $x^{(1)}$
x_w^2	Number of times word w occurs in the set $x^{(2)}$
L_1	Total number of words in the set $x^{(1)}$
L_2	Total number of words in the set $x^{(1)}$
x	Document to be classified
x_w	Number of times word w occurs in document x
L	Total number of words in document x
y	The true assignment of x
$\ell(x; x^{(1)}, x^{(2)})$	The log of the marginal likelihood ratio

Table 5.5: Notation

5.3.2 Classification rule

We classify the document x by appealing to the posterior distribution of the latent class indicator variable y conditioned on the documents x, $x^{(1)}$, and $x^{(2)}$, and the parameters θ and η. This posterior distribution is given by,

$$
\begin{aligned}
q_k(x; x^{(1)}, x^{(2)}) &= p(y = k|x, x^{(1)}, x^{(2)}, \eta, \theta) \\
&= \frac{p(x|x^{(1)}, x^{(2)}, y = k, \eta, \theta)\ p(y = k|x^{(1)}, x^{(2)}, \eta, \theta)}{p(x|x^{(1)}, x^{(2)}, \eta, \theta)} \\
&= \frac{p(x|x^{(k)}, \eta)\ p(y = k|\theta)}{p(x|x^{(1)}, x^{(2)}, \eta)}
\end{aligned}
$$

for $k \in \{1, 2\}$. We use the notation $q_k(x; x^{(1)}, x^{(2)})$ to reinforce the idea that x is the random quantity whereas $x^{(1)}$ and $x^{(2)}$ are assumed known and fixed. Note that the latent indicator variable y is conditionally independent of $x^{(1)}$ and $x^{(2)}$, i.e. $p(y = k|x^{(1)}, x^{(2)}, \eta, \theta) = p(y = k|\theta)$. This conditional independence can be read from the plate notation in Figure 5.21 (b)[15]. Finally, note that the denominator is independent of y and can be ignored. Given this posterior distribution over y, we use the classification rule:

> **Classification rule:** If $q_1(x; x^{(1)}, x^{(2)}) > q_2(x; x^{(1)}, x^{(2)})$ then we classify x as belonging to class 1. If $q_1(x; x^{(1)}, x^{(2)}) < q_2(x; x^{(1)}, x^{(2)})$ then we classify x as belonging to class 2. Otherwise, we make a random decision.

As before, it will often be more convenient to work in log space, so we rewrite the inequality $q_1(x; x^{(1)}, x^{(2)}) > q_2(x; x^{(1)}, x^{(2)})$ in terms of the log function and restate the classification

[15]The class indicator y is independent of $\{x^{(1)}, x^{(2)}\}$ if we do **not** condition on x since any path from y to a document in the set $\{x^{(1)}, x^{(2)}\}$ must either contain the edges $y \to x \leftarrow \phi_k$ or the edge $y \to \theta$. In the former case, the path is blocked since x has converging arrows and x is not in the conditioning set. In the latter case, the path is blocked by θ since θ is in the conditioning set.

rule.

$$
\begin{aligned}
q_1(x; x^{(1)}, x^{(2)}) &> q_2(x; x^{(1)}, x^{(2)}) && \text{iff} \\
p(x|x^{(1)}, \eta)p(y = 1|\theta) &> p(x|x^{(2)}, \eta)p(y = 2|\theta) && \text{iff} \\
\frac{p(x|x^{(1)}, \eta)}{p(x|x^{(2)}, \eta)} &> \frac{p(y = 2|\theta)}{p(y = 1|\theta)} && \text{iff} \\
\log p(x|x^{(1)}, \eta) - \log p(x|x^{(2)}, \eta) &> \log p(y = 2|\theta) - \log p(y = 1|\theta)
\end{aligned}
\tag{5.10}
$$

The left-hand side of the final inequality is the log of the marginal likelihood ratio. We denote this quantity as $\ell(x; x^{(1)}, x^{(2)})$. The right-hand side of the final inequality is the log of the prior odds, which we again rewrite as $\log \frac{1-\theta}{\theta}$. We now restate the classification rule:

Classification rule: If $\ell(x; x^{(1)}, x^{(2)}) > \log \frac{1-\theta}{\theta}$ then we classify x as belonging to class 1. If $\ell(x; x^{(1)}, x^{(2)}) < \log \frac{1-\theta}{\theta}$ then we classify x as belonging to class 2. Otherwise, we make a random decision.

5.3.3 The log of the marginal likelihood ratio

In this section, we compute the explicit form of the marginal likelihood ratio. The first term in the marginal likelihood ratio is $p(x|x^{(1)}, \eta)$:

$$\begin{aligned}
p(x|x^{(1)}, \eta) &= \frac{p(x, x^{(1)}|\eta)}{p(x^{(1)}|\eta)} \\
&= \frac{\int_{\phi_1} p(x, x^{(1)}|\phi_1) \cdot p(\phi_1|\eta)\, d\phi_1}{\int_{\phi_1} p(x^{(1)}|\phi_1) \cdot p(\phi_1|\eta)\, d\phi_1} \\
&= \frac{\binom{L}{x_1 \ldots x_W} \frac{1}{B(\eta)} \int_{\phi_1} \prod_{i=1}^{W} \phi_{1i}^{\eta + x_i^1 + x_i - 1}\, d\phi_1}{\frac{1}{B(\eta)} \int_{\phi_1} \prod_{i=1}^{W} \phi_{1i}^{\eta + x_i^1 - 1}\, d\phi_1} \\
&= \binom{L}{x_1 \ldots x_W} \frac{B(\eta + x^{(1)} + x)}{B(\eta + x^{(1)})}
\end{aligned}$$

In the first line, we use Bayes rule to rewrite $p(x|x^{(1)}, \eta)$. In the second line, we express the numerator and the denominator as integrals over the probability vector ϕ_1. In the third line, we have expanded the multinomial likelihoods $p(x, x^{(1)}|\phi_1)$ and $p(x^{(1)}|\phi_1)$ and the Dirichlet prior $p(\phi_1|\eta)$ which appears in both the numerator and the denominator. The multinomial coefficient $\binom{L}{x_1 \ldots x_W}$ is factored out of the integral. The multinomial coefficients for the documents in $x^{(1)}$ are present in both the numerator and the denominator and thus cancel out. We also factor out the normalizing constant of the Dirichlet prior $\frac{1}{B(\eta)}$. We use $B(\eta)$ to denote the multivariate Beta function which we discuss in more detail below. We then combine the remaining terms into a single product over the vocabulary. In the numerator, the exponent of the multinomial parameter ϕ_{1i} is the hyper-parameter η plus the word counts x_i^1 and x_i. In the denominator, the exponent contains only the hyper-parameter and the word count x_i^1. At this point, we recognize that we are integrating over an unnormalized Dirichlet density in both the numerator and the denominator. If we multiply by the correct

normalizing constant – $B(\eta+x^{(1)}+x)$ in the numerator and $B(\eta+x^{(1)})$ in the denominator – the integrals evaluate to 1. Thus, we are left in the last line with the multinomial coefficient from the document x and a ratio of Dirichlet normalizing constants. For more details of this derivation, see Section 2.2.2.

The function $B(\cdot)$ is the multivariate Beta function which takes a vector as argument. We are overloading our notation for η to represent both a strictly positive real-valued scalar and the W-dimensional vector $[\eta, \ldots, \eta]$ (when used as an argument to the Beta function). The notation $B(\eta + x)$ is shorthand for the Beta function with vector argument $[\eta \ldots \eta] + [x_1 \ldots x_W]$. Similarly, the notation $B(\eta + x^{(1)} + x)$ is shorthand for the Beta function with vector argument $[\eta \ldots \eta] + [x_1^1 \ldots x_W^1] + [x_1 \ldots x_W]$.

We perform the same computation to compute $p(x|x^{(2)}, \eta)$:

$$p(x|x^{(2)}, \eta) = \binom{L}{x_1, \ldots, x_W} \frac{B(\eta + x + x^{(2)})}{B(\eta + x^{(2)})}$$

Combining these two expressions gives us the log of the marginal likelihood ratio:

$$\begin{aligned} &\ell(x; x^{(1)}, x^{(2)}) \\ &= \log p(x|x^{(1)}, \eta) - \log p(x|x^{(2)}, \eta) \\ &= \log B(\eta + x + x^{(1)}) - \log B(\eta + x^{(1)}) - \log B(\eta + x + x^{(2)}) + \log B(\eta + x^{(2)}) \end{aligned} \tag{5.11}$$

Note that the multinomial coefficient for the document x cancels out. Equation 5.11 gives us an expression for the log of the marginal likelihood ratio in terms of the Beta function. The Beta function itself can be expressed in terms of the Gamma function. The Gamma function

is the extension of the factorial to the real and complex numbers. We use the following identity,

$$\log B(\eta + x) = \left(\sum_w \log \Gamma(\eta + x_w)\right) - \log \Gamma\left(\sum_w \eta + x_w\right)$$

to rewrite the log of the marginal likelihood ratio in terms of the Gamma function:

$$\begin{aligned} &\ell(x; x^{(1)}, x^{(2)}) = \\ &\mathcal{C} + \sum_{w=1}^{W} \left(\log \Gamma(\eta + x_w + x_w^1) - \log \Gamma(\eta + x_w^1) - \log \Gamma(\eta + x_w + x_w^2) + \log \Gamma(\eta + x_w^2) \right) \end{aligned} \tag{5.12}$$

where $\mathcal{C}$ is the non-random quantity[16],

$$\mathcal{C} = -\log \Gamma(W\eta + L + L_1) + \log \Gamma(W\eta + L_1) + \log \Gamma(W\eta + L + L_2) - \log \Gamma(W\eta + L_2) \tag{5.13}$$

5.3.4 Interpreting the log of the marginal likelihood ratio

In this section, we pause to provide an interpretation of the log marginal likelihood ratio that elucidates how evidence is accumulated in favor of both hypotheses $y = 1$ and $y = 2$. To the best of our knowledge, this interpretation has not been presented elsewhere.

[16] We are conditioning on the lengths L, L_1, and L_2

We start by re-writing the $\ell(x; x^{(1)}, x^{(2)})$ in a more convenient form using the identity $\log\Gamma(\eta + a) = \log\Gamma(\eta) + \sum_{k=0}^{a-1}\log(\eta + k)$ for a a non-negative integer. This allows us to rewrite each instance of log Gamma as a sum of two terms: a non-random log Gamma term, and a summation of log terms.

$$
\begin{aligned}
&\ell(x; x^{(1)}, x^{(2)}) \\
&= \mathcal{C} + \sum_{w=1}^{W}\Big(\log\Gamma(\eta + x_w + x_w^1) - \log\Gamma(\eta + x_w^1)\Big) - \Big(\log\Gamma(\eta + x_w + x_w^2) - \log\Gamma(\eta + x_w^2)\Big) \\
&= \mathcal{C} + \sum_{w=1}^{W}\Bigg[\sum_{k=0}^{x_w + x_w^1 - 1}\log(\eta + k) - \sum_{k=0}^{x_w^1 - 1} log(\eta + k) - \sum_{k=0}^{x_w + x_w^2 - 1}\log(\eta + k) + \sum_{k=0}^{x_w^2} log(\eta + k)\Bigg] \\
&= \mathcal{C} + \sum_{w=1}^{W}\Bigg[\sum_{k=x_w^1}^{x_w + x_w^1 - 1}\log(\eta + k) - \sum_{k=x_w^2}^{x_w + x_w^2 - 1} log(\eta + k)\Bigg] \\
&= \mathcal{C} + \sum_{w=1}^{W}\Bigg[\sum_{k=0}^{x_w - 1} log(\eta + x_w^1 + k) - \sum_{k=0}^{x_w - 1} log(\eta + x_w^2 + k)\Bigg] \\
&= \mathcal{C} + \sum_{w=1}^{W}\sum_{k=0}^{x_w - 1}\log(\eta + x_w^1 + k) - \log(\eta + x_w^2 + k)
\end{aligned}
\tag{5.14}
$$

The third line is a straightforward application of the identity. In the fourth line, we observe that the first x_w^1 terms of the first summation cancel entirely with the second summation – this allows us to rewrite the limits of the first summation to range from $k = x_w^1$ to $k = x_w + x_w^1 - 1$. Similarly, the first x_w^2 terms of the third summation cancel entirely with the last summation – this allows us to rewrite the limits of the last summation to range from $k = x_w^2$ to $k = x_w + x_w^2 - 1$. In the fifth line, we observe that we can partially move the summation variable k into the log function and we rewrite the summations again to range from $k = 0$ to $k = x_w - 1$. At this point, both summations sum over the same range and we combine them in the final line.

We now apply the same procedure to rewrite the non-random term $\mathcal{C}$,

$$\begin{aligned}
\mathcal{C} &= -log\Gamma(W\eta + L + L_1) - \log\Gamma(W\eta + L_2) + \log\Gamma(W\eta + L_1) + \log\Gamma(W\eta + L + L_2) \\
&= -\sum_{k=0}^{L+L_1-1} \log(W\eta + k) - \sum_{k=0}^{L_2-1} \log(W\eta + k) + \sum_{k=0}^{L_1-1} \log(W\eta + k) + \sum_{k=0}^{L+L_2-1} \log(W\eta + k) \\
&= -\sum_{k=L_1}^{L+L_1-1} \log(W\eta + k) + \sum_{k=L_2}^{L+L_2-1} \log(W\eta + k) \\
&= -\sum_{k=0}^{L-1} \log(W\eta + L_1 + k) + \sum_{k=0}^{L-1} \log(W\eta + L_2 + k) \\
&= \sum_{k=0}^{L-1} \log(W\eta + L_2 + k) - \log(W\eta + L_1 + k)
\end{aligned} \tag{5.15}$$

Combining these expressions allows us to rewrite $\ell(x; x^{(1)}, x^{(2)})$ as follows,

$$\begin{aligned}
&\ell(x; x^{(1)}, x^{(2)}) \\
&= \sum_{w=1}^{W} \sum_{l=0}^{x_w-1} \log(\eta + x_w^1 + l) - \log(\eta + x_w^2 + l) - \sum_{l=0}^{L-1} log(W\eta + L_1 + l) - log(W\eta + L_2 + l) \\
&\equiv f(x, x^{(1)}, x^{(2)}) - g(L, L_1, L_2)
\end{aligned}$$

Why have we gone to such lengths to rewrite $\ell(x; x^{(1)}, x^{(2)})$? First, in this form, we can rewrite our classification rule as follows:

Classification rule: If $f(x, x^{(1)}, x^{(2)}) - g(L, L_1, L_2) > \log\frac{1-\theta}{\theta}$ then we classify x as belonging to class 1. If $f(x, x^{(1)}, x^{(2)}) - g(L, L_1, L_2) < \log\frac{1-\theta}{\theta}$ then we classify x as belonging to class 2. Otherwise, we make a random decision.

The random quantity $f(x, x^{(1)}, x^{(2)})$ must exceed the non-random decision boundary $g(L, L_1, L_2)$ by $\log \frac{1-\theta}{\theta}$ in order for the document x to be assigned to class 1. Otherwise, x is assigned to class 2.

The second benefit of rewriting the $\ell(x; x^{(1)}, x^{(2)})$ in this form is that we have removed all of the Gamma terms and now have an expression in terms of the logarithm function only. We provide an interpretation for this expression.

First, note that $f(x, x^{(1)}, x^{(2)})$ has exactly L terms. A word must occur in both x and in either $x^{(1)}$ or $x^{(2)}$ to contribute toward the summation $f(x, x^{(1)}, x^{(2)})$. Let us examine the contribution of the wth word in the vocabulary to $f(x, x^{(1)}, x^{(2)})$. If the wth word in the vocabulary occurs with frequency x_w in document x, then it contributes a total of

$$\kappa(w) \equiv \sum_{l=0}^{x_w - 1} \log(\eta + x_w^1 + l) - \log(\eta + x_w^2 + l) \tag{5.16}$$

towards $f(x, x^{(1)}, x^{(2)})$. Recall that x_w^1 and x_w^2 denote the number of times the wth word in the vocabulary occurs in $x^{(1)}$ and $x^{(2)}$ respectively. If $x_w^1 > x_w^2$, then each term is positive and w contributes $\kappa(w)$ as evidence, or support, for the hypothesis $y = 1$. If $x_w^1 < x_w^2$, then each term is negative and w contributes $\kappa(w)$ as evidence, or support, for the hypothesis $y = 2$. If $x_w^1 = x_w^2$, then $\kappa(w)$ is zero and w contributes no evidence toward either hypothesis.

Assume for now that $x_w^1 > x_w^2$ and $\kappa(w)$ will be positive. The first occurrence of the wth vocabulary word in x (corresponding to the first term in the summation in Equation 5.16) contributes $\log(\eta + x_w^1) - \log(\eta + x_w^2)$. This quantity has a simple constructive definition: start with the interval $[\eta + x_w^2, \eta + x_w^1]$ on the real line. Note that this interval has length $(x_w^1 - x_w^2)$. Transform the interval by taking its log. The resulting interval is $[\log(\eta + x_w^2),\ \log(\eta + x_w^1)]$ which now has length $\log(\eta + x_w^1) - \log(\eta + x_w^2)$. This length is the contribution of the first

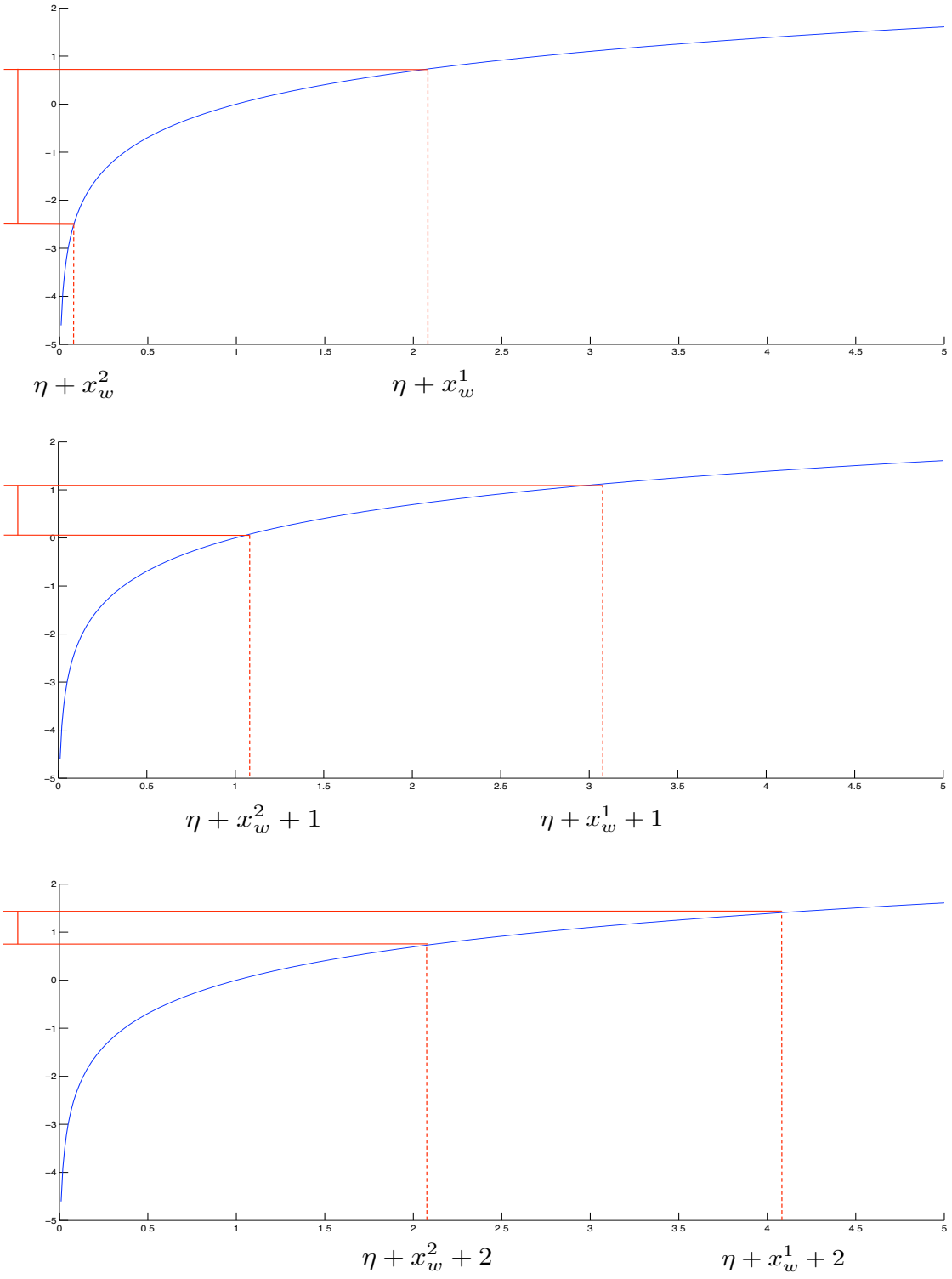

Figure 5.22: The first three steps in the construction. The blue line is the natural logarithm.

occurrence of the word w in x. This process is shown in the top plot of Figure 5.22.

For the second occurrence of the word w in x, the same process is repeated with the only difference that the initial interval is shifted by one: $[\eta + x_w^2 + 1,\ \eta + x_w^1 + 1]$. Note that this interval has the same length $(x_w^1 - x_w^2)$. The transformed interval now has length $\log(\eta + x_w^1 + 1) - \log(\eta + x_w^2 + 1)$, the contribution of the second occurrence of the word w. This process is shown in the middle plot of Figure 5.22.

Similarly, for the third occurrence of the word w in x, the same process is repeated except, again, the initial interval is shifted by one: $[\eta + x_w^2 + 2,\ \eta + x_w^1 + 2]$. Again, the interval has length $(x_w^1 - x_w^2)$. The length of the transformed interval is now $\log(\eta + x_w^1 + 2) - \log(\eta + x_w^2 + 2)$, the contribution of the third occurrence of the word w. This process is shown in the bottom plot of Figure 5.22.

This same process is repeated for every occurrence of the wth vocabulary word in the document x, i.e. this process is repeated x_w times. We summarize this process and make some observations:

1. Only the words that occur in the document x and either $x^{(1)}$ or $x^{(2)}$ contribute evidence, or support, toward either hypothesis $y = 1$ or $y = 2$.

2. For a word w, the above process is repeated x_w times.

3. Each time, the initial interval is shifted by one. This initial interval always has the same length $|x_w^1 - x_w^2|$. Thus, the larger the discrepancy between x_w^1 and x_w^2, the larger this interval and the greater the evidence will be towards either hypothesis $y = 1$ or $y = 2$.

4. The initial interval is then transformed by the logarithm function. If $x_w^1 > x_w^2$, then the ith occurrence of the word w contributes $|\log(\eta + x_w^1 + i) - \log(\eta + x_w^2 + i)|$ towards the hypothesis $y = 1$. If $x_w^1 < x_w^2$, then the ith occurrence of the word w contributes

$-|\log(\eta + x_w^1 + i) - \log(\eta + x_w^2 + i)|$ towards the hypothesis $y = 2$ (note the negative sign). If $x_w^1 = x_w^2$, the word w contributes no evidence.

5. The slope of the natural logarithm function has a great impact on the amount of evidence contributed by a word w.

   - The first occurrence of w contributes the most evidence. Since the slope of the natural logarithm function approaches zero as the argument approaches infinity, each successive occurrence of w contributes less and less evidence. Thus, a document x that shares 100 words with $x^{(1)}$, each of which occurs only 1 time, is preferred to a document that shares 1 word with $x^{(1)}$, which occurs 100 times.
   - The largest gain occurs when either x_w^1 or x_w^2 is 0. Assume $x_w^2 = 0$. In this case, the first occurrence of w contributes $\log(\eta + x_w^1) - \log(\eta + 0)$. Note that if $\eta < 1$, then $\log(\eta) < 0$. In fact, as η approaches zero, $\log(\eta)$ approaches negative infinity. In other words, as η approaches zero, the evidence contributed by the first occurrence of w approaches positive infinity. This makes intuitive sense since η is a measure of the sparsity of the Dirichlet prior. For $\eta \ll 1$, sharing even a single word between x and $x^{(1)}$ (or $x^{(2)}$) is significant evidence.

This interpretation has allowed us to uncover a number of implications that arise from our classification rule. First, each successive occurrence of w contributes less and less evidence which gives preference to a document with many shared words of low frequency over a document with few shared words of high frequency. Second, this formulation makes it clear that the natural logarithm function is at the heart of the marginal likelihood ratio for the multinomial Dirichlet mixture model. Modifying or exchanging the logarithm function would produce a classification rule with different assumptions. For example, one might replace the logarithm function with a linear function (so that every occurrence of the word w contributes an equal amount of evidence) or with a function whose slope increases (so that

every additional occurrence of the word w contributes an increasing amount). Or one might parameterize the base of the logarithm by a function of the word w. For example, instead of treating each word in the vocabulary identically, one could give more weight to entity words by changing the base of the logarithm.

Note that this same constructive process can be applied to $g(L, L_1, L_2)$.

$$g(L, L_1, L_2) = \sum_{l=0}^{L-1} \log(W\eta + L_1 + l) - \log(W\eta + L_2 + l)$$

One interpretation of $g(L, L_1, L_2)$ is that it acts as a non-random decision boundary. If the random quantity $f(x, x^{(1)}, x^{(2)})$ exceeds $g(L, L_1, L_2)$ by more than $\log \frac{1-\theta}{\theta}$, then document x is assigned to class 1. Otherwise, it is assigned to class 2.

Another interpretation is that $g(L, L_1, L_2)$ is analogous to a prior where the class whose set of representative documents contains the most words is a priori the likeliest class. If $L_1 = L_2$, then $g(L, L_1, L_2) = 0$. If $L_1 > L_2$ then $g(L, L_1, L_2)$ is positive. If $L_1 < L_2$ then $g(L, L_1, L_2)$ is negative. Thus, $g(L, L_1, L_2)$ is a prior belief of the class of x derived only from the model parameters (L, L_1, and L_2) without any knowledge of x. The greater the disparity between L_1 and L_2, the stronger this prior belief.

5.3.5 The Bayesian classifier error rate

Up to this point, we have taken a detailed look at the log of the marginal likelihood ratio. We have examined its form and attempted to understand how the words in the document x are used to weigh the evidence in favor of both hypotheses $y = 1$ and $y = 2$. We now derive an expression for the error rate of our classification rule. We repeat the classification rule

here for convenience:

Classification rule: If $\ell(x; x^{(1)}, x^{(2)}) > \log \frac{1-\theta}{\theta}$ then we classify x as belonging to class 1. If $\ell(x; x^{(1)}, x^{(2)}) < \log \frac{1-\theta}{\theta}$ then we classify x as belonging to class 2. Otherwise, we make a random decision.

At this point, it is important that we make a clear distinction between (1) the classifier and (2) the analysis of the error rate of the classifier.

The practitioner who wants to classify a document x but has available only a set of documents from class 1 and a set of documents from class 2 has a number of options available to them. We discussed a few of these options earlier – namely, using a MAP estimate of the parameters ϕ_k or integrating over the parameters ϕ_k. Both of these methods lead to a **classification function** – a real-valued function of the document x, the document sets $x^{(1)}$ and $x^{(2)}$, and the hyper-parameter η. It will be helpful to think of this classification function as a black box that takes in these quantities and outputs a real-valued scalar; how it computes this scalar value is, for now, irrelevant. In our case, the classification function is denoted $\ell(x; x^{(1)}, x^{(2)})$.

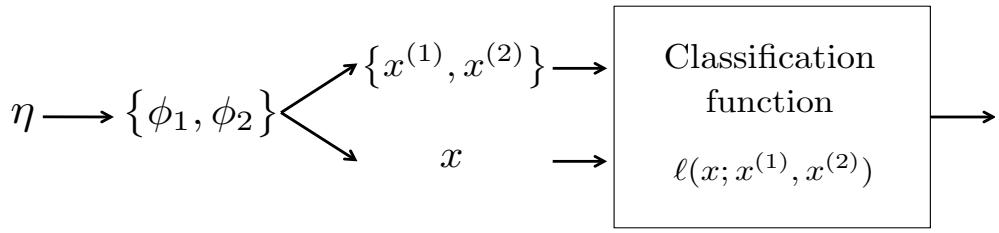

Figure 5.23: To estimate the true error, we sample ϕ_1, ϕ_2, $x^{(1)}$, and $x^{(2)}$. Fixing these quantities, we sample $S \gg 1$ documents x from class 1 and $S \gg 1$ documents x from class 2. We classify each document x using the classification function and compute the proportion of documents that were misclassified

Once we have this real-valued output we compare it to the decision boundary $\log \frac{1-\theta}{\theta}$ to

produce a classification decision. The probability that the real-valued output of the classification function leads to a mis-classification of the document x is the probability of error. If we average this probability of error over all documents x of length L, we have the expected probability of error (also called the true error in the statistical pattern recognition literature [76]). We term the expected probability of error for $\ell(x; x^{(1)}, x^{(2)})$ the *Bayesian classifier error rate*.

To compute the Bayesian classifier error rate, we sample two multinomial distributions, ϕ_1 and ϕ_2, from a symmetric Dirichlet prior with hyper-parameter η. Given ϕ_1 and ϕ_2, we sample two sets of representative documents $x^{(1)}$ and $x^{(2)}$. Once these quantities are fixed, we sample $S \gg 1$ documents of length L from ϕ_1 and $S \gg 1$ documents of length L from ϕ_2 and classify each document using our classification rule. We then estimate the Bayesian classifier error rate by computing the proportion of documents that were misclassified. Figure 5.23 illustrates this process.

We can mathematically express this process as follows,

$$\begin{aligned} p_\epsilon = \theta \int_{-\infty}^{\log \frac{1-\theta}{\theta}} p\big(\ell(x; x^{(1)}, x^{(2)}) \big| \phi_{1:2}, x^{(1)}, x^{(2)}, y = 1, \eta\big)\, d\ell(x; x^{(1)}, x^{(2)}) \;\; + \\ (1-\theta) \int_{\log \frac{1-\theta}{\theta}}^{+\infty} p\big(\ell(x; x^{(1)}, x^{(2)}) \big| \phi_{1:2}, x^{(1)}, x^{(2)}, y = 2, \eta\big)\, d\ell(x; x^{(1)}, x^{(2)}) \end{aligned} \tag{5.17}$$

The classification function $\ell(x; x^{(1)}, x^{(2)})$ is itself a random variable since it is a function of a random input x. Its conditional density is given by,

$$p\big(\ell(x; x^{(1)}, x^{(2)}) \big| \phi_{1:2}, x^{(1)}, x^{(2)}, y = k, \eta\big) \qquad \text{for } k \in \{1, 2\}$$

The first integral in Equation 5.17 is over the interval from negative infinity to the decision boundary. This is the interval in which we classify a document as belonging to class 2. Thus, the error in this interval is obtained by integrating over the conditional density of the classification function for $y = 1$. Similarly, the second integral is over the interval from the decision boundary to positive infinity. This is the interval in which we classify a document as belonging to class 1. Thus, the error in this interval is obtained by integrating over the conditional density of the classification function for $y = 2$. Both integrals are weighted by the prior probability of class 1 and class 2 respectively. We use p_ϵ to denote the Bayesian classifier error rate.

5.3.6 Monte Carlo estimates

We first examine Monte Carlo estimates of the conditional density of the classification function $\ell(x; x^{(1)}, x^{(2)})$. We experiment with $\eta \in \{0.1, 1.0, 10\}$, $L \in \{15, 250, 600, 1200\}$, and $N, M \in \{1, 10, 50, 100\}$. Recall that N and M are the number of representative documents in the sets $x^{(1)}$ and $x^{(2)}$ respectively. In all of our simulations, we set $N = M$. We used a vocabulary of size $W = 10,000$ words and a prior probability for class 1 of $\theta = 0.5$.

We follow the process described in the previous section (illustrated in Figure 5.23). For each configuration of the parameters η, L, and N we sample two multinomial distributions, ϕ_1 and ϕ_2, from a symmetric Dirichlet prior with hyper-parameter η. Given ϕ_1 and ϕ_2, we sample two sets of representative documents $x^{(1)}$ and $x^{(2)}$ both of size N. Once these quantities are fixed, we sample $S \gg 1$ documents of length L from ϕ_1 and $S \gg 1$ documents of length L from ϕ_2 and compute $\ell(x; x^{(1)}, x^{(2)})$ for each sampled document. The blue histograms correspond to the documents sampled from class 1. The red histograms correspond to the documents sampled from class 2. We used $S = 10,000$ Monte Carlo samples (for each class).

Figure 5.24 shows the results of the Monte Carlo simulations for $\eta = 10$. Each row cor-

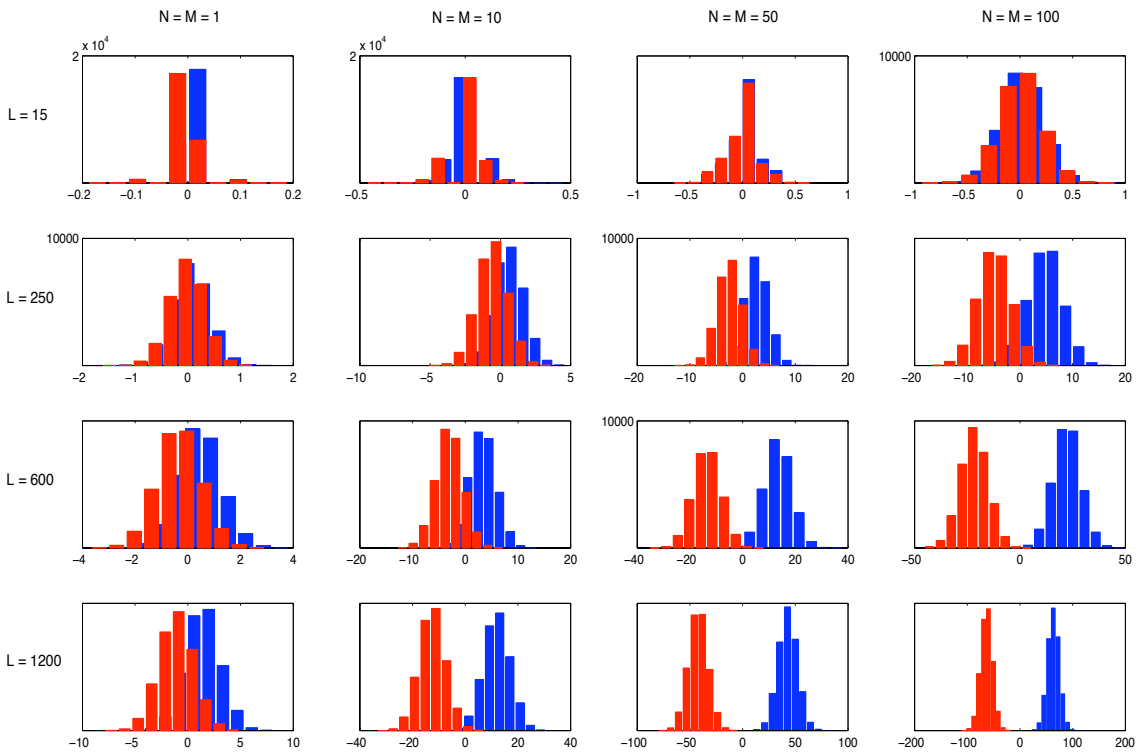

Figure 5.24: Monte Carlo estimates of the log marginal likelihood ratio $\ell(x; x^{(1)}, x^{(2)})$ for $\eta = 10$.

responds to an increasing document length. Each column corresponds to an increasing Dirichlet hyper-parameter. Thus, for $L = 15$ and $N = 1$ (the upper leftmost plot) we observe one document of length 15 from class 1 and one document of length 15 from class 2. For $L = 1200$ and $N = 100$ (the bottom rightmost plot) we observe 100 documents of length 1200 from class 1 and 100 documents of length 1200 from class 2.

The Bayesian classifier error rate p_ϵ can be estimated by the overlap of the conditional densities – that is, the proportion of the blue histograms below the decision boundary (which is at 0) weighted by the prior probability of class 1 (0.5) plus the proportion of the red histograms above the decision boundary weighted by the prior probability of class 2 (also 0.5). Figure 5.25 shows the results for $\eta = 1$ and Figure 5.26 shows the results for $\eta = 0.1$.

Not surprisingly, we see a higher degree of overlap when compared to the Monte Carlo

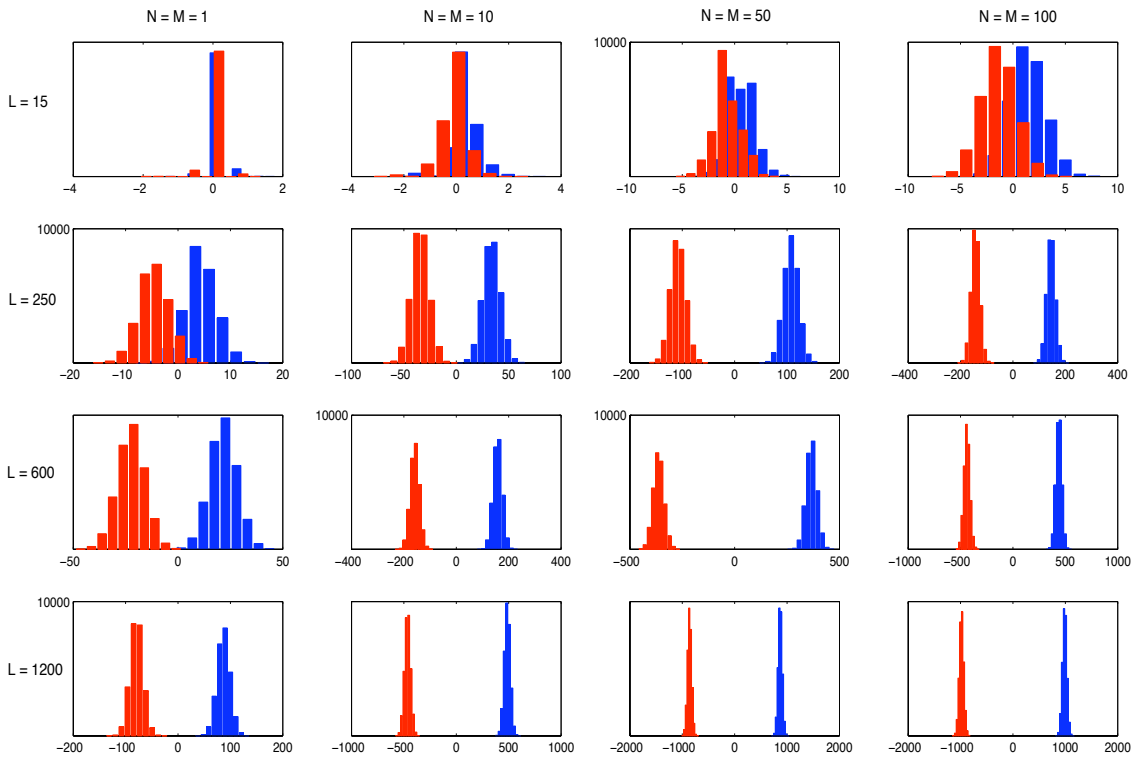

Figure 5.25: Monte Carlo estimates of the log marginal likelihood ratio $\ell(x; x^{(1)}, x^{(2)})$ for $\eta = 1$.

simulations in Section 5.2 (in which we observed the true class parameters). In Figure 5.24, there is a significant amount of overlap for all configurations of L and N except for $L = 600$ and $N \geq 50$ and $L = 1200$ and $N \geq 10$. In fact, the Bayesian classifier error rate for $L = 15$ is roughly 50% for all values of N.

Previously for $\eta = 10$ and $L = 250$ words the Bayes error was negligible (approximately 2×10^{-4}). However in this case when $\eta = 10$ and $L = 250$, even after observing 100 representative documents from each class, there is still a significant amount of overlap between the histograms (see Figure 5.25).

The histograms of $\ell(x; x^{(1)}, x^{(2)})$ appear to be Normal except for the edge case $L = 15$ and $N = 1$. In the next section, we present a Normal approximation to the conditional density of $\ell(x; x^{(1)}, x^{(2)})$ and derive the mean and variance of the approximation.

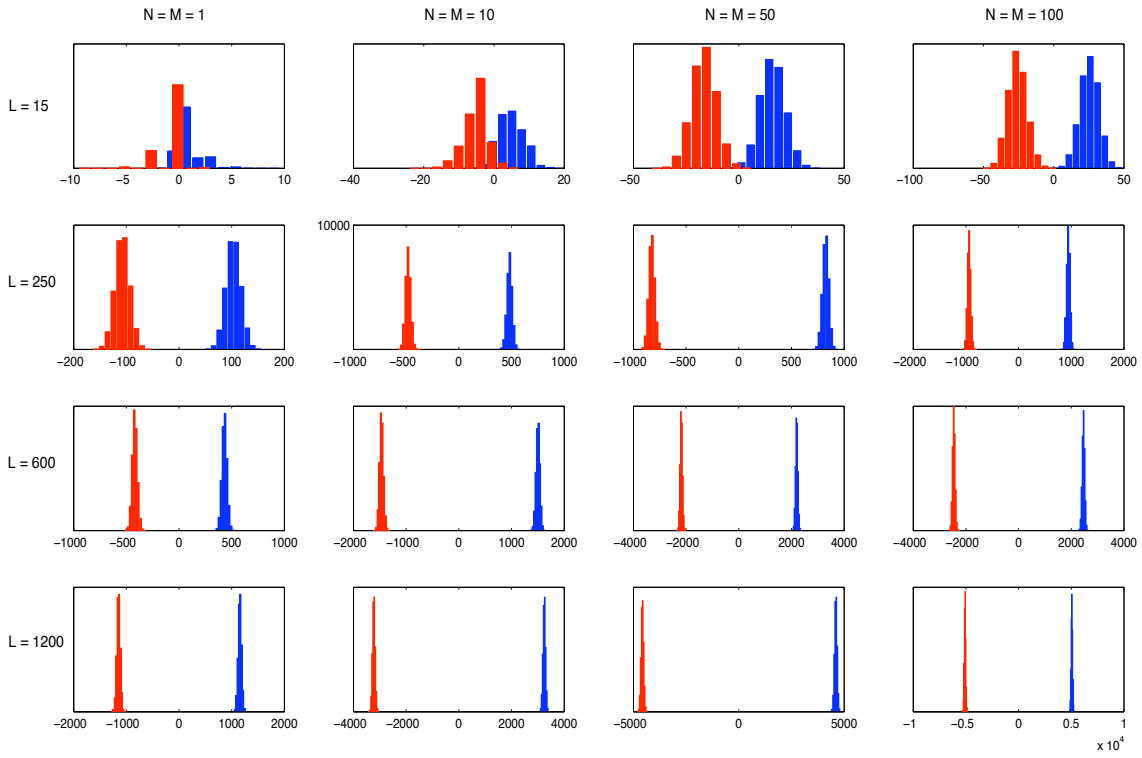

Figure 5.26: Monte Carlo estimates of the log marginal likelihood ratio $\ell(x; x^{(1)}, x^{(2)})$ for $\eta = 0.1$.

5.3.7 Normal approximation and moments

We approximate the conditional density of $\ell(x; x^{(1)}, x^{(2)})$ with a Normal distribution by appealing to the same central limit theorem for multinomial sums [46]. It will be easiest to work with the logarithm-only form of $\ell(x; x^{(1)}, x^{(2)})$ presented earlier:

$$\begin{aligned}\ell(x; x^{(1)}, x^{(2)}) &= \mathcal{C} + \sum_{w=1}^{W} \sum_{l=0}^{x_w - 1} \log(\eta + x_w^1 + l) - \log(\eta + x_w^2 + l) \\ &= \mathcal{C} + \sum_{w=1}^{W} \sum_{l=0}^{L-1} \mathrm{I}(l \leq x_w - 1) \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l}\right)\end{aligned}$$

Recall that $\mathcal{C}$ is not random. We change the upper limit of the summation to $l = L - 1$ and

introduce an indicator function $\mathrm{I}(l \leq x_w - 1)$ to avoid having a summation with a random number of terms.

The expected value and the variance of the Normal approximation of the conditional density of $\ell(x; x^{(1)}, x^{(2)})$ for $y = k$ are computed in terms of independent Poisson random variables, $z_w \sim \text{Poisson}(L\phi_{kw})$ (for $w \in \{1, \ldots, W\}$) where $\sum_w z_w = L$. Note that the mean of the Poisson random variables is a function of ϕ_k where k is determined by the value of y.

The expected value of the classification function $\ell(x; x^{(1)}, x^{(2)})$ conditioned on the hyper-parameter η, the class parameters $\phi_{1:2}$, the document sets $x^{(1)}$ and $x^{(2)}$, and the class indicator $y = k$ is given by,

$$\begin{aligned}
\mu_k - \mathcal{C} &= E[\ell(x; x^{(1)}, x^{(2)}) | \eta, \phi_{1:2}, x^{(1)}, x^{(2)}, y = k] - \mathcal{C} \\
&= \sum_{w=1}^{W} \sum_{k=0}^{L-1} E\left[\mathrm{I}(z_w \geq l + 1) \,\middle|\, y = k, \phi_{1:2}, x^{(1)}, x^{(2)}, \eta \right] \log \frac{(\eta + x_w^1 + l)}{(\eta + x_w^2 + l)} \\
&= \sum_{w=1}^{W} \sum_{l=0}^{L-1} p(z_w \geq l + 1 | \phi_{kw}) \log \frac{(\eta + x_w^1 + l)}{(\eta + x_w^2 + l)}
\end{aligned} \tag{5.18}$$

and the variance is given by

$$\begin{aligned}
\sigma_k^2 &= \sum_{w=1}^{W} \mathrm{Var}\Big(f_w(z_w) \Big) - L \left[\frac{1}{L} \sum_{w=1}^{W} \mathrm{Cov}(f_w(z_w), z_w) \right]^2 \\
&= \sum_{w=1}^{W} \Big(E[f_w(z_w)^2] - E[f_w(z_w)]^2 \Big) - \frac{1}{L} \Big(\sum_{w=1}^{W} E[z_w f_w(z_w)] - E[f_w(z_w)] E[z_w] \Big)^2
\end{aligned}$$

where

$$f_w(z_w) = \sum_{l=0}^{L-1} \mathrm{I}(z_w \geq l+1) \log \frac{\eta + x_w^1 + l}{\eta + x_w^2 + l}$$

The equation for σ_k^2 is taken directly from the central limit theorem in Section 5.2.6. When computing the variance, we still condition on η, $\phi_{1:2}$, $x^{(1)}$, $x^{(2)}$, and $y = k$ even though this is not shown for the sake of simplicity.

Computing the variance requires four quantities: the expected value of the square $E[f_w(z_w)^2]$, the square of the expected value $E[f_w(z_w)]^2$, the expected value of the product $E[z_w f_w(z_w)]$, and the product of the expected values $E[f_w(z_w)]E[z_w]$.

The square of the expected value $E[f_w(z_w)]^2$ can be computed from Equation 5.18. The product of the expected values can be computed from Equation 5.18 and the equation $E[z_w] = L\phi_{kw}$ (since z_w is a Poisson random variable with parameter $L\phi_{kw}$). The expected value of the square $E[f_w(z_w)^2]$ and the expected value of the product $E[z_w f_w(z_w)]$ are derived below.

$$
\begin{aligned}
E[f_w(z_w)^2] &= \sum_{i=0}^{\infty} p(z_w = i|\phi_{kw}) \left(\sum_{l=0}^{L-1} \mathrm{I}(i \geq l+1) \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l} \right) \right)^2 \\
&= \sum_{i=1}^{\infty} p(z_w = i|\phi_{kw}) \left(\sum_{l=0}^{L-1} \mathrm{I}(i \geq l+1) \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l} \right) \right)^2 \\
&= \sum_{i=1}^{L} p(z_w = i|\phi_{kw}) \left(\sum_{l=0}^{L-1} \mathrm{I}(i \geq l+1) \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l} \right) \right)^2 \\
&+ \sum_{i=L+1}^{\infty} p(z_w = i|\phi_{kw}) \left(\sum_{l=0}^{L-1} \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l} \right) \right)^2 \\
&= \sum_{i=1}^{L} \left[p(z_w = i|\phi_{kw}) \left(\sum_{l=0}^{L-1} \mathrm{I}(i \geq l+1) \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l} \right) \right)^2 \right] \\
&+ p(z_w \geq L+1|\phi_{kw}) \left(\sum_{l=0}^{L-1} \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l} \right) \right)^2
\end{aligned}
$$

and

$$
\begin{aligned}
E[z_w f_w(z_w)] &= \sum_{l=0}^{L-1} \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l} \right) E[z_w \ \mathrm{I}(z_w \geq l+1)] \\
&= \sum_{l=0}^{L-1} \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l} \right) \sum_{j=0}^{\infty} j \ p(z_w = j|\phi_{kw}) \ \mathrm{I}(j \geq l+1) \\
&= \sum_{l=0}^{L-1} \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l} \right) \sum_{j=l+1}^{\infty} j \ p(z_w = j|\phi_{kw}) \\
&= \sum_{l=0}^{L-1} \log\left(\frac{\eta + x_w^1 + l}{\eta + x_w^2 + l} \right) \left(L\phi_{ki} - \sum_{j=0}^{l} l \ p(z_w = j|\phi_{kw}) \right)
\end{aligned}
$$

Computing the mean μ_k and the variance σ_k^2 requires that we compute the Poisson probability mass function (pmf) $p(z_w = j|\phi_{kw})$ and the Poisson cumulative distribution function (cdf) $p(z_w \leq j|\phi_{kw})$ for $w \in \{1, \ldots, W\}$, $j \in \{1, \ldots, L\}$ and $k \in \{1, 2\}$. Thus computing the

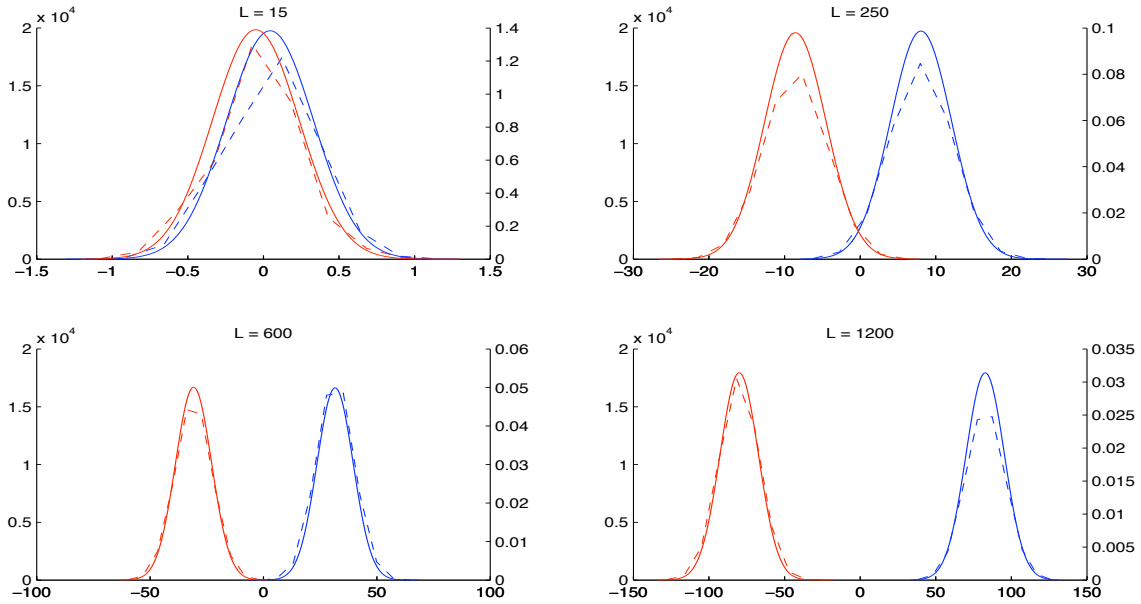

Figure 5.27: Monte Carlo estimates of $\ell(x; x^{(1)}, x^{(2)})$ along with the corresponding Normal approximation

moments of the Normal approximation requires $O(KWL)$ computations where each computation itself (i.e. computing a single Poisson statistic $p(z_w = j|\phi_{kw})$ or $p(z_w \leq j|\phi_{kw})$) requires computing either the factorial of j (for the normalizing constant of the Poisson pdf) or the incomplete Gamma function with argument j (for the Poisson cdf).

We can now approximate the conditional density of the classification rule $\ell(x; x^{(1)}, x^{(2)})$ for $y = k$ with a Normal(μ_k, σ_k^2) distribution. Given this Normal approximation, we estimate the Bayesian classifier error rate using the expression,

$$\begin{aligned} p_\epsilon = \theta \int_{-\infty}^{\log \frac{1-\theta}{\theta}} p\big(\ell(x; x^{(1)}, x^{(2)}) \big| \phi_{1:2}, x^{(1)}, x^{(2)}, y = 1\big)\, d\ell(x; x^{(1)}, x^{(2)}) \;\; + \\ (1-\theta) \int_{\log \frac{1-\theta}{\theta}}^{+\infty} p\big(\ell(x; x^{(1)}, x^{(2)}) \big| \phi_{1:2}, x^{(1)}, x^{(2)}, y = 2\big)\, d\ell(x; x^{(1)}, x^{(2)}) \qquad (5.19) \\ \approx \theta \Phi\left(\frac{\log \frac{1-\theta}{\theta} - \mu_1}{\sigma_1}\right) + (1-\theta)\Phi\left(\frac{\mu_2 - \log \frac{1-\theta}{\theta}}{\sigma_2}\right) \end{aligned}$$

Figure 5.27 shows the Monte Carlo simulations for $\eta = 10$, $N = 100$, and $L \in \{15, 250, 600, 1200\}$ plotted with a dashed line. This corresponds to the last column of Figure 5.24. We have plotted with a solid line the Normal approximation for the conditional density of $\ell(x; x^{(1)}, x^{(2)})$. Using Equation 5.19, we estimate the Bayesian classifier error rate to be $p_\epsilon = 4.3 \times 10^{-1}$, $p_\epsilon = 2.0 \times 10^{-2}$, $p_\epsilon = 4.7 \times 10^{-5}$, and $p_\epsilon = 9.6 \times 10^{-11}$ for $L = 15$, $L = 250$, $L = 600$, and $L = 1200$ respectively.

To assess how accurate of an approximation is given by Equation 5.19, we again consider the case where $\eta = 10$, $N = 100$, and $L \in \{15, 250, 600, 1200\}$. For each value of L, we sampled 10 pairs of multinomial distributions. For each pair, we sampled a set of representative documents $x^{(1)}$ and $x^{(2)}$. Given these 10 pairs (of multinomial parameters and representative documents), we sampled $S = 50,000$ documents from class 1 and $S = 50,000$ documents from class 2. We classified each document using our classification rule and estimated the Bayesian classifier error rate by computing the proportion of documents that were misclassified. We then compute the average and standard deviation across the 10 pairs. These Monte Carlo estimates of the Bayesian classifier error rate are shown in the first column of Table 5.6. For each of the 10 pairs, we also used Equation 5.19 to compute the Normal approximation to the Bayesian classifier error rate. We then computed the average and the standard deviation of the approximate Bayesian classifier error rate across the 10 pairs. This is reported in the second column of Table 5.6. The absolute difference between the means is shown in the third column. The absolute difference between the means as a percentage of the Monte Carlo estimate of the Bayesian classifier error rate is shown in the fourth column. We see that Equation 5.19 provides a good approximation to the Bayesian classifier error rate.

L	Monte Carlo	Normal approx.	Abs. diff.	%
15	$4.31 \times 10^{-1} \pm 4.27 \times 10^{-3}$	$4.41 \times 10^{-1} \pm 4.55 \times 10^{-3}$	9.83×10^{-3}	2.3%
250	$2.14 \times 10^{-2} \pm 2.80 \times 10^{-3}$	$2.12 \times 10^{-2} \pm 2.75 \times 10^{-3}$	1.54×10^{-4}	0.7%
600	$3.30 \times 10^{-5} \pm 1.25 \times 10^{-5}$	$3.59 \times 10^{-5} \pm 6.56 \times 10^{-6}$	2.97×10^{-6}	9.0%
1200	0 ± 0	$5.95 \times 10^{-11} \pm 2.76 \times 10^{-11}$	5.95×10^{-11}	–

Table 5.6: The absolute difference of the means between the true error computed via Monte Carlo simulations and the Normal approximation to the true error (p_ϵ) computed using Equation 5.19 for document lengths $L = 15, 250, 600, 1200$ words.

5.3.8 Analysis of the Bayesian classifier error rate

In this final section, we empirically investigate the relationship between the Bayesian classifier (BC) error rate – as given by the approximation in Equation 5.19 – and certain model parameters of interest. In particular, we look at the relationship between the error rate and the Jeffrey's divergence $D_J(\phi_1||\phi_2)$, the Dirichlet hyper-parameter η, the document length L, and the vocabulary size W.

The pseudocode used to generate the plots in this section are shown in Appendix F. For all plots, we use a vocabulary size of $W = 5,000$ words, $S = 1,000$ Monte Carlo samples, a smoothing constant of $\tau = 1 \times 10^{-8}$, and class prior probability $\theta = 0.5$ (again refer to the pseudocode). Using $\theta = 0.5$ means that in our simulations both classes are a priori equally likely. We set N equal to M for all simulations[17] and we experiment with $N, M \in \{1, 10, 50\}$.

The Jeffrey's Divergence and Dirichlet hyper-parameter η

The first row in Figure 5.28 shows the log of the BC error rate as a function of the Jeffrey's divergence. The second row in Figure 5.28 shows the BC error rate as a function of η. Each column corresponds to an increasing document length $L \in \{15, 250, 600, 1200\}$. The blue, red, and green lines correspond to $N = 1$, $N = 10$, and $N = 50$ respectively. The pseudocode

[17]Recall that N and M are the number of representative documents in $x^{(1)}$ and $x^{(2)}$ respectively.

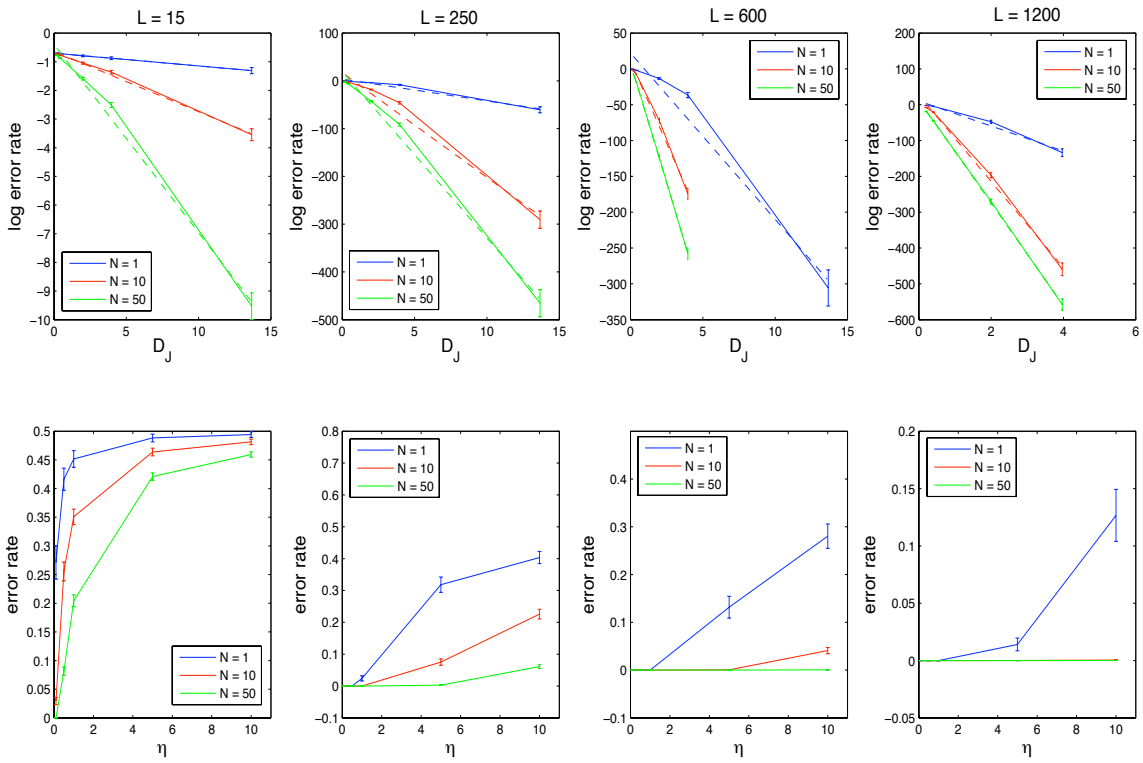

Figure 5.28: The first row shows the log of the Bayesian classifier error rate as a function of the Jeffrey's divergence. The second row shows the Bayesian classifier error rate as a function of the Dirichlet hyper-parameter η. Each column corresponds to an increasing document length $L \in \{15, 250, 600, 1200\}$. The blue, red, and green lines correspond to $N = 1$, $N = 10$ and $N = 50$ respectively.

used to generate these plots is shown in Algorithm 5.

From the first row in Figure 5.28, we again see an (approximately) exponential relationship between the BC error rate and the Jeffrey's divergence. We have plotted the least squares regression lines using a dashed line.

In Figure 5.29 we plot the rates of decay as the document length increases for $N = 1$ (in blue), $N = 10$ (in red) and $N = 50$ (in green). We see that for a fixed value of N, as the document length increases the rate of decay increases, e.g. for $N = 1$ (the blue line) the rate of decay increases from -0.04 for $L = 15$ words to -4.7 for $L = 250$ words. Similarly, we see that for a fixed value of L, as the number of representative documents increases the rate of

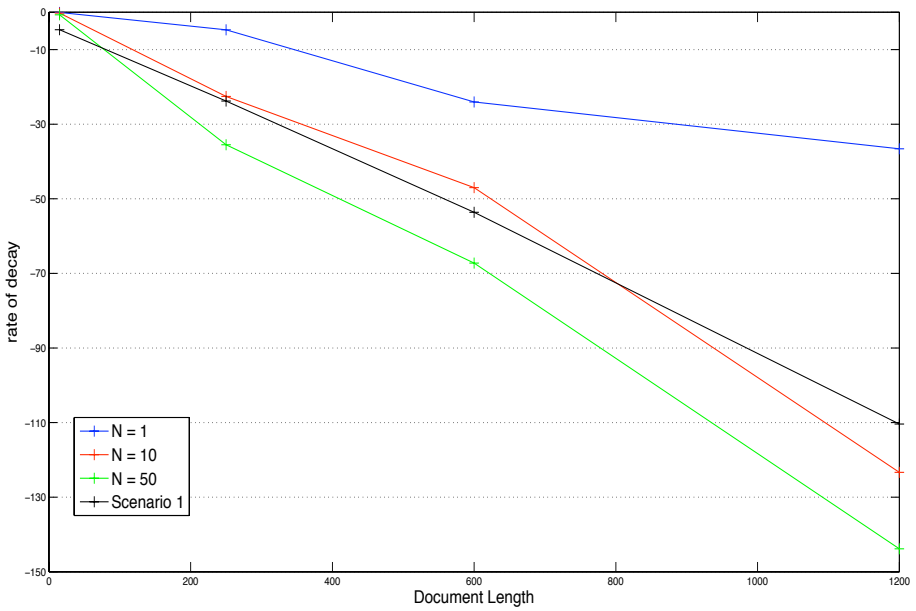

Figure 5.29: The rate of decay (of the exponential function relating the Bayesian classifier error rate and the Jeffrey's divergence) plotted as a function of the document length L for $N = 1$ (blue), $N = 10$ (red) and $N = 50$ (green). The black line shows the rate of decay observed in Section 5.2 between the Bayes error rate and the Jeffrey's divergence.

decay increases. For example, for $L = 600$ words, the rate of decay increases from -24 for $N = 1$ (the blue line) to -47 for $N = 10$ (the red line) to -67 for $N = 50$ (the green line).

We also show using a black line the rate of decay as a function of the document length for the exponential function relating the Bayes error and the Jeffrey's divergence from Section 5.2. Interestingly, the relationship between the rate of decay and the document length from Section 5.2 is similar to the rate of decay observed for $N = 10$ representative documents. Intuitively, for both cases (represented by the black and red line) each additional word added to the document has the same amount of "discriminative information." However, when $N = 1$ (the blue line) an equal increase in the document length does not result in the same decrease in the rate of decay. In this case, each additional word contains less "discriminative information" as compared to the cases represented by the black and red line. Analogously, when $N = 50$ (the green line) an equal increase in the document length results in an even greater decrease in the rate of decay. In this case, each additional word contains

more "discriminative information" as compared to the cases represented by the blue, red, and black line.

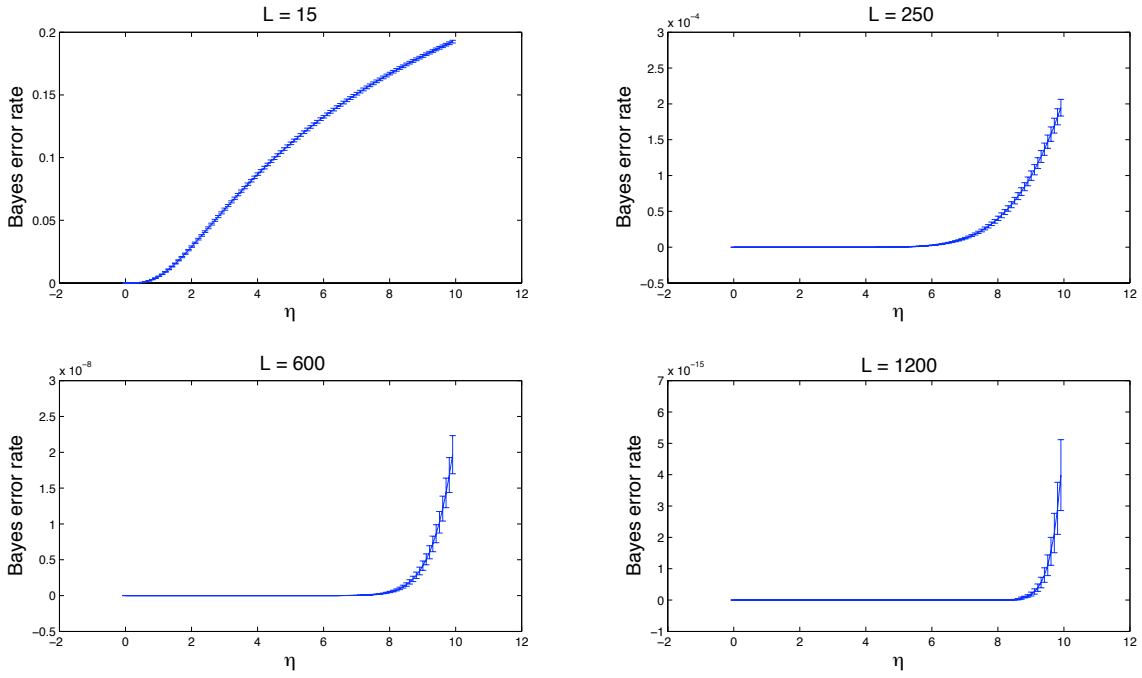

Figure 5.30: **(Formerly Figure 5.9)** The Bayes error rate p_ϵ plotted against the Dirichlet hyper-parameter η for $L = 15, 250, 600, 1200$

We also observe a similar relationship between the BC error rate and η as the relationship observed between the Bayes error and η in Section 5.2. We repeat here Figure 5.9 for the reader's convenience. Figure 5.9 showed the Bayes error as a function of η. Note the similarities with the last row of Figure 5.28 (particularly for $L = 15$).

We see in Figure 5.28 that when η is small, the BC error rate is small and as η increases the BC error rate increases. We established in the previous scenario that the relationship between the Bayes error and η was exponential where the rate of decay was not constant but followed a power-law distribution. Based on the similarities between the second row in Figure 5.28 and Figure 5.9 we conjecture a similar relationship between the BC error rate and η.

Document length L

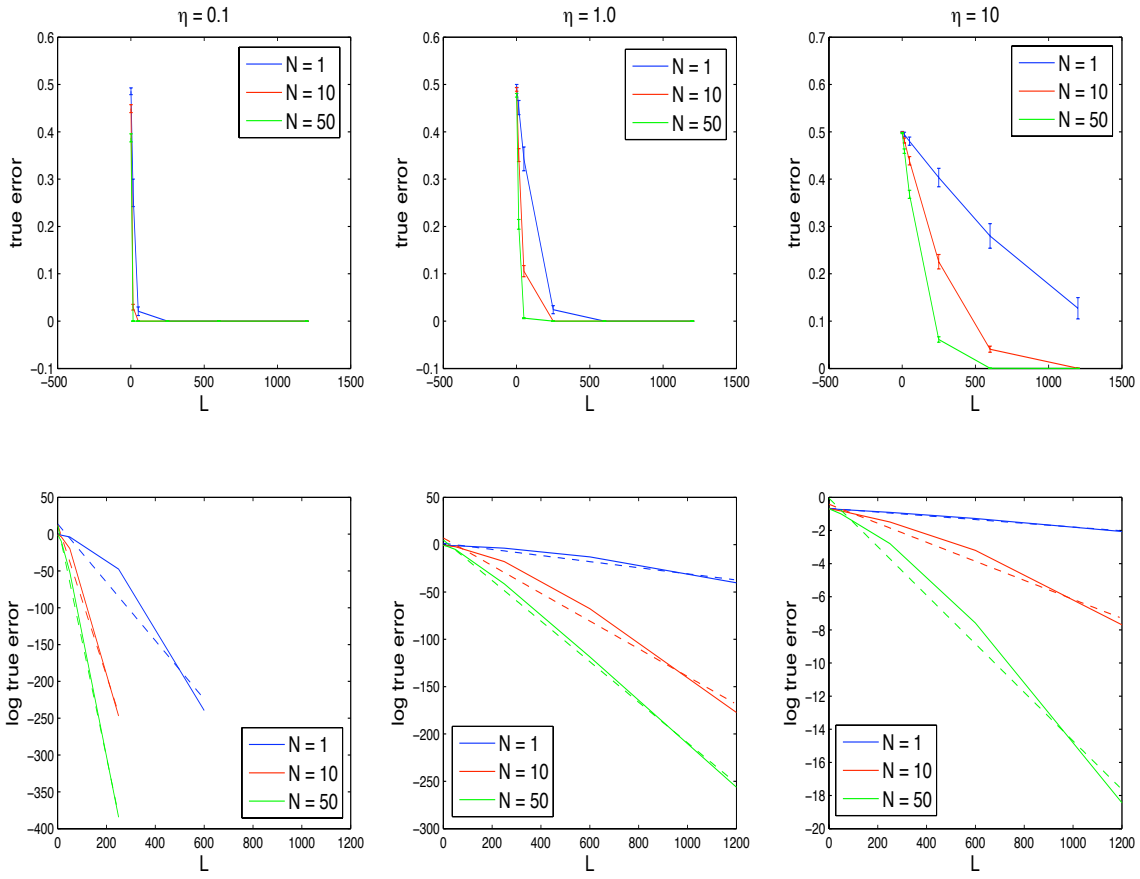

Figure 5.31: The first row shows the log of the Bayesian classifier as a function of the document length. The second row shows the log of the Bayesian classifier error rate as a function of the document length. Each column corresponds to an increasing hyper-parameter $\eta \in \{0.1, 1.0, 10\}$. The blue, red, and green lines correspond to $N = 1$, $N = 10$, and $N = 50$ respectively.

Figure 5.31 shows the BC error rate as a function of the document length. We experiment with $L \in \{1, 15, 250, 600, 1200\}$, $\eta \in \{0.1, 1.0, 10\}$ and $N \in \{1, 10, 50\}$. The red, blue, and green lines again correspond to the number of representative documents. The pseudocode used to generate these plots is shown in Algorithm 6. The first row shows the BC error rate plotted against the document length. The second row shows the log of the BC error rate plotted against the document length. Each column corresponds to an increasing value for η. The least squares regression lines are plotted with a dashed line. We see again an

approximately exponential relationship between the BC error rate and the document length. Not surprisingly, as η increases, the rate of decay decreases.

Vocabulary size W

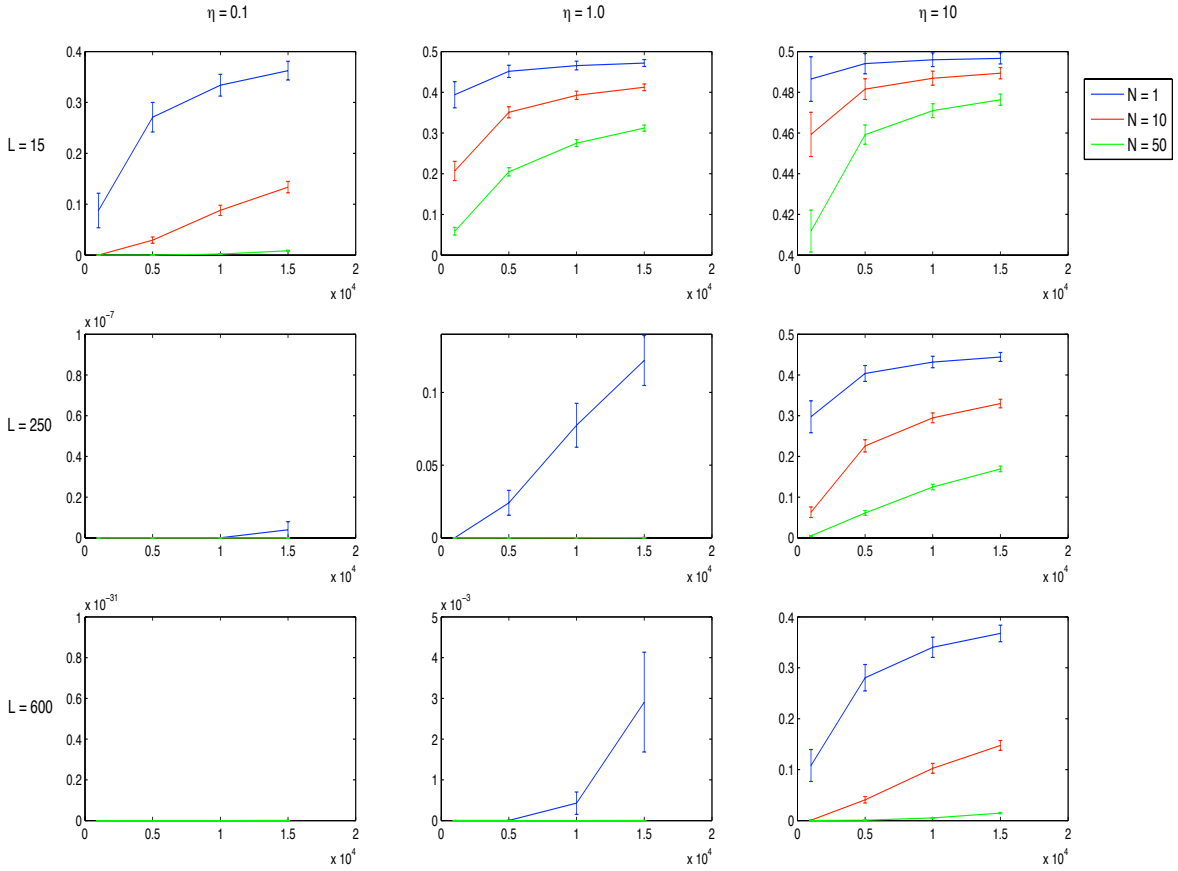

Figure 5.32: The first row shows the Bayesian classifier error rate as a function of the vocabulary size for $L = 15$. The second row shows the Bayesian classifier error rate as a function of the vocabulary size for $L = 250$. The third row shows the Bayesian classifier error rate as a function of the vocabulary size for $L = 600$. Each column corresponds to an increasing value for $\eta \in \{0.1, 1.0, 10\}$. The blue, red, and green lines correspond to $N = 1$, $N = 10$, and $N = 50$ respectively.

Finally, Figure 5.32 shows the BC error rate as the vocabulary size increases. We experiment with $W \in \{1k, 5k, 10k, 15k\}$. We restricted the vocabulary size to more modest values since computing the BC error rate using Equation 5.19 is computationally expensive for large values of W. The pseudocode used to generate these plots is shown in Algorithm 7. The

first row of plots in Figure 5.32 suggests that as the vocabulary size W increases, the BC error rate plateaus (similar to the relationship established in Section 5.2) although more Monte Carlo simulations with higher values of W would be needed to confirm this.

Overall, we see a very similar relationship between the Bayesian classifier error rate and the model parameters as was established in the previous scenario (Section 5.2) in which the class multinomial parameters were observed.

5.4 Summary of contributions

We have presented a theoretical analysis of the accuracy of the multinomial Dirichlet mixture model when used for text classification. The multinomial Dirichlet mixture model is the starting point for many of the more complex statistical models used for probabilistic text classification.

We considered two scenarios in which different information is available. In the first scenario, we observed the parameters of the class multinomial likelihood functions. We derived an expression for the Bayes error of the log likelihood ratio test statistic in terms of an integral over the conditional density of the log likelihood ratio (conditioned on $y = k$). We showed via Monte Carlo simulations that these conditional densities are approximately Normal and supported this claim by appealing to a central limit theorem for multinomial sums [46]. Using this Normal approximation, we derived a closed-form estimate of the Bayes error which allowed us to empirically establish the relationship between the Bayes error and the similarity of the class multinomial parameters (as measured by the Jeffrey's divergence), the Dirichlet hyper-parameter, the document length, and the vocabulary size.

In the second scenario, we observed only a set of representative documents from each class. We derived an analogous classification rule using the ratio of marginalized likelihoods. We

presented an interpretation of this classification rule that elucidated how evidence is accumulated in favor of both classes and highlighted the important role of the natural logarithm function. We then derived an expression for the average error rate of this classifier, which we called the Bayesian classifier error rate, and appealed to the same central limit theorem for multinomial sums to derive a closed-form Normal approximation. Finally, we established the relationship between the Bayesian classifier error rate (computed using our Normal approximation) and the similarity of the class multinomial parameters (as measured by the Jeffrey's divergence), the Dirichlet hyper-parameter, the document length, and the vocabulary size.

In summary:

- We presented an analysis of the accuracy of the multinomial Dirichlet mixture model when used for text classification.
- We considered two scenarios in which different information is available. In the first scenario, we observed the parameters of the class multinomial likelihood functions. In the second scenario, we observed only a set of representative documents from each class. In both cases, we analyzed the relationship between the error rate of the classifier and certain quantities of interest.
- Contributions of the first scenario
  - We presented an expression for the Bayes error of the log likelihood ratio test and we approximate this expression using Monte Carlo simulations.
  - We presented a closed-form approximation to the Bayes error rate by appealing to a central limit theorem for multinomial sums [46].
  - We established the relationship between the Bayes error rate (computed using our Normal approximation) and the similarity of the class multinomial parameters (as measured by the Jeffrey's divergence), the Dirichlet hyper-parameter, the document length, and the vocabulary size.

- Contributions of the second scenario
  - We derived a fully Bayesian classification rule using the ratio of the *marginal* likelihoods.
  - We presented an interpretation of this classification rule that elucidates how evidence is accumulated in favor of both classes.
  - We derived an expression for the average error rate of this classifier which we call the Bayesian classifier error rate.
  - We derived a closed-form approximation to the Bayesian classifier error rate by appealing to the same central limit theorem.
  - We established the relationship between the Bayesian classifier error rate (computed using our Normal approximation) and the similarity of the class multinomial parameters (as measured by the Jeffrey's divergence), the Dirichlet hyperparameter, the document length, and the vocabulary size.

5.5 Future directions

There are a number of promising extensions to the work presented in this chapter. First, we have made some simplifying assumptions in order to make our analysis tractable – for example restricting ourselves to a 2-way classification task and using a symmetric Dirichlet prior over the class multinomial parameters. It would be interesting to relax these assumptions. In particular, allowing for a non-symmetric Dirichlet prior which more faithfully models the distributions seen in real text data.

Another extension would be to consider the relationship between the average error rate of a classifier that uses the maximum a posteriori (MAP) estimates of the class multinomial parameters (estimated given the representative documents $x^{(1)}$ and $x^{(2)}$) and the average

error rate of the fully Bayesian classifier that integrates over the unknown class multinomial parameters.

We presented an interpretation of the marginal likelihood ratio classification rule that elucidates how evidence is accumulated in favor of both hypotheses $y = 1$ and $y = 2$. In particular, we showed that the natural logarithm (and the slope of the natural logarithm) plays a crucial role in determining the class assignment of x. Although lacking a theoretical basis, it would be interesting nonetheless to explore generalizations of the function $f(x, x^{(1)}, x^{(2)})$ that either exchanged the natural logarithm for a function with a constant or increasing slope or parametrized the base of the logarithm to encode different modeling assumptions.

Finally, we have treated the class probabilities θ as fixed and known and we have assumed a mixture model where all of the words in a document are assigned to the same class. Relaxing these assumptions leads to more commonly used models of text, e.g. latent Dirichlet allocation. An obvious extension would be to analyze the error rate of the likelihood ratio classifier for these more complex statistical models.

Chapter 6

Conclusion

In this dissertation, we investigated the usefulness of statistical models of text for both organizing large collections of text documents as well as mining individual text documents.

We presented two new statistical models, SentenceLDA and Multicorpus SentenceLDA, for sentence classification in scientific articles. We created a data set of labeled sentences from scientific articles that span three different domains: computational biology, machine learning, and psychology. Appendix C shows a complete list of indicator words for each label in our annotation scheme. We used this labeled data to evaluate the performance of SentenceLDA and Multicorpus SentenceLDA and to compare the performance of SentenceLDA and Multicorpus SentenceLDA to five other supervised and semi-supervised classifiers. We showed that both SentenceLDA and Multicorpus SentenceLDA are competitive with, or outperform, the baseline classifiers.

Next, we presented a flexible non-parametric statistical model based on latent Dirichlet allocation for learning concept graphs from text. We constructed this prior by specifying a stick-breaking distribution at each node that governs the probability of transitioning from the given node to another node in the graph. We combineed this prior over graphs with

latent Dirichlet allocation to create a new generative model called GraphLDA for learning concept graphs from text. We showed how GraphLDA could be used to learn a concept graph from a collection of documents or to update an existing graph structure in the presence of new labeled documents. We illustrateed the performance of GraphLDA on a set of simulated documents where we increase the proportion of labeled documents used for training. We compareed the performance of GraphLDA to the hierarchical Pachinko allocation model (hPAM) and hierarchical latent Dirichlet allocation (hLDA) using both the empirical likelihood algorithm and the left-to-right algorithm [74]. GraphLDA was competitive with both hLDA and hPAM in terms of per-word log likelihood as computed by the empirical likelihood algorithm. Finally, we illustrated an application of GraphLDA to Wikipedia. We showed how GraphLDA could be used to update a portion of the Wikipedia category graph rooted at the node MACHINE LEARNING given a collection of machine learning abstracts.

Finally, in the last chapter, we moved from application to analysis. We presented an analysis of the accuracy of the multinomial Dirichlet mixture model when used for text classification. We derived a classification rule assuming this generative model for two scenarios where we had access to differing amounts of information. In the first scenario, we observed the parameters of the class multinomial likelihood functions. In the second scenario, we observed only a set of representative documents from each class. In both cases, we derived a closed-form approximation for the average error rate of the classifier by appealing to a central limit theorem for multinomial sums. We then established the relationship between the average error rate and certain quantities of interest, e.g. the document length, providing insight into the multinomial Dirichlet mixture model – a starting point of many of the more complex statistical models used for probabilistic text classification.

Bibliography

[1] S. Agarwal and H. Yu. Automatically classifying sentences in full-text biomedical articles into introduction, method, results and discussion. *Bioinformatics*, 25(23):3174–3180, December 2009.

[2] M. A. Angrosh, S. Cranefield, and N. Stanger. Context identification of sentences in related work sections using a conditional random field: towards intelligent digital libraries. In *Proc. of the 10th Annual Joint Conf. on Digital Libraries*, pages 293–302, 2010.

[3] D. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the Acm*, 57, 2010.

[4] D. Blei and J. D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1:17–35, 2007.

[5] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] S. Bloehdorn, P. Cimiano, and A. Hotho. Learning ontologies to improve text clustering and classification. In *From Data and Inf. Analysis to Know. Eng.: Proc. of the 29th Annual Conf. the German Classification Society (GfKl '05)*, volume 30 of *Studies in Classification, Data Analysis and Know. Org.*, pages 334–341. Springer, Feb. 2005.

[7] J. Boyd-Graber, D. Blei, and X. Zhu. A topic model for word sense disambiguation. In *EMNLP*, 2007.

[8] C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[9] N. V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, 6(1):1–6, 2004.

[10] C. Chemudugunta. *Probabilistic topic models for information retrieval and concept modeling*. PhD thesis, University of California, Irvine, 2009.

[11] G. Chung. Sentence retrieval for abstracts of randomized controlled trials. In *BMC Medical Informatics and Decision Making*, volume 9, 2009.

[12] P. Cimiano, A. Hotho, and S. Staab. Learning concept hierarchies from text using formal concept analysis. *J. Artificial Intelligence Research (JAIR)*, 24:305–339, 2005.

[13] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712, 2006.

[14] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[15] S. Eyheramendy, D. Lewis, and D. Madigan. On the naive bayes model for text categorization, 2003.

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

[17] J. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[18] B. Fortuna, M. Grobelnki, and D. Mladenic. Ontogen: Semi-automatic ontology editor. In *Proceedings of theHuman Computer Interaction International Conference*, volume 4558, pages 309–318, 2007.

[19] K. Fukunaga. *Introduction to statistical pattern recognition.* Academic Press, 1990.

[20] K. Fukunaga and D. Hummels. Bayes error estimation using parzen and k-nn procedures. *IEEE Transactions on Pattern Analysis and Macine Intelligence*, 9(5):634–643, May 1987.

[21] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian data analysis.* Chapman and Hall/CRC, 2009.

[22] A. Gelman, X. li Meng, and H. Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, 6:733–807, 1995.

[23] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the Natl. Academy of the Sciences of the U.S.A.*, 101 Suppl 1:5228–5235, 2004.

[24] Y. Guo, A. Korhonen, M. Liakata, I. Silins, J. Hogberg, and U. Stenius. A comparison and user-based evaluation of models of textual information structure in the context of cancer risk assessment. *BMC Bioinformatics*, 12:69, 2011.

[25] Y. Guo, A. Korhonen, and T. Poibeau. A weakly-supervised approach to argumentative zoning of scientific documents. In *Proc. of the 2011 Conf. on Empirical Methods in Natural Lang. Proc.*, 2011.

[26] S. Gupta and C. D. Manning. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *In Proc. of the Intl. Joint Conference on Natural Language Processing*, 2011.

[27] B. Hachey and C. Grover. Sentence classification experiments for legal text summarisation. In *Proc. of the 17th Annual Conf. on Legal Knowledge and Inf. Systems*, 2004.

[28] K. Hirohata, N. Okazaki, S. Ananiadou, and M. Ishizuka. Identifying sections in scientific abstracts using conditional random fields. In *Proc. of the Intl. Joint Conference on Natural Language Processing*, 2008.

[29] W. Jien-Chen, C. Yu-Chia, H.-C. Liou, and J. Chang. Computational analysis of move structures in academic abstracts. In *Proc. of the COLING/ACL on Interactive presentation sessions*, COLING-ACL '06, pages 41–44, 2006.

[30] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

[31] S. N. Kim, D. Martinez, and L. Cavedon. Automatic classification of sentences for evidence based medicine. In *Proc. of the ACM Fourth Intl. Workshop on Data and Text Mining in Biomedical Informatics*, pages 13–22, 2010.

[32] L. Kuncheva. On the optimality of naive bayes with dependent binary features. *Pattern Recognition Letters*, 27(7):830–837, 2006.

[33] L. Kuncheva and Z. Hoare. Error-dependency relationships for the naive bayes classifier with binary features. *IEEE Transactions on Pattern Analysis and Macine Intelligence*, 30(4):735–740, 2008.

[34] W. Li, D. Blei, and A. McCallum. Nonparametric bayes pachinko allocation. In *Proceedings of the Twenty-Third Annual Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 243–250, 2007.

[35] Y. Li, S. L. Gorman, and N. Elhadad. Section classification in clinical notes using supervised hidden markov model. In *Proc. of the 1st ACM Intl. Health Informatics Symposium*, pages 744–750, 2010.

[36] M. Liakata, S. Teufel, A. Siddharthan, and C. R. Batchelor. Corpora for the conceptualisation and zoning of scientific papers. In *Language Resources and Evaluation (LREC)*, 2010.

[37] R. V. Lindsey, W. P. H. III, and M. J. Stipicevic. A phrase-discovering topic model using hierarchical pitman-yor processes. In *Proc. of the 2012 Joint Conf. on Empirical Methods in Natural Lang. Proc.*, pages 214–222, 2012.

[38] N. Madnani, M. Heilman, J. Tetreault, and M. Chodorow. Identifying high-level organizational elements in argumentative discourse. In *Proc. of the 2012 Conf. of the North American Chapter of the Assoc. for Comptl. Ling.: Human Lang. Tech.*, pages 20–28, 2012.

[39] R. E. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the dirichlet distribution. In *Proc. of the 22nd Intl. Conf. on Machine Learning*, pages 545–552, 2005.

[40] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[41] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *In AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998.

[42] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[43] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 21st Intl. Conf. on Machine Learning*, 2007.

[44] D. M. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial. In *Uncertainty in Artificial Intelligence*, pages 411–418, 2008.

[45] T. Minka. Estimating a dirichlet distribution. Technical report, 2000.

[46] C. Morris. Central limit theorems for multinomial sums. *Annals of Statistics*, 3(1):165–188, 1975.

[47] J. E. Mosimann. On the compound multinomial distribution, the multivariate β-distribution, and correlations among populations. *Biometrika*, 49:65–89, 1962.

[48] R. Neal. Slice sampling. *Annals of Statistics*, 31:705–767, 2000.

[49] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10:1801–1828, 2009.

[50] D. Newman and S. Block. Probabilistic topic decomposition of an eighteenth-century american newspaper. *Journal of the American Society for Information Science and Technology*, 2006.

[51] D. Newman, N. Koilada, J. H. Lau, and T. Baldwin. Bayesian text segmentation for index term identification and keyphrase extraction. In *COLING*, pages 2077–2092, 2012.

[52] L. Pardo. *Statistical inference based on divergence measures*. Chapman and Hall/CRC, 2006.

[53] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers, 1988.

[54] I. Porteous, A. Ihler, P. Smyth, and M. Welling. Gibbs sampling for coupled infinite mixture models in the stick-breaking representation. In *Proceedings of UAI 2006*, pages 385–392, July 2006.

[55] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In *Proc. of the 2009 Conf. on Empirical Methods in Natural Lang. Processing*, pages 248–256, 2009.

[56] J. Randolph. Free-marginal multirater kappa: an alternative to fleiss' fixed-marginal multirater kappa. In *Joensuu Learning and Instruction Symposium*, 2005.

[57] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proc. of the 20th Intl. Conf. on Machine Learning*, pages 616–623, 2003.

[58] I. Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[59] I. Rish, J. Hellerstein, and T. Jayram. An analysis of data characteristics that affect naive bayes performance. Technical report, IBM Watson Research Center, 2001.

[60] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, 2004.

[61] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 88(1-2):157–208, July 2011.

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

[63] M. Shimbo, T. Yamaski, and Y. Matsumoto. Using sectioning information for text retrieval: a case study with medline abstracts. In *Proc. of the 2nd Intl. Workshop on Active Mining*, pages 32–41, 2003.

[64] F. Sinz and M. Roffilli. Universvm, 2012.

[65] R. Soricut and D. Marcu. Sentence level discourse parsing using syntactic and lexical information. In *Proc. of the 2003 Conf. of the North American Chapter of the Assoc. for Comptl. Ling. on Human Lang. Tech.*, pages 149–156, 2003.

[66] Y. W. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

[67] S. Teufel. *Argumentative zoning: information extraction from scientific text*. PhD thesis, School of Informatics, University of Edinburgh, 1999.

[68] S. Teufel and M.-Y. Kan. Robust argumentative zoning for sensemaking in scholarly documents. In *Proc. of the 2009 Intl. Conf. on Advanced Language Technologies for Digital Libraries*, NLP4DL'09/AT4DL'09, pages 154–170, 2011.

[69] S. Teufel and M. Moens. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445, Dec 2002.

[70] S. Teufel, A. Siddharthan, and C. R. Batchelor. Towards discipline-independent argumentative zoning: evidence from chemistry and computational linguistics. In *Proc. of Empirical Methods on Natural Lang. Proc.*, pages 1493–1502, 2009.

[71] K. Tumer and J. Ghosh. Bayes error rate estimation using classifier ensembles. *International Journal of Smart Engineering System Design*, 5:95–105, 2003.

[72] E. van Dyk and E. Barnard. Naive bayesian classifiers for multinomial features: a theoretical analysis. *South African Computer Journal*, 40:37–43, 2008.

[73] A. Varga, D. Preotiuc-Pietro, and F. Ciravegna. Unsupervised document zone identification using probabilistic graphical models. In *Proc. of the 8th Intl. Conf. on Lang. Resources and Eval.*, May 2012.

[74] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Intl. Conf. on Machine Learning (ICML 2009)*, 2009.

[75] C. Wang and D. Blei. Collaborative topic modeling for recommending scientific articles. In *Proc. of the 17th ACM SIGCKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 448–456, 2011.

[76] A. Webb. *Statistical pattern recognition*. Wiley, 2002.

[77] H. Zhang. Exploring conditions for the optimality of naive bayes. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(2):183–198, 2005.

Appendices

A Derivations for GraphLDA

A.1 Sampling probability of a new path

Let $\vec{v}_x = (v_{x1}, v_{x2}, \ldots)$ denote the collection of beta random variables associated with node x in the graph. The probability of selecting the ith feasible node from $\mathcal{P}_x(\cdot)$ (the stick-breaking distribution at node x) is given by

$$\pi_{x1} = v_{x1} \quad \text{and} \quad \pi_{xi} = v_{xi} \prod_{r=1}^{i-1} (1 - v_{xr}) \text{ for } i = 2, 3, \ldots \tag{A.1}$$

Posterior probability of an edge

Let $\mathbf{p}$ be a collection of paths. We want to compute the posterior probability of selecting the ith feasible node from $\mathcal{P}_x$ conditioned on the paths $\mathbf{p}$. We compute this probability by

integrating over the vector of beta random variables $\vec{v}_x$.

$$
\begin{aligned}
p(x \to y_i|\mathbf{p}) &= \int_{\vec{v}_x} p(x \to y_i|\vec{v}_x) \cdot p(\vec{v}_x|\mathbf{p})\, d\vec{v}_x \\
&= \int_{\vec{v}_x} p(x \to y_i|\vec{v}_x) \cdot p(\vec{v}_x|N_{(x,y_1)}, N_{(x,y_2)}, \ldots)\, d\vec{v}_x
\end{aligned}
\tag{A.2}
$$

Notice that in the second line we replace $\mathbf{p}$ with the sufficient statistics $N_{(x,y_j)}$ for $j = 1, 2, \ldots$. The count $N_{(x,y_j)}$ is the number of paths in $\mathbf{p}$ that contain the edge (x, y_j). Given the edge counts $N_{(x,y_j)}$, $\vec{v}_x$ is independent of the remaining edge counts $N_{(i,l)}$ for $i \neq x$.

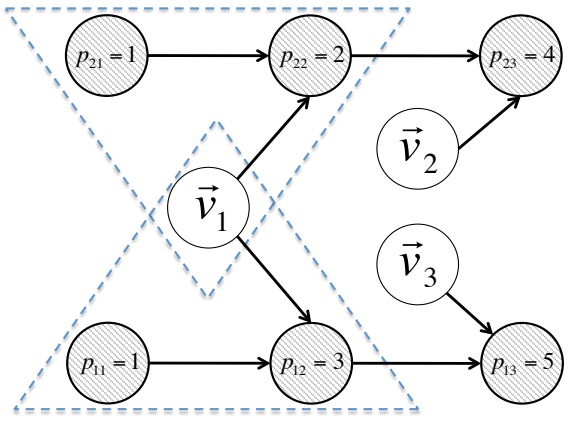

Figure A.1: Plate notation for a small example with two documents. The path variables are shaded to indicate that they are observed. The dashed line shows the Markov blanket for $\vec{v}_1$

This independence can be seen in the plate notation for GraphLDA. Figure A.1 shows the partial plate notation for a small example with two documents. The first document has path $p_1 = (1, 3, 5)$ and the second document has path $p_2 = (1, 2, 4)$. The path nodes have been shaded to indicate that they are observed, i.e. we are conditioning on the paths $\mathbf{p} = \{p_1, p_2\}$. Given the paths $\mathbf{p}$, the Markov blanket for the set of beta random variables $\vec{v}_1$ consists of the nodes $\{p_{11}, p_{12}, p_{21}, p_{22}\}$. This is shown with a dashed line. Since the path $\vec{v}_1 \to p_{12} \leftarrow p_{11}$ contains converging arrows at p_{12}, if we condition on the value of p_{12} then $\vec{v}_1$ becomes dependent upon the value of p_{11}. Similarly, if we condition on p_{22} then $\vec{v}_1$ is dependent upon

the value of p_{21}. Thus, the Markov blanket of $\vec{v}_1$ consists of the pairs of random variables $(p_{ij}, p_{i,j+1})$ where $p_{ij} = 1$. Given these random variables, $\vec{v}_1$ is independent of the other random variables in the Bayesian network.

The integral in Equation A.2 is equivalent to Equation 9 in the paper by Porteous et al. [54] where $z_n = i$ in their notation is equivalent to $x \to y_i$, V is equivalent to $\vec{v}_x$, and $Z_{(-n)}$ is equivalent to the edge counts$N_{(x,y_i)}$. This integral evaluates to the expected value of π_{x,y_i} conditioned on $\mathbf{p}$ which is given by,

$$E[\pi_{x,y_i}|\mathbf{p}] = \frac{\alpha + N_{(x,y_i)}}{\alpha + \beta + N_{(x,\geq y_i)}} \prod_{r=1}^{i-1} \frac{\beta + N_{(x,>y_r)}}{\alpha + \beta + N_{(x,\geq y_r)}}$$

See Equation 6 in [54]. Thus, we have

$$p(x \to y_i|\mathbf{p}) = \frac{\alpha + N_{(x,y_i)}}{\alpha + \beta + N_{(x,\geq y_i)}} \prod_{r=1}^{i-1} \frac{\beta + N_{(x,>y_r)}}{\alpha + \beta + N_{(x,\geq y_r)}} \tag{A.3}$$

An infinite number of feasible nodes

Assume that the node x has a maximum of M feasible nodes. Let $\{y_1, y_2, \ldots, y_M\}$ be the set of such feasible nodes where y_1 has the first position in the stick-breaking permutation, y_2 has the second position, y_3 the third, and so on. If y_m is the last node with a nonzero count $N_{(x,y_m)}$ and $m \ll M$, it is convenient to compute the probability of transitioning to y_i, for each $i \leq m$, and the probability of transitioning to a node higher than y_m which is given by

$$\sum_{k=m+1}^{M} p(x \to y_k|\mathbf{p}) = \sum_{k=m+1}^{M} \frac{\alpha + N_{(x,y_k)}}{\alpha+\beta+N_{(x,\geq y_k)}} \prod_{r=1}^{k-1} \frac{\beta + N_{(x,>y_r)}}{\alpha+\beta+N_{(x,\geq y_r)}}$$

Note that since the counts $N_{(x,y_i)}$, $N_{(x,>y_i)}$, and $N_{(x,\geq y_i)}$ equal zero for $i \geq m+1$ we can rewrite the above summation as

$$\begin{aligned}
&\sum_{k=m+1}^{M} p(x \to y_k|\mathbf{p}) \\
&= \prod_{r=1}^{m} \frac{\beta + N_{(x,>y_m)}}{\alpha+\beta+N_{(x,\geq y_m)}} \left[\frac{\alpha}{\alpha+\beta}\left(\frac{\beta}{\alpha+\beta}\right)^0 + \ldots + \frac{\alpha}{\alpha+\beta}\left(\frac{\beta}{\alpha+\beta}\right)^{M-(m+1)+1} \right] \\
&= \Delta \cdot \left(\frac{\alpha}{\alpha+\beta}\right) \sum_{k=0}^{M-m} \left(\frac{\beta}{\alpha+\beta}\right)^k \\
&= \Delta \cdot \left(\frac{\alpha}{\alpha+\beta}\right) \left(\frac{1 - \left(\frac{\beta}{\alpha+\beta}\right)^{M-m}}{1 - \left(\frac{\beta}{\alpha+\beta}\right)} \right) \\
&= \Delta \left(1 - \left(\frac{\beta}{\alpha+\beta}\right)^{M-m}\right)
\end{aligned}$$

where $\Delta \equiv \prod_{r=1}^{m} \frac{\beta+N_{(x,>y_m)}}{\alpha+\beta+N_{(x,\geq y_m)}}$. If M equals infinity, then we have

$$\begin{aligned}
\sum_{k=m+1}^{\infty} p(x \to y_k|\mathbf{p}) &= \Delta \cdot \left(\frac{\alpha}{\alpha+\beta}\right) \sum_{k=0}^{\infty} \left(\frac{\beta}{\alpha+\beta}\right)^k \\
&= \Delta \cdot \left(\frac{\alpha}{\alpha+\beta}\right) \left(\frac{1}{1 - \left(\frac{\beta}{\alpha+\beta}\right)} \right) \\
&= \Delta
\end{aligned} \tag{A.4}$$

Thus, using Equation A.3 we can compute the probability of transitioning from x to y_i for

each $i \leq m$ and the probability of transitioning to a node higher than y_m using Equation A.4.

Probability of a path

Let p_d denote the path assignment for document d. Let $\mathbf{p}_{-d}$ denote the path assignment for every document in our corpus except for document d. Let $\hat{p} = (\hat{p}_1, \ldots, \hat{p}_\lambda)$ represent a path in the graph. Then we want to compute $p(p_d = \hat{p}|\mathbf{p}_{-d})$ for all paths $\hat{p}$.

Although there are an infinite number of nodes in the graph, we are only concerned with those finitely many nodes that are contained in at least one path in $\mathbf{p}_{-d}$. We say that these nodes are "active" and we denote the number of such nodes as T. To compute $p(p_d = \hat{p}|\mathbf{p}_{-d})$ we must integrate over the vectors of beta random variables $\vec{v}_1, \ldots, \vec{v}_T$ for the active nodes.

$$
\begin{aligned}
p(p_d = \hat{p}|\mathbf{p}_{-d}) &= \int_{\vec{v}_1} \cdots \int_{\vec{v}_T} p(p_d = \hat{p}|\vec{v}_1, \ldots, \vec{v}_T) \cdot p(\vec{v}_1, \ldots, \vec{v}_T|\mathbf{p}_{-d})\ d\vec{v}_1, \ldots d\vec{v}_T \\
&= \int_{\vec{v}_1} \cdots \int_{\vec{v}_T} \prod_{l=1}^{\lambda-1} p(\hat{p}_l \to \hat{p}_{l+1}|\vec{v}_1, \ldots, \vec{v}_T) \cdot \prod_{t=1}^{T} p(\vec{v}_t|\mathbf{p}_{-d})\ d\vec{v}_1, \ldots d\vec{v}_T \\
&= \int_{\vec{v}_1} \cdots \int_{\vec{v}_T} \prod_{l=1}^{\lambda-1} p(\hat{p}_l \to \hat{p}_{l+1}|\vec{v}_{\hat{p}_l}) \cdot \prod_{t=1}^{T} p(\vec{v}_t|\mathbf{p}_{-d})\ d\vec{v}_1, \ldots d\vec{v}_T \\
&= \prod_{l=1}^{\lambda-1} \int_{\vec{v}_{\hat{p}_l}} p(\hat{p}_l \to \hat{p}_{l+1}|\vec{v}_{\hat{p}_l}) \cdot p(\vec{v}_{\hat{p}_l}|\mathbf{p}_{-d})\ d\vec{v}_{\hat{p}_l} \cdot \prod_{s \notin \hat{p}} \int_{\vec{v}_s} p(\vec{v}_s|\mathbf{p}_{-d})\ d\vec{v}_s \\
&= \prod_{l=1}^{\lambda-1} \int_{\vec{v}_{\hat{p}_l}} p(\hat{p}_l \to \hat{p}_{l+1}|\vec{v}_{\hat{p}_l}) \cdot p(\vec{v}_{\hat{p}_l}|\mathbf{p}_{-d})\ d\vec{v}_{\hat{p}_l} \\
&= \prod_{l=1}^{\lambda-1} p(\hat{p}_l \to \hat{p}_{l+1}|\mathbf{p}_{-d})
\end{aligned}
\tag{A.5}
$$

In the second line of Equation A.5, we expand the likelihood $p(p_d = \hat{p}|\vec{v}_1, \ldots, \vec{v}_T)$ as a product over the edges in the path. We also expand the conditional distribution $(\vec{v}_1, \ldots, \vec{v}_T|\mathbf{p}_{-d})$ as a product over the active nodes. That is, given the path assignments $\mathbf{p}_{-d}$ the beta random

variables $\vec{v}_i$ are independent of the beta random variables $\vec{v}_j$ for $i \neq j$. Again, this can be seen in the plate notation in Figure A.1 where $p(\vec{v}_1, \vec{v}_3|\mathbf{p}) = p(\vec{v}_1|\vec{v}_3, \mathbf{p})p(\vec{v}_3|\mathbf{p})$ by the chain rule. As discussed above, the probability of $\vec{v}_1$ is independent of $\vec{v}_3$ given its Markov blanket which consists of the random variables $\{p_{11}, p_{12}, p_{21}, p_{22}\}$. Thus, $p(\vec{v}_1|\vec{v}_3, \mathbf{p}) = p(\vec{v}_1|\mathbf{p})$ and $p(\vec{v}_1, \vec{v}_3|\mathbf{p}) = p(\vec{v}_1|\mathbf{p})p(\vec{v}_3|\mathbf{p})$.

In the third line, we note that the probability of the edge $(\hat{p}_l, \hat{p}_{l+1})$ depends only on the beta random variables for the parent node $\vec{v}_{\hat{p}_l}$. In the next line, we note that the integrals over the $\vec{v}_t$ can be factored into a product of T independent integrals. That is, each probability in the integral is a function of only one of the variables we are integrating over, $\vec{v}_t$ for some t. Thus, we factor the expression into a product of T integrals where each integral involves only those probabilities that contain the variable of integration $\vec{v}_t$. For those active nodes that correspond to a node on the path, we have a product over the edges of the path. For those active nodes that do not correspond to a node on the path (denoted as s), we have a product over the integral of the conditional distribution $p(\vec{v}_s|\mathbf{p}_{-d})$. Note that each of these integrals evaluates to 1 and the entire product over s evaluates to 1. Thus, we are left with a product over each edge in our path.

Finally, in the last line we recognize that each integral in the product can be computed using Equation A.3.

A.2 Computing per-word log likelihood

Let x_d^{test} be the dth document in the test set. Let Φ^{train} be a point estimate of the word distributions learned at train time for each node in the graph. Let $\mathbf{p}^{\text{train}}$ be the path assignments for the documents in the training set. Then we want to compute the probability of the test document given the learned word distributions and the path assignments, $p(x_d^{\text{test}}|\Phi^{\text{train}}, \mathbf{p}^{\text{train}})$. We do so by marginalizing over the unknown value of the random vari-

ables $\{p_d, \tau_d, l_{di}\}$ where p_d is the path assignment for the test document, τ_d is the parameter for the distribution over levels for the test document, and l_{di} is the level assignment for the ith word in the test document.

$$
\begin{aligned}
&p(x_d^{\text{test}}|\Phi^{\text{train}}, \mathbf{p}^{\text{train}}) \\
&= \sum_{p=1}^{P} p(x_d^{\text{test}}|p_d = p, \Phi^{\text{train}}) \cdot p(p_d = p|\mathbf{p}^{\text{train}}) \\
&= \sum_{p=1}^{P} \left[\prod_{i=1}^{N} \sum_{l=1}^{\lambda_p} p(x_{di}^{\text{test}}|\phi_{p[l]}^{\text{train}}) \left(\int_{\tau_d} p(l_{di} = l|p_d = p, \tau_d) \cdot p(\tau_d|a, b)\, d\tau_d \right) \right] \cdot p(p_d = p|\mathbf{p}^{\text{train}})
\end{aligned}
$$

The probability $p(x_{di}^{\text{test}}|\phi_{p[l]}^{\text{train}})$ is the probability of the word x_{di}^{test} according to the topic at the node $p[l]$. Let $x_{di}^{\text{test}} = w$. Then this probability is given by

$$
p(x_{di}^{\text{test}}|\phi_{p[l]}^{\text{train}}) = \frac{\eta + n_{p[l],w}}{W\eta + n_{p[l],\cdot}}
$$

where $n_{p[l],w}$ is the number of times the word token w was assigned to the node $p[l]$ in the training set and $n_{p[l],\cdot} = \sum_{w'} n_{p[l],w'}$. The integral over τ_d can be approximated using the Monte Carlo estimate,

$$
\begin{aligned}
p(l_{di} = l|p_d = p) &= \int_{\tau_d} p(l_{di} = l|p_d = p, \tau_d) \cdot p(\tau_d|a, b)\, d\tau_d \\
&= \frac{1}{R} \sum_{r=1}^{R} \frac{(1 - \tau^{(r)})^{l-1} \tau^{(r)}}{1 - (1 - \tau^{(r)})^{\lambda_p}}
\end{aligned}
$$

where $\tau^{(r)} \sim \text{Beta}(a, b)$. Finally, the probability of the path $p_d = p$ given the path assign-

ments from the documents in the training set $\mathbf{p}^{\text{train}}$ is given by Equation A.5.

B Annotation procedure

To find people who were willing to label sentences, we used word-of-mouth, sent an email advertisement, and posted announcements in academic buildings at the University of California, Irvine[1]. Each participant received a directory containing a text version of the article to be labeled, a pdf version of the article to be labeled, a 7-page instruction manual, and three examples of already labeled articles. Participants were told to read the instruction manual and encouraged to look at the three examples before starting the task. The instructions were taken and modified from the instructions written by Teufel [67]. Figure B.2 shows the decision tree included in the instruction manual that participants used to label sentences.

Since the participants were not experts in the scientific domain of the articles nor experts at annotating sentences, and they received no training beyond the instruction manual we used two methods to ensure the quality of their work. First, the last sentence in the instruction manual asked the participants to respond by email with the time of day. We used this to assess who read the entire instruction manual. Second, we inspected each labeled article. We checked that any sentence with the words "we" or "our" were labeled with either OWN or AIM, since it was made clear in the instruction manual that all sentence's describing the author's own work should have one of these two labels. We also checked for general understanding of the task, e.g. one person marked all sentences that did not discuss the author's own work as either BASE or CONTRAST – they marked no MISC sentences. Any participant that failed to pass these two quality checks was not asked to participate any further and the labels they provided were not used. Those that did pass the quality checks were paid a $15 dollar gift card to Amazon and were allowed to do additional articles for $5 each (or $7 for the PLOS articles) up to a total of a $50 Amazon gift card.

[1]This task did not constitute human subjects research since we were not obtaining information *about* the people who helped to label sentences. We were given an exemption by the UCI institutional review board (IRB).

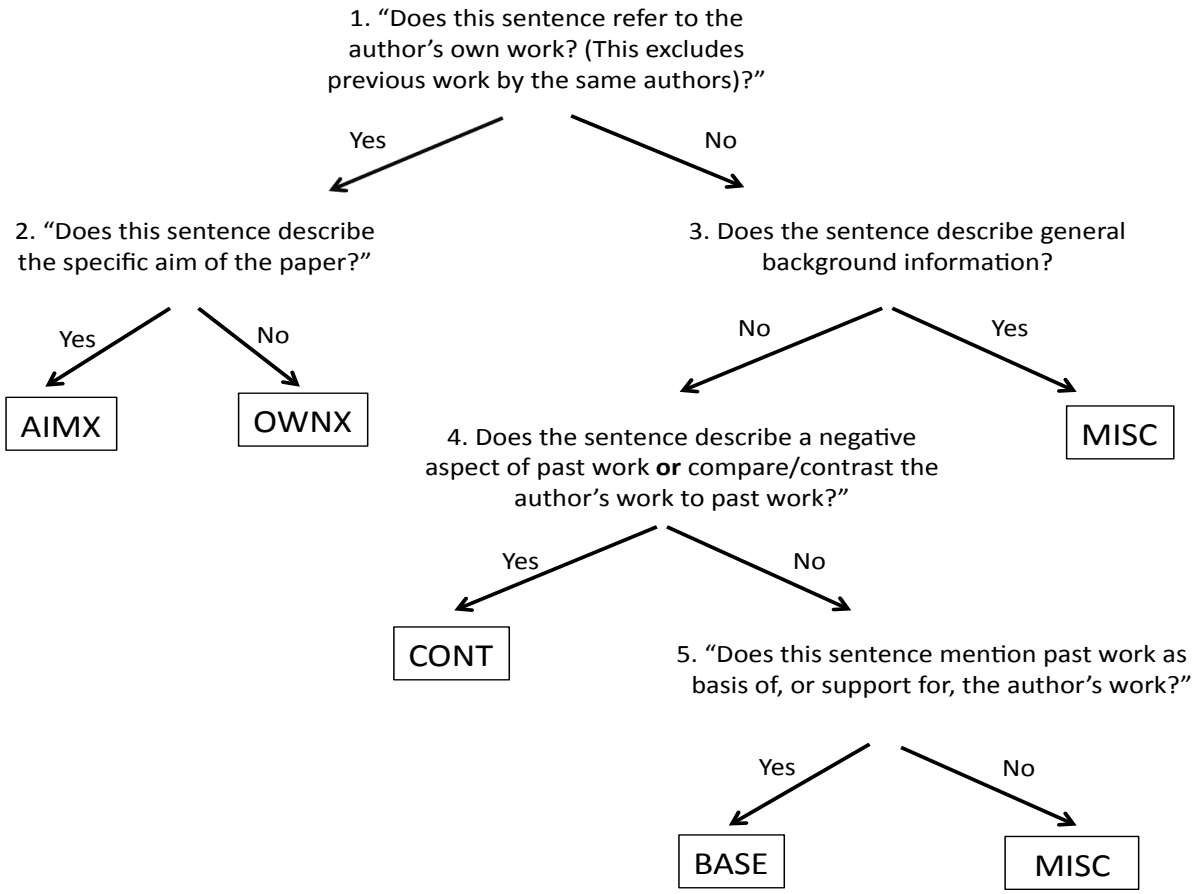

Figure B.2: Decision tree used by participants to help guide the labeling process

Each article was annotated independently by two people. The author of this thesis then provided a third independent annotation. We took as the ground truth for each sentence the majority label. For 2 sentences, there was no majority label in which case the author of this thesis made a decision as to which of the three labels was the best.

To measure inter-annotator agreement, we computed the precision between one annotator's labels (used as ground truth) and the labels provided by the remaining two annotators[2]. We used each annotator in turn as the ground truth. Table B.1 and Table B.2 shows this inter-annotator precision averaged over the three annotators for the validity set and test set respectively. The label CONTRAST consistently had the lowest agreement between

[2]Here we use the more specific term *annotators* to refer to those participants that passed the quality checks.

	AIM	OWN	CONTRAST	BASE	MISC
PLOS	0.87 ± 0.13	0.92 ± 0.09	0.82 ± 0.05	0.90 ± 0.08	0.92 ± 0.05
ARXIV	0.73 ± 0.14	0.79 ± 0.12	0.62 ± 0.07	0.73 ± 0.25	0.88 ± 0.11
JDM	0.64 ± 0.21	0.82 ± 0.10	0.63 ± 0.07	0.87 ± 0.16	0.94 ± 0.03
Average	**0.75 ± 0.19**	**0.84 ± 0.12**	**0.69 ± 0.11**	**0.83 ± 0.19**	**0.91 ± 0.08**

Table B.1: Inter-annotator precision for the **validity** set. Each annotator in turn was used as ground truth, and the precision of the remaining two annotators was computed.

	AIM	OWN	CONTRAST	BASE	MISC
PLOS	0.72 ± 0.23	0.89 ± 0.08	0.82 ± 0.17	0.80 ± 0.27	0.94 ± 0.02
ARXIV	0.75 ± 0.08	0.71 ± 0.10	0.64 ± 0.22	0.62 ± 0.26	0.82 ± 0.14
JDM	0.90 ± 0.10	0.85 ± 0.16	0.63 ± 0.22	0.75 ± 0.22	0.93 ± 0.05
Average	**0.79 ± 0.17**	**0.81 ± 0.14**	**0.69 ± 0.22**	**0.72 ± 0.26**	**0.90 ± 0.10**

Table B.2: Inter-annotator precision for the **test** set. Each annotator in turn was used as ground truth, and the precision of the remaining two annotators was computed.

annotators, and the ARXIV corpus had the lowest agreement among annotators for all labels except AIM. It is not surprising that agreement was lowest for ARXIV since ARXIV is the least uniform of the three corpora; the remaining two corpora are journals with specific mission statements and guidelines for articles. In contrast, there are no guidelines for articles uploaded to ARXIV.

We also computed the fixed-marginal [17] and the free-marginal [56] multi-rater kappa scores. Both scores have the form $\frac{P_o - P_e}{1 - P_e}$ where P_e is the proportion of agreement between annotators that is due to random chance and P_o is the observed proportion of agreement between annotators. The fixed-marginal kappa is used in cases where annotators know beforehand the number of instances (i.e. sentences) that should be assigned to each category (i.e. label). In this case, the marginal probability of observing each category is known and fixed and P_e and P_o are defined as follows:

$$P_o \equiv \frac{1}{Nn(n-1)}\left[\left(\sum_{i=1}^{N}\sum_{j=1}^{K} n_{ij}^2\right) - Nn\right]$$
$$P_e \equiv \sum_{j=1}^{K}\left(\frac{1}{Nn}\sum_{i=1}^{N} n_{ij}\right)^2$$

where N is the number of instances (sentences), n is the number of annotators, K is the number of categories (labels), and n_{ij} is the number of annotators that assigned instance i to category j. The free-marginal kappa is used in cases where annotators do not know beforehand the number of instances that should be assigned to each category. In this case, P_o has the same definition but $P_e \equiv \frac{1}{K}$. Both the fixed-marginal and free-marginal kappa scores are between 1 and -1 where 1 indicates inter-annotator agreement better than chance, 0 indicates inter-annotator agreement no better than random chance and -1 indicates inter-annotator agreement worse than chance.

In our case, annotators were encouraged to look at 3 example articles from which it was easy to discern the frequency of each label – e.g. that MISC sentences occurred most often, that there was almost always one AIM sentence per section, and that BASE and CONTRAST sentences were relatively rare. In addition, in the instruction manual annotators were told to find at least one AIM sentence in each section and that BASE and CONTRAST sentences may not occur at all. Thus, although annotators did not know the exact number of sentences to assign to each category they did have some knowledge of the marginal probability of each label. For this reason, we computed both the fixed-marginal and free-marginal kappa scores. For the test set, the number of sentences was $N = 472$, the number of annotators was $n = 3$, and the number of labels was $K = 5$. The fixed-marginal kappa score was 0.83 and the free-marginal kappa score was 0.78. For the validity set, the number of sentences was $N = 567$. The fixed-marginal kappa score was 0.86 and the free-marginal kappa score was 0.82. These kappa scores indicate a high-level of agreement between annotators for both

the test and validity sets.

C Stopword and indicator lists for sentence classification

Stopwords

a, an, and, are, as, be, by, can, for, from, has, have, in, is, it, of, on, or, such, that, the, these, to, which, with

Indicators for aim

address, analyze, answer, applied, approach, argue, assess, compare, concentrate, current, designed, developed, discover, discuss, establish, examine, examines, formalize, give, goal, have, how, identification, influence, influences, interested, introduce, investigate, investigated, investigates, investigation, main, make, model, motivate, new, paper, present, propose, purpose, show, study, suggest, survey, theoretical, this, to, use, validate, was, we, work

Indicators for own

accomplish, accuracy, accurately, algorithm, analysis, analyzed, answer, apply, approach, are, ask, claim, classify, compare, comparison, comparisons, concludes, conclusions, consider, constructed, contributions, converted, couple, created, define demonstrate, derive, detect, determined, developed, discovered, discuss, discussion, examined, expected, explain, exploit findings, focus, followed, formulated, found, future, generalize, give, have, here, hypothesis, hypothesize, I, identified, identify, illustrate, implications, indicate, insights, is, measured, method, model, motivation, named, new, notation, novel, observe, observed, obtain, offers, our, paper, predict, predicted, predicts, present, presents, propose, provided, rate, reformulate, result, results, reveals, section, sections, show, significantly, simulations, study, suggest, suggesting, suggests, terminology, test, this, to, treat, true-positive, used, using, was, we, were, work

Indicators for contrast

accurately, although, apparently, artificial, assume, assumed, capture, confidence, difficult, directly, disparate, drawbacks, early, even, expensive, fall, formidable, full, generally, handful, high, however, ignore, increasingly, indirect, infeasible, less, limited, little, low, many, may, might, missing, most, must, nevertheless, not, notorious, obstacles, often, only, readily, reasonable, relatively, require, restricted, seems, short, shortcomings, sometimes, somewhat, sparse, substantial, tailored, trades-offs, typically, unfortunately, unlike, unrealistically, variability, variable, whereas, while, without, yet

Indicators for base

earlier, extend, extended, foundation, legacy, on, previously, refined, relies, reuse, similar, spirit, using

The position of the sentence in the section.
The section (abstract or introduction)
Does the sentence contain a citation?
Does the previous sentence contain a citation?
Does the sentence contain "we", "I", or "the authors"?
Is this the first sentence in the section to contain "we", "I", or "the authors"?

Table D.3: Features used in Dirichlet multinomial regression

D Dirichlet multinomial regression for sentence classification

Dirichlet multinomial regression [44] (DMR) is a variation of latent Dirichlet allocation (LDA) that allows the user to condition on additional features beyond just the bag of words.

We describe DMR in the context of sentence classification. Recall the notion of **groups** from SentenceLDA and Multicorpus SentenceLDA. A group is a generalization of a document or a sentence. Sentences that have a similar distribution over labels are grouped together to share statistical strength. We use all of the sentences in the group to learn a single distribution over labels. This distribution over labels is denoted θ_g where g is the group index. For SentenceLDA and Multicorpus SentenceLDA, θ_g is sampled from a Dirichlet prior with hyper-parameters $A_g = (A_{g1}, \ldots, A_{gL})$. DMR introduces a mechanism for learning these hyper-parameters (as opposed to estimating and fixing them) by regressing on a set of group-specific features. One disadvantage of using DMR for our task is that each sentence must be its own group, i.e. $G = S$, since the features we are interested in using are expressed naturally at the sentence level. The full list of features that we use is shown in Table D.3.

The generative model for DMR applied to the task of sentence classification is shown in Table D.4. Let K be the number of features. Then $\lambda_l = (\lambda_{l1}, \ldots, \lambda_{lK})$ is the vector of regression coefficients for label l. The "weight" of label l in sentence s is given by the dot product $\pi_{sl} = x_s^\intercal \lambda_l$ which is exponentiated to ensure positivity. The resulting vector

1. For label $l \in \{1, \ldots, L\}$

   Sample regression coefficients $\lambda_l \sim \text{Normal}(0, \sigma^2 I)$

   Sample distribution over words $\phi_l \sim \text{Dirichlet}(B_{l1}, \ldots, B_{lW})$

2. For sentence $s \in \{1, \ldots S\}$

   Compute $\pi_{sl} = \exp(x_s^\intercal \lambda_l)$ for all l

   Compute $\alpha_{sl} = \alpha_0 \cdot \frac{\pi_{sl}}{\sum_{j=1}^{L} \pi_{sj}}$ for all l

   Sample distribution over labels $\theta_s \sim \text{Dirichlet}(\alpha_{s1}, \ldots, \alpha_{sL})$

3. Follow generative model for LDA to generate words

Table D.4: Generative model for Dirichlet multinomial regression (DMR) applied to sentence classification.

$\pi_s = [\pi_{s1}, \ldots, \pi_{sL}]$ is then normalized and scaled by α_0. The scalar α_0 allows us to control the concentration, i.e. strength, of the Dirichlet prior. The resulting vector is then used as the hyper-parameters for the Dirichlet prior over θ_s. Note the subscript for θ_s is now "s" since each sentence is its own group, and therefore each sentence has its own distribution over labels. Generating the words in the sentence then proceeds in a manner analogous to LDA.

We still use the hyper-parameter matrix B for the Dirichlet prior over the label distributions ϕ_l. Also, we use one latent assignment variable for each word. We do not experiment with one latent assignment variable for the entire sentence.

D.1 Learning the regression coefficients

Given a collection of sentences, the word variables $\mathbf{w} = \{w_{si}\}$ and the feature vectors $\mathbf{x} = \{x_s\}$ are known and observed. We again marginalize out the label distributions $\boldsymbol{\phi} = \{\phi_l\}$ and the sentence distributions $\boldsymbol{\theta} = \{\theta_s\}$ from the joint distribution $p(\mathbf{w}, \mathbf{x}, \mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\lambda})$. What remains to learn are the regression coefficients $\boldsymbol{\lambda} = \{\lambda_{lk}\}$ and the cluster assignment variables $\mathbf{z} = \{z_{si}\}$. Given the regression coefficients, we can deterministically compute the hyper-

parameters $\boldsymbol{\alpha} = \{\alpha_{sl}\}$.

We use a collapsed Gibbs sampler to infer the latent assignments $\mathbf{z}$. The sampling equation for z_{si} is equivalent to Equation 3.1 where the hyper-parameter matrix A is replaced by the sentence-specific vectors $\alpha_s = (\alpha_{s1}, \ldots, \alpha_{sL})$.

Given the cluster assignments $\mathbf{z}$, we use the optimization toolbox in Matlab® to find the regression coefficients $\boldsymbol{\lambda}$ that maximize the joint distribution $\log p(\mathbf{z}, \boldsymbol{\lambda})$. This joint distribution is given by,

$$\log p(\mathbf{z}, \boldsymbol{\lambda}) = \log\Big(\prod_s p(z_s|\boldsymbol{\lambda})\Big) + \log\Big(\prod_l p(\lambda_l|\sigma^2)\Big) \tag{D.6}$$

where the first term is computed by marginalizing over θ_s

$$\begin{aligned}
\log \prod_s p(z_s|\boldsymbol{\lambda}) &= \log\Big(\prod_s \int_{\theta_s} p(z_s|\theta_s) \cdot p(\theta_s|\boldsymbol{\lambda})\, d\theta_s\Big) \\
&\propto \log \prod_s \frac{B(\alpha_{sl} + n_{sl})}{B(\alpha_{sl})} \\
&= \log \prod_s \frac{\Gamma(\sum_l \alpha_{sl})}{\Gamma(\sum_l \alpha_{sl} + n_{sl})} \prod_l \frac{\Gamma(\alpha_{sl} + n_{sl})}{\Gamma(\alpha_{sl})} \\
&= \log \prod_s \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_0 + \sum_l n_{sl})} \prod_l \frac{\Gamma(\alpha_{sl} + n_{sl})}{\Gamma(\alpha_{sl})} \\
&\propto \sum_{s=1}^{S} \sum_{l=1}^{L} \Big(\log \Gamma(\alpha_{sl} + n_{sl}) - \log \Gamma(\alpha_{sl})\Big)
\end{aligned}$$

The last line is obtained by noting that $\Gamma(\alpha_0)$ and $\Gamma(\alpha_0 + \sum_l n_{sl})$ are constants with respect to $\boldsymbol{\lambda}$. The second term in Equation D.6 is given by

$$\begin{aligned}\log \prod_l p(\lambda_l|\sigma^2) &= \sum_l \sum_k -\frac{1}{2}\log(2\pi\sigma^2) - \frac{\lambda_{lk}^2}{2\sigma^2} \\ &= -\sum_{l=1}^{L}\sum_{k=1}^{K}\frac{\lambda_{lk}^2}{2\sigma^2}\end{aligned}$$

The derivative of the log joint with respect to λ_{ij}, the regression coefficient for label i covariate j, is given by

$$\frac{\mathrm{d}\log p(\mathbf{z},\boldsymbol{\lambda})}{\mathrm{d}\lambda_{ij}} = \sum_{s=1}^{S}\sum_{l=1}^{L}\left(\frac{\mathrm{d}\log\Gamma(\alpha_{sl}+n_{sl})}{\mathrm{d}\lambda_{ij}} - \frac{\mathrm{d}\log\Gamma(\alpha_{sl})}{\mathrm{d}\lambda_{ij}}\right) - \frac{\lambda_{ij}}{\sigma^2}$$

Using the chain rule, we have

$$\frac{\mathrm{d}\log\Gamma(\alpha_{sl}+n_{sl})}{\mathrm{d}\lambda_{ij}} = \frac{\mathrm{d}\log\Gamma(\alpha_{sl}+n_{sl})}{\mathrm{d}\alpha_{si}} \cdot \frac{\mathrm{d}\alpha_{si}}{\mathrm{d}\pi_{si}} \cdot \frac{\mathrm{d}\pi_{si}}{\mathrm{d}\lambda_{ij}}$$

where

$$\frac{\mathrm{d}\log\Gamma(\alpha_{sl}+n_{sl})}{\mathrm{d}\alpha_{si}} = \begin{cases} \Psi(\alpha_{sl}+n_{sl}) & : l = i \\ 0 & : l \neq i \end{cases}$$

$$\frac{\mathrm{d}\alpha_{si}}{\mathrm{d}\pi_{si}} = \alpha_0\left(\frac{1}{\sum_j \pi_{sj}} - \frac{\pi_{si}^2}{(\sum_j \pi_{sj})^2}\right) \qquad \text{since } \alpha_{si} = \frac{\pi_{si}}{\sum_j \pi_{sj}}$$

$$\frac{\mathrm{d}\pi_{si}}{\mathrm{d}\lambda_{ij}} = \pi_{si} \cdot x_{sj} \qquad \text{since } \pi_{si} = \exp(x_s^\intercal \lambda_i)$$

and $\Psi(\cdot)$ is the digamma function. The derivative of $\log\Gamma(\alpha_{sl})$ is the same except $\Psi(\alpha_{sl}+n_{sl})$ is replaced with $\Psi(\alpha_{sl})$. Thus, the final derivative is

$$\frac{\mathrm{d}\log p(\mathbf{z},\boldsymbol{\lambda})}{\mathrm{d}\lambda_{ij}} = \sum_{s=1}^{S}\left[\alpha_0(\pi_{si}\cdot x_{sj})\left(\frac{1}{\sum_j \pi_{sj}} - \frac{\pi_{si}^2}{(\sum_j \pi_{sj})^2}\right)\Big(\Psi(\alpha_{si}+n_{si}) - \Psi(\alpha_{si})\Big)\right] - \frac{\lambda_{ij}}{\sigma^2}$$

The original DMR algorithm did not entail normalizing and scaling by the scalar α_0. Thus, our derivation of the joint distribution and the gradient of the joint distribution is different from the derivation presented in [44].

D.2 Experiments

In all of our experiments, we set σ^2 – the variance of the Normal prior over the regression coefficients – to 0.01 and experimented with $\alpha_0 \in \{50, 100\}$. Although DMR learns the hyper-parameters for the Dirichlet prior over θ_s, we still use an informative prior for the label distributions ϕ_l with hyper-parameters given by the hyper-parameter matrix B. Recall that the rows of B are normalized and scaled by the parameter $\beta \in \{50, 100, 500\}$.

We performed $5,000$ Gibbs sampling iterations where a single iteration consisted of sampling every random variable z_{si} from its conditional distribution. After every 10 Gibbs iterations, we updated the regression coefficients. After $5,000$ iterations, samples were taken every 100 iterations for a total of 10 samples from the posterior distribution.

A single sample from the posterior distribution consists of an assignment of every word in the training set to one of L clusters. Using these assignments, we computed point estimates for the label distributions ϕ_l given by Equation 3.11. We also compute the vector of hyper-parameters α_s for each sentence s in the test set using the regression coefficients learned from the training data:

$$\begin{aligned} \pi_{sl} &= \exp(x_s^\intercal \hat{\lambda}_l) \\ \alpha_s &= \alpha_0 \cdot \left[\frac{\pi_{s1}}{\sum_{j=1}^{L} \pi_{sj}} \cdots \frac{\pi_{sL}}{\sum_{j=1}^{L} \pi_{sj}} \right] \end{aligned} \tag{D.7}$$

where $\hat{\lambda}_l$ is the vector of regression coefficients for label l learned from the training data and x_s is the feature vector for sentence s in the test set.

Given these point estimates, we sample the latent assignment variables for the words in the test set. We perform $1,000$ such Gibbs iterations and use the assignment of words to labels in the last iteration for prediction. We report F_1 scores for the best configuration of DMR (i.e. the setting of β and α_0 that resulted in the highest macro-F_1 score).

E Illustrated examples of test articles

Figure E.3: The color scheme used to illustrate the predictions of sentence labels

In this section, we present three articles from the test set one from each corpus (chosen for brevity). For each article, we illustrate the labels determined by the human annotators along with the labels predicted by S-LDA-S, S-LDA-W, MC-LDA, and NB-inform. Figure E.3 shows the color-coding scheme used.

Introduction

The failure of antiretroviral therapies to completely suppress viral replication in some patients represents a major difficulty in the management of HIV infection. In therapy-naive patients without clinically apparent resistance mutations, triple-drug therapy with two nucleoside–analog reverse transcriptase inhibitors and a protease inhibitor or a non-nucleoside reverse transcriptase inhibitor is standard [1]. In these patients, treatment success rates, defined as viral load <50 copies/ml at 48 wk, range from 70% to 80%–85% (reviewed in [2]). However, in patients with previous regimen failure requiring salvage therapy, response rates are usually considerably lower [3–5], and it is frequently not possible to assemble a three-drug regimen with uncompromised activity against all viral strains present. In these individuals, treatment failure often occurs after an initial period of response to a new regimen, and is usually associated with the appearance of multiply drug-resistant viral strains. This has led to attempts to treat highly experienced patients with various deep salvage regimens consisting of four, five, or six individual drugs [6–11]. These patients are particularly vulnerable to the many drug interactions [12] (also reviewed in [13]) and adverse metabolic, hematologic, neurologic, cardiovascular, and gastrointestinal side effects that complicate HIV therapy and seriously undermine the success of clinical management [14–20] (also reviewed in [21]).

The need to minimize drug resistance while reducing treatment-related toxicities has engendered an interest in induction–maintenance (IM) strategies, in which a period of intensified antiretroviral therapy (induction phase) is followed by a simplified long-term regimen (maintenance phase) [22–25]. Most such trials have yielded higher failure rates in the treatment group than in controls receiving conventional therapy. Failure typically occurs during maintenance therapy, and has been attributed to poor regimen adherence [25] and recrudescence of resistance mutations present before institution of induction therapy [23]. One weakness of existing studies has been that induction therapy consisted of standard three-drug antiretroviral therapy (ART) regimens in common clinical use at the time of the study, under conditions now recognized to permit subclinical viral replication [26,27]. Moreover, in these early studies, the induction phase only lasted between 3 to 6 mo, which may be insufficient. However, two recent studies have shown the apparent effectiveness of induction therapy for 48 wk followed by maintenance therapy with atazanavir [28] or lopinvir/ritonavir [29,30], and this has led to new optimism concerning IM approaches.

We have hypothesized that a longer period of a highly suppressive induction therapy that is appropriately timed relative to the start of maintenance therapy may allow minority resistant variants to decay below a stochastic extinction threshold, allowing for successful long-term treatment with simpler and better-tolerated regimens. To explore this hypothesis quantitatively, we constructed a detailed computer simulation model of the dynamics of sensitive and resistant viruses during a hypothetical IM regimen. We show that the timing and duration of induction therapy relative to maintenance therapy can affect the probability that viruses resistant to the maintenance regimen will be eradicated in ways that are somewhat counterintuitive. Under biologically plausible conditions, we find that 6–10 mo of induction therapy are required to maximize the probability

Labels determined by human annotators (PLOS)

Introduction

The failure of antiretroviral therapies to completely suppress viral replication in some patients represents a major difficulty in the management of HIV infection. In therapy-naive patients without clinically apparent resistance mutations, triple-drug therapy with two nucleoside–analog reverse transcriptase inhibitors and a protease inhibitor or a non-nucleoside reverse transcriptase inhibitor is standard [1]. In these patients, treatment success rates, defined as viral load <50 copies/ml at 48 wk, range from 70% to 80%–85% (reviewed in [2]). However, in patients with previous regimen failure requiring salvage therapy, response rates are usually considerably lower [3–5], and it is frequently not possible to assemble a three-drug regimen with uncompromised activity against all viral strains present. In these individuals, treatment failure often occurs after an initial period of response to a new regimen, and is usually associated with the appearance of multiply drug-resistant viral strains. This has led to attempts to treat highly experienced patients with various deep salvage regimens consisting of four, five, or six individual drugs [6–11]. These patients are particularly vulnerable to the many drug interactions [12] (also reviewed in [13]) and adverse metabolic, hematologic, neurologic, cardiovascular, and gastrointestinal side effects that complicate HIV therapy and seriously undermine the success of clinical management [14–20] (also reviewed in [21]).

The need to minimize drug resistance while reducing treatment-related toxicities has engendered an interest in induction–maintenance (IM) strategies, in which a period of intensified antiretroviral therapy (induction phase) is followed by a simplified long-term regimen (maintenance phase) [22–25]. Most such trials have yielded higher failure rates in the treatment group than in controls receiving conventional therapy. Failure typically occurs during maintenance therapy, and has been attributed to poor regimen adherence [25] and recrudescence of resistance mutations present before institution of induction therapy [23]. One weakness of existing studies has been that induction therapy consisted of standard three-drug antiretroviral therapy (ART) regimens in common clinical use at the time of the study, under conditions now recognized to permit subclinical viral replication [26,27]. Moreover, in these early studies, the induction phase only lasted between 3 to 6 mo, which may be insufficient. However, two recent studies have shown the apparent effectiveness of induction therapy for 48 wk followed by maintenance therapy with atazanavir [28] or lopinvir/ritonavir [29,30], and this has led to new optimism concerning IM approaches.

We have hypothesized that a longer period of a highly suppressive induction therapy that is appropriately timed relative to the start of maintenance therapy may allow minority resistant variants to decay below a stochastic extinction threshold, allowing for successful long-term treatment with simpler and better-tolerated regimens. To explore this hypothesis quantitatively, we constructed a detailed computer simulation model of the dynamics of sensitive and resistant viruses during a hypothetical IM regimen. We show that the timing and duration of induction therapy relative to maintenance therapy can affect the probability that viruses resistant to the maintenance regimen will be eradicated in ways that are somewhat counterintuitive. Under biologically plausible conditions, we find that 6–10 mo of induction therapy are required to maximize the probability

Labels predicted by Sent-LDA-S

Introduction

The failure of antiretroviral therapies to completely suppress viral replication in some patients represents a major difficulty in the management of HIV infection. In therapy-naive patients without clinically apparent resistance mutations, triple-drug therapy with two nucleoside–analog reverse transcriptase inhibitors and a protease inhibitor or a non-nucleoside reverse transcriptase inhibitor is standard [1]. In these patients, treatment success rates, defined as viral load <50 copies/ml at 48 wk, range from 70% to 80%–85% (reviewed in [2]). However, in patients with previous regimen failure requiring salvage therapy, response rates are usually considerably lower [3–5], and it is frequently not possible to assemble a three-drug regimen with uncompromised activity against all viral strains present. In these individuals, treatment failure often occurs after an initial period of response to a new regimen, and is usually associated with the appearance of multiply drug-resistant viral strains. This has led to attempts to treat highly experienced patients with various deep salvage regimens consisting of four, five, or six individual drugs [6–11]. These patients are particularly vulnerable to the many drug interactions [12] (also reviewed in [13]) and adverse metabolic, hematologic, neurologic, cardiovascular, and gastrointestinal side effects that complicate HIV therapy and seriously undermine the success of clinical management [14–20] (also reviewed in [21]).

The need to minimize drug resistance while reducing treatment-related toxicities has engendered an interest in induction–maintenance (IM) strategies, in which a period of intensified antiretroviral therapy (induction phase) is followed by a simplified long-term regimen (maintenance phase) [22–25]. Most such trials have yielded higher failure rates in the treatment group than in controls receiving conventional therapy. Failure typically occurs during maintenance therapy, and has been attributed to poor regimen adherence [25] and recrudescence of resistance mutations present before institution of induction therapy [23]. One weakness of existing studies has been that induction therapy consisted of standard three-drug antiretroviral therapy (ART) regimens in common clinical use at the time of the study, under conditions now recognized to permit subclinical viral replication [26,27]. Moreover, in these early studies, the induction phase only lasted between 3 to 6 mo, which may be insufficient. However, two recent studies have shown the apparent effectiveness of induction therapy for 48 wk followed by maintenance therapy with atazanavir [28] or lopinvir/ritonavir [29,30], and this has led to new optimism concerning IM approaches.

We have hypothesized that a longer period of a highly suppressive induction therapy that is appropriately timed relative to the start of maintenance therapy may allow minority resistant variants to decay below a stochastic extinction threshold, allowing for successful long-term treatment with simpler and better-tolerated regimens. To explore this hypothesis quantitatively, we constructed a detailed computer simulation model of the dynamics of sensitive and resistant viruses during a hypothetical IM regimen. We show that the timing and duration of induction therapy relative to maintenance therapy can affect the probability that viruses resistant to the maintenance regimen will be eradicated in ways that are somewhat counterintuitive. Under biologically plausible conditions, we find that 6–10 mo of induction therapy are required to maximize the probability

Labels predicted by Sent-LDA-W

Introduction

The failure of antiretroviral therapies to completely suppress viral replication in some patients represents a major difficulty in the management of HIV infection. In therapy-naive patients without clinically apparent resistance mutations, triple-drug therapy with two nucleoside–analog reverse transcriptase inhibitors and a protease inhibitor or a non-nucleoside reverse transcriptase inhibitor is standard [1]. In these patients, treatment success rates, defined as viral load <50 copies/ml at 48 wk, range from 70% to 80%–85% (reviewed in [2]). However, in patients with previous regimen failure requiring salvage therapy, response rates are usually considerably lower [3–5], and it is frequently not possible to assemble a three-drug regimen with uncompromised activity against all viral strains present. In these individuals, treatment failure often occurs after an initial period of response to a new regimen, and is usually associated with the appearance of multiply drug-resistant viral strains. This has led to attempts to treat highly experienced patients with various deep salvage regimens consisting of four, five, or six individual drugs [6–11]. These patients are particularly vulnerable to the many drug interactions [12] (also reviewed in [13]) and adverse metabolic, hematologic, neurologic, cardiovascular, and gastrointestinal side effects that complicate HIV therapy and seriously undermine the success of clinical management [14–20] (also reviewed in [21]).

The need to minimize drug resistance while reducing treatment-related toxicities has engendered an interest in induction–maintenance (IM) strategies, in which a period of intensified antiretroviral therapy (induction phase) is followed by a simplified long-term regimen (maintenance phase) [22–25]. Most such trials have yielded higher failure rates in the treatment group than in controls receiving conventional therapy. Failure typically occurs during maintenance therapy, and has been attributed to poor regimen adherence [25] and recrudescence of resistance mutations present before institution of induction therapy [23]. One weakness of existing studies has been that induction therapy consisted of standard three-drug antiretroviral therapy (ART) regimens in common clinical use at the time of the study, under conditions now recognized to permit subclinical viral replication [26,27]. Moreover, in these early studies, the induction phase only lasted between 3 to 6 mo, which may be insufficient. However, two recent studies have shown the apparent effectiveness of induction therapy for 48 wk followed by maintenance therapy with atazanavir [28] or lopinvir/ritonavir [29,30], and this has led to new optimism concerning IM approaches.

We have hypothesized that a longer period of a highly suppressive induction therapy that is appropriately timed relative to the start of maintenance therapy may allow minority resistant variants to decay below a stochastic extinction threshold, allowing for successful long-term treatment with simpler and better-tolerated regimens. To explore this hypothesis quantitatively, we constructed a detailed computer simulation model of the dynamics of sensitive and resistant viruses during a hypothetical IM regimen. We show that the timing and duration of induction therapy relative to maintenance therapy can affect the probability that viruses resistant to the maintenance regimen will be eradicated in ways that are somewhat counterintuitive. Under biologically plausible conditions, we find that 6–10 mo of induction therapy are required to maximize the probability

Labels predicted by MC-LDA

Introduction

The failure of antiretroviral therapies to completely suppress viral replication in some patients represents a major difficulty in the management of HIV infection. In therapy-naive patients without clinically apparent resistance mutations, triple-drug therapy with two nucleoside–analog reverse transcriptase inhibitors and a protease inhibitor or a non-nucleoside reverse transcriptase inhibitor is standard [1]. In these patients, treatment success rates, defined as viral load <50 copies/ml at 48 wk, range from 70% to 80%–85% (reviewed in [2]). However, in patients with previous regimen failure requiring salvage therapy, response rates are usually considerably lower [3–5], and it is frequently not possible to assemble a three-drug regimen with uncompromised activity against all viral strains present. In these individuals, treatment failure often occurs after an initial period of response to a new regimen, and is usually associated with the appearance of multiply drug-resistant viral strains. This has led to attempts to treat highly experienced patients with various deep salvage regimens consisting of four, five, or six individual drugs [6–11]. These patients are particularly vulnerable to the many drug interactions [12] (also reviewed in [13]) and adverse metabolic, hematologic, neurologic, cardiovascular, and gastrointestinal side effects that complicate HIV therapy and seriously undermine the success of clinical management [14–20] (also reviewed in [21]).

The need to minimize drug resistance while reducing treatment-related toxicities has engendered an interest in induction–maintenance (IM) strategies, in which a period of intensified antiretroviral therapy (induction phase) is followed by a simplified long-term regimen (maintenance phase) [22–25]. Most such trials have yielded higher failure rates in the treatment group than in controls receiving conventional therapy. Failure typically occurs during maintenance therapy, and has been attributed to poor regimen adherence [25] and recrudescence of resistance mutations present before institution of induction therapy [23]. One weakness of existing studies has been that induction therapy consisted of standard three-drug antiretroviral therapy (ART) regimens in common clinical use at the time of the study, under conditions now recognized to permit subclinical viral replication [26,27]. Moreover, in these early studies, the induction phase only lasted between 3 to 6 mo, which may be insufficient. However, two recent studies have shown the apparent effectiveness of induction therapy for 48 wk followed by maintenance therapy with atazanavir [28] or lopinvir/ritonavir [29,30], and this has led to new optimism concerning IM approaches.

We have hypothesized that a longer period of a highly suppressive induction therapy that is appropriately timed relative to the start of maintenance therapy may allow minority resistant variants to decay below a stochastic extinction threshold, allowing for successful long-term treatment with simpler and better-tolerated regimens. To explore this hypothesis quantitatively, we constructed a detailed computer simulation model of the dynamics of sensitive and resistant viruses during a hypothetical IM regimen. We show that the timing and duration of induction therapy relative to maintenance therapy can affect the probability that viruses resistant to the maintenance regimen will be eradicated in ways that are somewhat counterintuitive. Under biologically plausible conditions, we find that 6–10 mo of induction therapy are required to maximize the probability

Labels predicted by NB-inform

Abstract

The paper extends research on fixed-pie perceptions by suggesting that disputants may prefer proposals that are perceived to be equally attractive to both parties (i.e., balanced) rather than one-sided, because balanced agreements are seen as more likely to be successfully implemented. We test our predictions using data on Israeli support for the Geneva Accords, an agreement for a two state solution negotiated by unofficial delegations of Israel and the Palestinian Authority in 2003. The results demonstrate that Israelis are more likely to support agreements that are seen favorably by other Israelis, but — contrary to fixed-pie predictions — Israeli support for the accords does not diminish simply because a majority of Palestinians favors (rather than opposes) the accords. We show that implementation concerns create a demand among Israelis for balance in the degree to which each side favors (or opposes) the agreement. The effect of balance is noteworthy in that it creates considerable support for proposals even when a majority of Israelis and Palestinians *oppose* the deal.

Keywords: Israel, Palestine, negotiation, fixed pie, balance, peace.

1 Introduction

> "I have had a philosophy for some time in regard to SALT, and it goes like this: the Russians will not accept a SALT treaty that is not in their best interests, and it seems to me that if it is in their best interests, it cannot be in our best interest." U.S. Congressman Floyd Spence.[1]

Normative models of bargaining and negotiation suggest that if there is potential for mutual benefit, conflicting parties should be able to achieve it (Raiffa, 1982). Descriptive accounts and empirical investigations of negotiation behavior (e.g., Bazerman & Neale, 1983; Thompson, 1990; Thompson & Hrebec, 1996; Walton & McKersie, 1965), however, suggest that a number of psychological barriers to conflict resolution are likely to make efficient deal making difficult (Bazerman & Neale, 1990; Ross and Stillinger, 1991; Thompson & Hastie, 1990). For example, research on cognitive biases associated with egocentric perceptions suggests that negotiators and evaluators of negotiated agreements are likely to exhibit a "fixed-pie bias" (Bazerman, 1986; Bazerman & Neale, 1983; Schelling, 1960). The fixed-pie bias refers to the belief that any gain for one party will be associated with an equivalent loss to the other party. This belief is a "bias" when it persists even in contexts where there is a possibility of compatible interests or mutual benefit. A large body of research finds that negotiators are susceptible to the fixed-pie bias prior to, during, and even after negotiations (de Dreu, Koole, & Steinel, 2000; Pinkley, Griffith, & Northcraft, 1995; Thompson & Hastie, 1990; Thompson & Hrebec, 1996). In the current paper we investigate and extend research on fixed pie bias in the context of protracted intergroup conflict.

Labels determined by human annotators (JDM)

Abstract

The paper extends research on fixed-pie perceptions by suggesting that disputants may prefer proposals that are perceived to be equally attractive to both parties (i.e., balanced) rather than one-sided, because balanced agreements are seen as more likely to be successfully implemented. We test our predictions using data on Israeli support for the Geneva Accords, an agreement for a two state solution negotiated by unofficial delegations of Israel and the Palestinian Authority in 2003. The results demonstrate that Israelis are more likely to support agreements that are seen favorably by other Israelis, but — contrary to fixed-pie predictions — Israeli support for the accords does not diminish simply because a majority of Palestinians favors (rather than opposes) the accords. We show that implementation concerns create a demand among Israelis for balance in the degree to which each side favors (or opposes) the agreement. The effect of balance is noteworthy in that it creates considerable support for proposals even when a majority of Israelis and Palestinians *oppose* the deal.

Keywords: Israel, Palestine, negotiation, fixed pie, balance, peace.

1 Introduction

> "I have had a philosophy for some time in regard to SALT, and it goes like this: the Russians will not accept a SALT treaty that is not in their best interests, and it seems to me that if it is in their best interests, it cannot be in our best interest." U.S. Congressman Floyd Spence.[1]

Normative models of bargaining and negotiation suggest that if there is potential for mutual benefit, conflicting parties should be able to achieve it (Raiffa, 1982). Descriptive accounts and empirical investigations of negotiation behavior (e.g., Bazerman & Neale, 1983; Thompson, 1990; Thompson & Hrebec, 1996; Walton & McKersie, 1965), however, suggest that a number of psychological barriers to conflict resolution are likely to make efficient deal making difficult (Bazerman & Neale, 1990; Ross and Stillinger, 1991; Thompson & Hastie, 1990). For example, research on cognitive biases associated with egocentric perceptions suggests that negotiators and evaluators of negotiated agreements are likely to exhibit a "fixed-pie bias" (Bazerman, 1986; Bazerman & Neale, 1983; Schelling, 1960). The fixed-pie bias refers to the belief that any gain for one party will be associated with an equivalent loss to the other party. This belief is a "bias" when it persists even in contexts where there is a possibility of compatible interests or mutual benefit. A large body of research finds that negotiators are susceptible to the fixed-pie bias prior to, during, and even after negotiations (de Dreu, Koole, & Steinel, 2000; Pinkley, Griffith, & Northcraft, 1995; Thompson & Hastie, 1990; Thompson & Hrebec, 1996). In the current paper we investigate and extend research on fixed pie bias in the context of protracted intergroup conflict

Labels predicted by Sent-LDA-S

Abstract

The paper extends research on fixed-pie perceptions by suggesting that disputants may prefer proposals that are perceived to be equally attractive to both parties (i.e., balanced) rather than one-sided, because balanced agreements are seen as more likely to be successfully implemented. We test our predictions using data on Israeli support for the Geneva Accords, an agreement for a two state solution negotiated by unofficial delegations of Israel and the Palestinian Authority in 2003. The results demonstrate that Israelis are more likely to support agreements that are seen favorably by other Israelis, but — contrary to fixed-pie predictions — Israeli support for the accords does not diminish simply because a majority of Palestinians favors (rather than opposes) the accords. We show that implementation concerns create a demand among Israelis for balance in the degree to which each side favors (or opposes) the agreement. The effect of balance is noteworthy in that it creates considerable support for proposals even when a majority of Israelis and Palestinians *oppose* the deal.

Keywords: Israel, Palestine, negotiation, fixed pie, balance, peace.

1 Introduction

> "I have had a philosophy for some time in regard to SALT, and it goes like this: the Russians will not accept a SALT treaty that is not in their best interests, and it seems to me that if it is in their best interests, it cannot be in our best interest." U.S. Congressman Floyd Spence.[1]

Normative models of bargaining and negotiation suggest that if there is potential for mutual benefit, conflicting parties should be able to achieve it (Raiffa, 1982). Descriptive accounts and empirical investigations of negotiation behavior (e.g., Bazerman & Neale, 1983; Thompson, 1990; Thompson & Hrebec, 1996; Walton & McKersie, 1965), however, suggest that a number of psychological barriers to conflict resolution are likely to make efficient deal making difficult (Bazerman & Neale, 1990; Ross and Stillinger, 1991; Thompson & Hastie, 1990). For example, research on cognitive biases associated with egocentric perceptions suggests that negotiators and evaluators of negotiated agreements are likely to exhibit a "fixed-pie bias" (Bazerman, 1986; Bazerman & Neale, 1983; Schelling, 1960). The fixed-pie bias refers to the belief that any gain for one party will be associated with an equivalent loss to the other party. This belief is a "bias" when it persists even in contexts where there is a possibility of compatible interests or mutual benefit. A large body of research finds that negotiators are susceptible to the fixed-pie bias prior to, during, and even after negotiations (de Dreu, Koole, & Steinel, 2000; Pinkley, Griffith, & Northcraft, 1995; Thompson & Hastie, 1990; Thompson & Hrebec, 1996). In the current paper we investigate and extend research on fixed pie bias in the context of protracted intergroup conflict.

Labels predicted by Sent-LDA-W

Abstract

The paper extends research on fixed-pie perceptions by suggesting that disputants may prefer proposals that are perceived to be equally attractive to both parties (i.e., balanced) rather than one-sided, because balanced agreements are seen as more likely to be successfully implemented. We test our predictions using data on Israeli support for the Geneva Accords, an agreement for a two state solution negotiated by unofficial delegations of Israel and the Palestinian Authority in 2003. The results demonstrate that Israelis are more likely to support agreements that are seen favorably by other Israelis, but — contrary to fixed-pie predictions — Israeli support for the accords does not diminish simply because a majority of Palestinians favors (rather than opposes) the accords. We show that implementation concerns create a demand among Israelis for balance in the degree to which each side favors (or opposes) the agreement. The effect of balance is noteworthy in that it creates considerable support for proposals even when a majority of Israelis and Palestinians *oppose* the deal.

Keywords: Israel, Palestine, negotiation, fixed pie, balance, peace.

1 Introduction

> "I have had a philosophy for some time in regard to SALT, and it goes like this: the Russians will not accept a SALT treaty that is not in their best interests, and it seems to me that if it is in their best interests, it cannot be in our best interest." U.S. Congressman Floyd Spence.[1]

Normative models of bargaining and negotiation suggest that if there is potential for mutual benefit, conflicting parties should be able to achieve it (Raiffa, 1982). Descriptive accounts and empirical investigations of negotiation behavior (e.g., Bazerman & Neale, 1983; Thompson, 1990; Thompson & Hrebec, 1996; Walton & McKersie, 1965), however, suggest that a number of psychological barriers to conflict resolution are likely to make efficient deal making difficult (Bazerman & Neale, 1990; Ross and Stillinger, 1991; Thompson & Hastie, 1990). For example, research on cognitive biases associated with egocentric perceptions suggests that negotiators and evaluators of negotiated agreements are likely to exhibit a "fixed-pie bias" (Bazerman, 1986; Bazerman & Neale, 1983; Schelling, 1960). The fixed-pie bias refers to the belief that any gain for one party will be associated with an equivalent loss to the other party. This belief is a "bias" when it persists even in contexts where there is a possibility of compatible interests or mutual benefit. A large body of research finds that negotiators are susceptible to the fixed-pie bias prior to, during, and even after negotiations (de Dreu, Koole, & Steinel, 2000; Pinkley, Griffith, & Northcraft, 1995; Thompson & Hastie, 1990; Thompson & Hrebec, 1996). In the current paper we investigate and extend research on fixed pie bias in the context of protracted intergroup conflict

Labels predicted by MC-LDA

Abstract

The paper extends research on fixed-pie perceptions by suggesting that disputants may prefer proposals that are perceived to be equally attractive to both parties (i.e., balanced) rather than one-sided, because balanced agreements are seen as more likely to be successfully implemented. We test our predictions using data on Israeli support for the Geneva Accords, an agreement for a two state solution negotiated by unofficial delegations of Israel and the Palestinian Authority in 2003. The results demonstrate that Israelis are more likely to support agreements that are seen favorably by other Israelis, but — contrary to fixed-pie predictions — Israeli support for the accords does not diminish simply because a majority of Palestinians favors (rather than opposes) the accords. We show that implementation concerns create a demand among Israelis for balance in the degree to which each side favors (or opposes) the agreement. The effect of balance is noteworthy in that it creates considerable support for proposals even when a majority of Israelis and Palestinians *oppose* the deal.

Keywords: Israel, Palestine, negotiation, fixed pie, balance, peace.

1 Introduction

> "I have had a philosophy for some time in regard to SALT, and it goes like this: the Russians will not accept a SALT treaty that is not in their best interests, and it seems to me that if it is in their best interests, it cannot be in our best interest." U.S. Congressman Floyd Spence.[1]

Normative models of bargaining and negotiation suggest that if there is potential for mutual benefit, conflicting parties should be able to achieve it (Raiffa, 1982). Descriptive accounts and empirical investigations of negotiation behavior (e.g., Bazerman & Neale, 1983; Thompson, 1990; Thompson & Hrebec, 1996; Walton & McKersie, 1965), however, suggest that a number of psychological barriers to conflict resolution are likely to make efficient deal making difficult (Bazerman & Neale, 1990; Ross and Stillinger, 1991; Thompson & Hastie, 1990). For example, research on cognitive biases associated with egocentric perceptions suggests that negotiators and evaluators of negotiated agreements are likely to exhibit a "fixed-pie bias" (Bazerman, 1986; Bazerman & Neale, 1983; Schelling, 1960). The fixed-pie bias refers to the belief that any gain for one party will be associated with an equivalent loss to the other party. This belief is a "bias" when it persists even in contexts where there is a possibility of compatible interests or mutual benefit. A large body of research finds that negotiators are susceptible to the fixed-pie bias prior to, during, and even after negotiations (de Dreu, Koole, & Steinel, 2000; Pinkley, Griffith, & Northcraft, 1995; Thompson & Hastie, 1990; Thompson & Hrebec, 1996). In the current paper we investigate and extend research on fixed pie bias in the context of protracted intergroup conflict.

Labels predicted by NB-inform

1 Introduction

The rapid expansion of the Internet in the last two decades has produced a large-scale system of thousands of diverse, independently managed networks that collectively provide global connectivity across a wide spectrum of geopolitical environments. From 1997 to 2005 the number of globally routable AS identifiers has increased from less than 2,000 to more than 20,000, exerting significant pressure on interdomain routing as well as other functional and structural parts of the Internet. This impressive growth has resulted in a heterogenous and highly complex system that challenges accurate and realistic modeling of the Internet infrastructure. In particular, the AS-level topology is an intermix of networks owned and operated by many different organizations, e.g., backbone providers, regional providers, access providers, universities and private companies. Statistical information that faithfully characterizes different AS types is on the critical path toward understanding the structure of the Internet, as well as for modeling its topology and growth.

In topology modeling, knowledge of AS types is mandatory for augmenting synthetically constructed or measured AS topologies with realistic intra-AS and inter-AS router-level topologies. For example, we expect the network of a dual-homed university to be drastically different from that of a dual-homed small company. The university will likely contain dozens of internal routers, thousands of hosts, and many other network elements (switches, servers, firewalls). On the other hand, the small company will most probably have a single router and a simple network topology. Since there is such a diversity among different network types, we cannot accurately augment the AS-level topology with appropriate router-level topologies if we cannot characterize the composing ASes.

Moreover, annotating the ASes in the AS topology with their types is a prerequisite for modeling the evolution of the Internet, since different types of ASes exhibit different growth patterns. For example, Internet Service Providers (ISP) grow by attracting new customers and by engaging in business agreements with other ISPs. On the other hand, small companies that connect to the Internet through one or few ISPs do not grow significantly over time. Thus, categorizing different types of ASes in the Internet is necessary to identify network evolution patterns and develop accurate evolution models.

An AS taxonomy is also necessary for mapping IP addresses to different types of users. For example, in traffic analysis studies its often required to distinguish between packets that come from home and business users. Given an AS taxonomy, its possible to realize this goal by checking the type of AS that originates the prefix in which an IP address lies.

In this work, we introduce a radically new approach based on machine learning to construct a representative AS taxonomy. We develop an algorithm to classify ASes based on empirically observed differences between AS characteristics. We use a large set of data from the Internet Routing Registries (IRR) [12] and from RouteViews [9] to identify intrinsic differences between ASes of different types. Then, we employ a novel machine learning technique to build a classification algorithm that exploits these differences to classify ASes into six representative classes that reflect ASes with different network properties and infrastructures. We derive macroscopic statistics on the different types of ASes in the Internet and validate our results using a sample of 1200 manually identified AS types. Our validation demonstrates that our classification algorithm achieves high accuracy: 78.1% of the examined classifications were correct. Finally, we make our results and our classifier publicly available to promote further research and understanding of the Internet's structure and evolution.

In Section 2 we start with a brief discussion of related work. Section 3 describes the data we used, and in Section 4 we specify the set of AS classes we use in our experiments. Section 5 introduces our classification approach and results. We validate them in Section 6 and conclude in Section 7.

Labels determined by human annotators (ARXIV)

1 Introduction

The rapid expansion of the Internet in the last two decades has produced a large-scale system of thousands of diverse, independently managed networks that collectively provide global connectivity across a wide spectrum of geopolitical environments. From 1997 to 2005 the number of globally routable AS identifiers has increased from less than 2,000 to more than 20,000, exerting significant pressure on interdomain routing as well as other functional and structural parts of the Internet. This impressive growth has resulted in a heterogenous and highly complex system that challenges accurate and realistic modeling of the Internet infrastructure. In particular, the AS-level topology is an intermix of networks owned and operated by many different organizations, e.g., backbone providers, regional providers, access providers, universities and private companies. Statistical information that faithfully characterizes different AS types is on the critical path toward understanding the structure of the Internet, as well as for modeling its topology and growth.

In topology modeling, knowledge of AS types is mandatory for augmenting synthetically constructed or measured AS topologies with realistic intra-AS and inter-AS router-level topologies. For example, we expect the network of a dual-homed university to be drastically different from that of a dual-homed small company. The university will likely contain dozens of internal routers, thousands of hosts, and many other network elements (switches, servers, firewalls). On the other hand, the small company will most probably have a single router and a simple network topology. Since there is such a diversity among different network types, we cannot accurately augment the AS-level topology with appropriate router-level topologies if we cannot characterize the composing ASes.

Moreover, annotating the ASes in the AS topology with their types is a prerequisite for modeling the evolution of the Internet, since different types of ASes exhibit different growth patterns. For example, Internet Service Providers (ISP) grow by attracting new customers and by engaging in business agreements with other ISPs. On the other hand, small companies that connect to the Internet through one or few ISPs do not grow significantly over time. Thus, categorizing different types of ASes in the Internet is necessary to identify network evolution patterns and develop accurate evolution models.

An AS taxonomy is also necessary for mapping IP addresses to different types of users. For example, in traffic analysis studies its often required to distinguish between packets that come from home and business users. Given an AS taxonomy, its possible to realize this goal by checking the type of AS that originates the prefix in which an IP address lies.

In this work, we introduce a radically new approach based on machine learning to construct a representative AS taxonomy. We develop an algorithm to classify ASes based on empirically observed differences between AS characteristics. We use a large set of data from the Internet Routing Registries (IRR) [12] and from RouteViews [9] to identify intrinsic differences between ASes of different types. Then, we employ a novel machine learning technique to build a classification algorithm that exploits these differences to classify ASes into six representative classes that reflect ASes with different network properties and infrastructures. We derive macroscopic statistics on the different types of ASes in the Internet and validate our results using a sample of 1200 manually identified AS types. Our validation demonstrates that our classification algorithm achieves high accuracy: 78.1% of the examined classifications were correct. Finally, we make our results and our classifier publicly available to promote further research and understanding of the Internet's structure and evolution.

In Section 2 we start with a brief discussion of related work. Section 3 describes the data we used, and in Section 4 we specify the set of AS classes we use in our experiments. Section 5 introduces our classification approach and results. We validate them in Section 6 and conclude in Section 7.

Labels predicted by Sent-LDA-S

1 Introduction

The rapid expansion of the Internet in the last two decades has produced a large-scale system of thousands of diverse, independently managed networks that collectively provide global connectivity across a wide spectrum of geopolitical environments. From 1997 to 2005 the number of globally routable AS identifiers has increased from less than 2,000 to more than 20,000, exerting significant pressure on interdomain routing as well as other functional and structural parts of the Internet. This impressive growth has resulted in a heterogenous and highly complex system that challenges accurate and realistic modeling of the Internet infrastructure. In particular, the AS-level topology is an intermix of networks owned and operated by many different organizations, e.g., backbone providers, regional providers, access providers, universities and private companies. Statistical information that faithfully characterizes different AS types is on the critical path toward understanding the structure of the Internet, as well as for modeling its topology and growth.

In topology modeling, knowledge of AS types is mandatory for augmenting synthetically constructed or measured AS topologies with realistic intra-AS and inter-AS router-level topologies. For example, we expect the network of a dual-homed university to be drastically different from that of a dual-homed small company. The university will likely contain dozens of internal routers, thousands of hosts, and many other network elements (switches, servers, firewalls). On the other hand, the small company will most probably have a single router and a simple network topology. Since there is such a diversity among different network types, we cannot accurately augment the AS-level topology with appropriate router-level topologies if we cannot characterize the composing ASes.

Moreover, annotating the ASes in the AS topology with their types is a prerequisite for modeling the evolution of the Internet, since different types of ASes exhibit different growth patterns. For example, Internet Service Providers (ISP) grow by attracting new customers and by engaging in business agreements with other ISPs. On the other hand, small companies that connect to the Internet through one or few ISPs do not grow significantly over time. Thus, categorizing different types of ASes in the Internet is necessary to identify network evolution patterns and develop accurate evolution models.

An AS taxonomy is also necessary for mapping IP addresses to different types of users. For example, in traffic analysis studies its often required to distinguish between packets that come from home and business users. Given an AS taxonomy, its possible to realize this goal by checking the type of AS that originates the prefix in which an IP address lies.

In this work, we introduce a radically new approach based on machine learning to construct a representative AS taxonomy. We develop an algorithm to classify ASes based on empirically observed differences between AS characteristics. We use a large set of data from the Internet Routing Registries (IRR) [12] and from Route-Views [9] to identify intrinsic differences between ASes of different types. Then, we employ a novel machine learning technique to build a classification algorithm that exploits these differences to classify ASes into six representative classes that reflect ASes with different network properties and infrastructures. We derive macroscopic statistics on the different types of ASes in the Internet and validate our results using a sample of 1200 manually identified AS types. Our validation demonstrates that our classification algorithm achieves high accuracy: 78.1% of the examined classifications were correct. Finally, we make our results and our classifier publicly available to promote further research and understanding of the Internet's structure and evolution.

In Section 2 we start with a brief discussion of related work. Section 3 describes the data we used, and in Section 4 we specify the set of AS classes we use in our experiments. Section 5 introduces our classification approach and results. We validate them in Section 6 and conclude in Section 7.

Labels predicted by Sent-LDA-W

1 Introduction

The rapid expansion of the Internet in the last two decades has produced a large-scale system of thousands of diverse, independently managed networks that collectively provide global connectivity across a wide spectrum of geopolitical environments. From 1997 to 2005 the number of globally routable AS identifiers has increased from less than 2,000 to more than 20,000, exerting significant pressure on interdomain routing as well as other functional and structural parts of the Internet. This impressive growth has resulted in a heterogenous and highly complex system that challenges accurate and realistic modeling of the Internet infrastructure. In particular, the AS-level topology is an intermix of networks owned and operated by many different organizations, e.g., backbone providers, regional providers, access providers, universities and private companies. Statistical information that faithfully characterizes different AS types is on the critical path toward understanding the structure of the Internet, as well as for modeling its topology and growth.

In topology modeling, knowledge of AS types is mandatory for augmenting synthetically constructed or measured AS topologies with realistic intra-AS and inter-AS router-level topologies. For example, we expect the network of a dual-homed university to be drastically different from that of a dual-homed small company. The university will likely contain dozens of internal routers, thousands of hosts, and many other network elements (switches, servers, firewalls). On the other hand, the small company will most probably have a single router and a simple network topology. Since there is such a diversity among different network types, we cannot accurately augment the AS-level topology with appropriate router-level topologies if we cannot characterize the composing ASes.

Moreover, annotating the ASes in the AS topology with their types is a prerequisite for modeling the evolution of the Internet, since different types of ASes exhibit different growth patterns. For example, Internet Service Providers (ISP) grow by attracting new customers and by engaging in business agreements with other ISPs. On the other hand, small companies that connect to the Internet through one or few ISPs do not grow significantly over time. Thus, categorizing different types of ASes in the Internet is necessary to identify network evolution patterns and develop accurate evolution models.

An AS taxonomy is also necessary for mapping IP addresses to different types of users. For example, in traffic analysis studies its often required to distinguish between packets that come from home and business users. Given an AS taxonomy, its possible to realize this goal by checking the type of AS that originates the prefix in which an IP address lies.

In this work, we introduce a radically new approach based on machine learning to construct a representative AS taxonomy. We develop an algorithm to classify ASes based on empirically observed differences between AS characteristics. We use a large set of data from the Internet Routing Registries (IRR) [12] and from RouteViews [9] to identify intrinsic differences between ASes of different types. Then, we employ a novel machine learning technique to build a classification algorithm that exploits these differences to classify ASes into six representative classes that reflect ASes with different network properties and infrastructures. We derive macroscopic statistics on the different types of ASes in the Internet and validate our results using a sample of 1200 manually identified AS types. Our validation demonstrates that our classification algorithm achieves high accuracy: 78.1% of the examined classifications were correct. Finally, we make our results and our classifier publicly available to promote further research and understanding of the Internet's structure and evolution.

In Section 2 we start with a brief discussion of related work. Section 3 describes the data we used, and in Section 4 we specify the set of AS classes we use in our experiments. Section 5 introduces our classification approach and results. We validate them in Section 6 and conclude in Section 7.

Labels predicted by MC-LDA

1 Introduction

The rapid expansion of the Internet in the last two decades has produced a large-scale system of thousands of diverse, independently managed networks that collectively provide global connectivity across a wide spectrum of geopolitical environments. From 1997 to 2005 the number of globally routable AS identifiers has increased from less than 2,000 to more than 20,000, exerting significant pressure on interdomain routing as well as other functional and structural parts of the Internet. This impressive growth has resulted in a heterogenous and highly complex system that challenges accurate and realistic modeling of the Internet infrastructure. In particular, the AS-level topology is an intermix of networks owned and operated by many different organizations, e.g., backbone providers, regional providers, access providers, universities and private companies. Statistical information that faithfully characterizes different AS types is on the critical path toward understanding the structure of the Internet, as well as for modeling its topology and growth.

In topology modeling, knowledge of AS types is mandatory for augmenting synthetically constructed or measured AS topologies with realistic intra-AS and inter-AS router-level topologies. For example, we expect the network of a dual-homed university to be drastically different from that of a dual-homed small company. The university will likely contain dozens of internal routers, thousands of hosts, and many other network elements (switches, servers, firewalls). On the other hand, the small company will most probably have a single router and a simple network topology. Since there is such a diversity among different network types, we cannot accurately augment the AS-level topology with appropriate router-level topologies if we cannot characterize the composing ASes.

Moreover, annotating the ASes in the AS topology with their types is a prerequisite for modeling the evolution of the Internet, since different types of ASes exhibit different growth patterns. For example, Internet Service Providers (ISP) grow by attracting new customers and by engaging in business agreements with other ISPs. On the other hand, small companies that connect to the Internet through one or few ISPs do not grow significantly over time. Thus, categorizing different types of ASes in the Internet is necessary to identify network evolution patterns and develop accurate evolution models.

An AS taxonomy is also necessary for mapping IP addresses to different types of users. For example, in traffic analysis studies its often required to distinguish between packets that come from home and business users. Given an AS taxonomy, its possible to realize this goal by checking the type of AS that originates the prefix in which an IP address lies.

In this work, we introduce a radically new approach based on machine learning to construct a representative AS taxonomy. We develop an algorithm to classify ASes based on empirically observed differences between AS characteristics. We use a large set of data from the Internet Routing Registries (IRR) [12] and from RouteViews [9] to identify intrinsic differences between ASes of different types. Then, we employ a novel machine learning technique to build a classification algorithm that exploits these differences to classify ASes into six representative classes that reflect ASes with different network properties and infrastructures. We derive macroscopic statistics on the different types of ASes in the Internet and validate our results using a sample of 1200 manually identified AS types. Our validation demonstrates that our classification algorithm achieves high accuracy: 78.1% of the examined classifications were correct. Finally, we make our results and our classifier publicly available to promote further research and understanding of the Internet's structure and evolution.

In Section 2 we start with a brief discussion of related work. Section 3 describes the data we used, and in Section 4 we specify the set of AS classes we use in our experiments. Section 5 introduces our classification approach and results. We validate them in Section 6 and conclude in Section 7.

Labels predicted by NB-inform

F Monte Carlo pseudocode

Algorithm 2 shows pseudocode for generating Monte Carlo estimates of the Bayes error rate as the Dirichlet hyper-parameter η varies. The input is W the vocabulary size, L the length of the document, S the number of Monte Carlo samples, τ a smoothing parameter, and θ the prior probability of assigning x to class 1. Recall that the latent class variable y is sampled from a Discrete$(\theta, 1-\theta)$ distribution

We iterate over a grid of η values from 0.01 to 10 in increments of 0.1. For each value of η, we sample two multinomials ϕ_1 and ϕ_2 from a Dirichlet distribution with parameter η. We add a small constant τ to each multinomial and then re-normalize[3] Given ϕ_1, ϕ_2, and L we compute the Bayes error rate. We also compute the Jeffrey's divergence between ϕ_1 and ϕ_2.

In Algorithm 3, we compute the Bayes error rate as the document length L varies. We compute the Bayes error rate over a grid of document lengths ranging from 10 words to 1200 words in increments of 5 words. In Algorithm 4, we compute the Bayes error rate as the vocabulary size W varies. We compute the Bayes error rate over a grid of vocabulary sizes ranging from $1,000$ words to $100,000$ words in increments of $5,000$ words.

In Algorithms 5, 6, and 7 we show the analogous pseudocode for computing the Bayesian classifier error rate as a function of the hyper-parameter η, the document length L, and the vocabulary size W.

[3]Mathematically, a probability vector sampled from a Dirichlet distribution would have no zero entries and thus would not need to be smoothed. However, it is often the case that the entries are so small that they cannot be represented by a computer and are effectively zero.

Algorithm 2: Bayes error rate plotted against the η and Jeffrey's Divergence

Input: W, L, S, τ, θ

for $\eta \in \{.01 : .1 : 10\}$ **do**
  for $s = 1 : S$ **do**
    $\phi_1 \sim \text{Dirichlet}(\eta, W)$;
    $\phi_1 \leftarrow \text{Normalize}(\phi_1 + \tau)$; // $\tau \ll 1$
    $\phi_2 \sim \text{Dirichlet}(\eta, W)$;
    $\phi_2 \leftarrow \text{Normalize}(\phi_2 + \tau)$;
    $E(\eta, s) \leftarrow p_\epsilon(\theta, \phi_1, \phi_2, L, W)$; // Compute using Equation 5.7
    $J(\eta, s) \leftarrow D_J(\phi_1 || \phi_2)$; // Compute Jeffrey's divergence

$\bar{J} \leftarrow \text{mean}(J)$; // For each value of η, average over S samples
$\bar{K} \leftarrow \text{mean}(\log(J))$; // For each value of η, average over S samples
$\bar{E} \leftarrow \text{mean}(\log(E))$; // For each value of η, average over S samples
$\bar{\sigma} \leftarrow \text{std}(\log(E))$;
Plot **errorbar**$([.01 : 10], \bar{E}, \bar{\sigma})$; // Plot $\log(p_\epsilon)$ against η
Plot **errorbar**$(\log([.01 : 10]), \bar{E}, \bar{\sigma})$; // Plot $\log(p_\epsilon)$ against $\log(\eta)$
Plot **errorbar**$(\bar{J}, \bar{E}, \bar{\sigma})$; // Plot $\log(p_\epsilon)$ against average D_J
Plot **errorbar**$(\bar{K}, \bar{E}, \bar{\sigma})$; // Plot $\log(p_\epsilon)$ against average $\log(D_J)$

Algorithm 3: Compute Bayes error rate as document length L varies

Input: W, η, S, τ, θ

for $l \in \{10 : 5 : 1200\}$ **do**
  for $s = 1 : S$ **do**
    $\phi_1 \sim \text{Dirichlet}(\eta, W)$;
    $\phi_1 \leftarrow \text{Normalize}(\phi_1 + \tau)$; // $\tau \ll 1$
    $\phi_2 \sim \text{Dirichlet}(\eta, W)$;
    $\phi_2 \leftarrow \text{Normalize}(\phi_2 + \tau)$;
    $E(l, s) \leftarrow p_\epsilon(\theta, \phi_1, \phi_2, l, W)$; // Compute Equation 5.7

$\bar{E} \leftarrow \text{mean}(\log(E))$; // For each value of l, average over S samples
$\bar{\sigma} \leftarrow \text{std}(\log(E))$
Plot **errorbar**$([10 : 1200],\ \bar{E},\ \bar{\sigma})$; // Plot $\log(p_\epsilon)$ against the length l

Algorithm 4: Compute Bayes error rate as vocabulary size W varies

Input: L, η, S, τ, θ

for $W \in \{1k : 5k : 100k\}$ **do**
  for $s = 1 : S$ **do**
    $\phi_1 \sim \text{Dirichlet}(\eta, W)$;
    $\phi_1 \leftarrow \text{Normalize}(\phi_1 + \tau)$; // $\tau \ll 1$
    $\phi_2 \sim \text{Dirichlet}(\eta, W)$;
    $\phi_2 \leftarrow \text{Normalize}(\phi_2 + \tau)$;
    $E(w, s) \leftarrow p_\epsilon(\theta, \phi_1, \phi_2, l, W)$; // Compute Equation 5.7

$\bar{E} \leftarrow \text{mean}(\log(E))$; // For each value of w, average over S samples
$\bar{\sigma} \leftarrow \text{std}(\log(E))$

Plot **errorbar**$([10k : 50k], \bar{E}, \bar{\sigma})$; // Plot $\log(p_\epsilon)$ against W

Algorithm 5: Bayesian classifier error rate plotted against η and the Jeffrey's Divergence

Input: W, S, τ, θ

for $\eta \in \{0.1, 0.5, 1.0, 5.0, 10.0\}$ **do**
  for $l \in \{15, 250, 600, 1200\}$ **do**
    for $n \in \{1, 10, 50\}$ **do**
      for $s = 1 : S$ **do**
        $\phi_1 \sim \text{Dirichlet}(\eta, W)$;
        $\phi_1 \leftarrow \text{Normalize}(\phi_1 + \tau)$; // $\tau \ll 1$
        $\phi_2 \sim \text{Dirichlet}(\eta, W)$;
        $\phi_2 \leftarrow \text{Normalize}(\phi_2 + \tau)$;
        $x^{(1)} \leftarrow \text{Multinomial}(\phi_1, l, n)$;
        $x^{(2)} \leftarrow \text{Multinomial}(\phi_2, l, n)$;
        $E(\eta, l, n, s) \leftarrow p_\epsilon(l, W, \eta, x^{(1)}, x^{(2)}, \phi_1, \phi_2, \theta)$; // Equation 5.19
        $J(\eta, s) \leftarrow D_J(\phi_1 || \phi_2)$; // Compute Jeffrey's divergence

$\bar{J} \leftarrow \text{mean}(J)$; // For each value of η, average over S samples
$\bar{K} \leftarrow \text{mean}(\log(J))$; // For each value of η, average over S samples
$\bar{E} \leftarrow \text{mean}(\log(E))$; // For each value of η, average over S samples
$\bar{\sigma} \leftarrow \text{std}(\log(E))$;

Plot **errorbar**$(\bar{J}, \bar{E}(:, l, n), \bar{\sigma})$; // $\forall\, l, n$
Plot **errorbar**$([.1, .5, 1, 5, 10], \text{mean}(E(:, l, n)), \text{std}(E(:, l, n)))$; // $\forall\, l, n$

Algorithm 6: Compute Bayesian classifier error rate as document length L varies

Input: W, S, τ, θ

for $\eta \in \{0.1, 1.0, 10.0\}$ **do**
  for $l \in \{1, 15, 50, 250, 600, 1200\}$ **do**
    for $n \in \{1, 10, 50\}$ **do**
      for $s = 1 : S$ **do**
        $\phi_1 \sim \text{Dirichlet}(\eta, W)$;
        $\phi_1 \leftarrow \text{Normalize}(\phi_1 + \tau)$; // $\tau \ll 1$
        $\phi_2 \sim \text{Dirichlet}(\eta, W)$;
        $\phi_2 \leftarrow \text{Normalize}(\phi_2 + \tau)$;
        $x^{(1)} \leftarrow \text{Multinomial}(\phi_1, l, n)$;
        $x^{(2)} \leftarrow \text{Multinomial}(\phi_2, l, n)$;
        $E(\eta, l, n, s) \leftarrow p_\epsilon(l, W, \eta, x^{(1)}, x^{(2)}, \phi_1, \phi_2, \theta)$; // Equation 5.19

$\bar{E} \leftarrow \text{mean}(\log(E))$; // For each value of l, average over S samples
$\bar{\sigma} \leftarrow \text{std}(\log(E))$ Plot **errorbar**$([1, 15, 50, 250, 600, 1200],\ \bar{E}(\eta, :, n),\ \bar{\sigma})$; // $\forall\, \eta, n$

Algorithm 7: Compute Bayesian classifier error rate as vocabulary size W varies

Input: η, S, τ, θ

for $w \in \{1k : 5k : 15k\}$ **do**
  for $\eta \in \{0.1, 1.0, 10.0\}$ **do**
    for $l \in \{1, 15, 50, 250, 600, 1200\}$ **do**
      for $n \in \{1, 10, 50\}$ **do**
        for $s = 1 : S$ **do**
          $\phi_1 \sim \text{Dirichlet}(\eta, w)$;
          $\phi_1 \leftarrow \text{Normalize}(\phi_1 + \tau)$; // $\tau \ll 1$
          $\phi_2 \sim \text{Dirichlet}(\eta, w)$;
          $\phi_2 \leftarrow \text{Normalize}(\phi_2 + \tau)$;
          $x^{(1)} \leftarrow \text{Multinomial}(\phi_1, l, n)$;
          $x^{(2)} \leftarrow \text{Multinomial}(\phi_2, l, n)$;
          $E(w, \eta, l, n, s) \leftarrow p_\epsilon(l, w, \eta, x^{(1)}, x^{(2)}, \phi_1, \phi_2, \theta)$; // Equation 5.19

$\bar{E} \leftarrow \text{mean}(\log(E))$; // For each value of w, average over S samples
$\bar{\sigma} \leftarrow \text{std}(\log(E))$ Plot **errorbar**$([1k : 5k : 15k],\ \bar{E}(:, \eta, l, n),\ \bar{\sigma})$; // $\forall\, \eta, l, n$