5-11-2017

Bayesian Hidden Topic Markov Models

Kenneth Tyler Wilcox
ktw5691@rit.edu

Bayesian Hidden Topic Markov Models

by

Kenneth Tyler Wilcox

A thesis submitted in partial fulfillment of the
requirements for the degree of
Master of Science

in

Applied Statistics

in the

School of Mathematical Sciences

of the

College of Science

of the

Rochester Institute of Technology

Committee in charge:

Associate Professor Ernest Fokoué, Chair
Assistant Professor Cecilia Alm
Assistant Professor Emily Prud'hommeaux
Assistant Professor Wei Qian
Professor Joseph Voelkel

May 11, 2017

Bayesian Hidden Topic Markov Models

Abstract

Bayesian Hidden Topic Markov Models

by

Kenneth Tyler Wilcox

Master of Science in Applied Statistics

Rochester Institute of Technology

Associate Professor Ernest Fokoué, Chair

Recent developments in topic modeling for text corpora have incorporated Markov models in the latent space to better learn contextual content. Known as the Hidden Topic Markov Model (HTMM), this natural extension of probabilistic mixture models relaxes the "bag-of-words" assumption of the foundational latent Dirichlet allocation topic model by allowing the discrete latent variables, or topics, to follow a special first-order Markov process. Parameter estimation is performed using an expectation-maximization (EM) algorithm with fixed dimensionality of the topic space (Gruber, Rosen-Zvi, and Weiss 2007). I fully derive the state space and EM algorithm for the HTMM. I then extend the Hidden Topic Markov Model (HTMM) into a fully Bayesian framework using a Gibbs sampler. The necessary full conditional distributions are derived and a Gibbs sampling algorithm proposed. I implement both the HTMM EM algorithm (Gruber, Rosen-Zvi, and Weiss 2007) and the HTMM Gibbs sampling algorithm in the `R` and `C++` programming languages. The performance of both inferential algorithms is evaluated on twelve simulated data sets and on a collection of proceedings from the Conference on Neural Information Processing Systems (NIPS). The results suggest that the Gibbs sampling algorithm provides better recovery of the topic space than a combination of the EM and Viterbi algorithms. Parameter estimation is comparable using point estimates with both algorithms. The convergence of the Gibbs sampler is studied and is reliable for reasonably large data sets. Evaluation of both algorithms on the NIPS corpus suggests that the HTMM is better able to handle polysemy than LDA and provides coherent and contiguous topics. Predictive accuracy measured by perplexity is better on training and test documents using the HTMM than using LDA on the NIPS corpus. Introducing Markovian dynamics in topical space provides better topical segmentation of a corpus and increased predictive accuracy for unseen documents.

Gibbs sampler; hidden Markov models; hierarchical Bayes; latent variable modeling; mixture models; natural language processing; text mining; topic modeling.

To Jess

For everything before, now, and to come.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor Dr. Ernest Fokoué for his unfailing support and inspiration. If it were not for your encouragement, I would not have completed this work and would not be embarking on a PhD. I would like to thank my committee for their wisdom and support. I would like to thank my parents for never doubting me and encouraging me to persevere. Finally, I offer my gratitude to Buddha, Dharma, and Sangha.

Chapter 1

Introduction

1.1 Problem Statement

The Hidden Topic Markov Model of Gruber, Rosen-Zvi, and Weiss (2007) is extended into a Bayesian framework: posterior distributions are derived and a Gibbs sampling algorithm is developed and implemented in the `R` and `C++` programming languages. The expectation-maximization algorithm for the Hidden Topic Markov Model is derived fully along with a special extension of the forward-backward and Viterbi algorithms for hidden Markov models in order to perform inference. The performance of the expectation-maximization algorithm for the Hidden Topic Markov Model proposed by Gruber, Rosen-Zvi, and Weiss and the novel Gibbs sampling algorithm is evaluated in a simulation study and on a real-world corpus of conference proceedings.

1.2 Overview

The accessibility of vast quantities of text-based information has spurred the development of many computational and statistical approaches to extract and summarize text-based data. Document analysis, in particular, has posed a fruitful and challenging task within the broad set of natural language processing problems. Primary tasks when modeling discrete data like text often include classifying documents or queries, summarizing a body of text, information retrieval, novelty detection, authorship identification, and structural analysis. This thesis concerns itself with topic modeling; topic models offer a statistical model of textual structure.

In this thesis, a novel extension of recent efforts to learn topical representations that are more semantically coherent through the use of hidden Markov models is proposed using a fully Bayesian framework. This is a marked departure from Blei, Ng, and Jordan (2003), whose seminal latent Dirichlet allocation (LDA) assumes that topics are independently distributed in a document. While LDA has become popular for topic modeling, its underlying assumptions ignore the meaning created by the order of words in sentences, paragraphs, and documents. The introduction of a Markov process over the topics is expected to better model the sequential meaning contained in sentences and paragraphs by assuming that the topic of a sentence depends on the topic of the previous sentence. However, introducing a Markov process into the latent space makes estimation and inference non-trivial in this context. Analytical solutions can be intractable, necessitating computational solutions such as the expectation-maximization (EM) algorithm or sampling algorithms like the Gibbs sampler. The inferential approach for the Hidden Topic Markov Model proposed by Gruber, Rosen-Zvi, and Weiss (2007) relies on an EM algorithm (Dempster, Laird, and Rubin 1977). It is well known that the EM algorithm is not guaranteed to converge to a global maximum, which motivates the use of a Gibbs sampling approach since Gibbs sampling produces a Markov chain that will converge to the true posterior distribution in the limit (Gelfand and Smith 1990), although Gibbs samplers may not converge in practice when taking a finite sample. Another advantage of Gibbs sampling is the ability to recover the posterior distribution of model parameters. The EM algorithm treats model parameters as fixed values, and yields point estimates of parameters rather than distributions.

The Hidden Markov Topic Model of Andrews and Vigliocco (2010) proposes a Gibbs sampler for a similar model that makes use of a traditional hidden Markov model, but relies on a different generative model than that of Gruber, Rosen-Zvi, and Weiss (2007). While other time-dependent topic processes have been introduced by others (Blei and Moreno 2001; Blei and Lafferty 2006; Wang and McCallum 2006), there does not appear to be any Bayesian extensions of the Hidden Topic Markov Model.

The Hidden Topic Markov Model aims to produce more coherent topics than LDA by capturing semantic relationships within and between sentences through its use of Markovian dynamics in the topic space that LDA is unable to identify. Another potential benefit of the Markov process is the extraction of topics that are more robust to chaining than LDA. Chained topics occur when two or more distinct subsets of words are combined with words that are ambiguous since they can belong to more than one subset (Boyd-Graber, Mimno, and Newman 2014). Allowing a first-order Markov process to drive the topic transitions is expected to reduce the negative impact of polysemy – multiple meanings for a single word – on topic quality. A fully Bayesian framework for such a topic model allows the approximation of the posterior distributions.

1.3 Organization

The contents of this proposal are organized into four sections. Chapter 1 motivates and introduces the work presented in this thesis. In Chapter 2, related work in the context of document analysis and topic modeling is reviewed with the objective of 1) arguing that a hierarchical probabilistic model can represent the statistical structure of documents and 2) motivating the departure from the "bag-of-words" assumption common to many topic models as a means of obtaining more coherent topic assignments. In Chapter 3, the Gibbs sampler and its use in Bayesian approaches to statistical modeling is discussed and the use of Gibbs sampling for continuous latent variable models and discrete latent variable mixture modeling is illustrated. Chapter 4 presents the hidden Markov model to motivate its use in topic modeling. Chapter 5 introduces the state space required by the Hidden Topic Markov Model and presents derivations of a special forward-backward algorithm, expectation-maximization algorithm, and special Viterbi algorithm. Chapter 6 presents the Bayesian formulation of the Hidden Topic Markov model, derives full conditional distributions for the model parameters and state space, and proposes a Gibbs sampling algorithm. Chapter 7 studies the performance of the EM and Gibbs sampling algorithms on simulated data and then compares the Hidden Topic Markov Model to Latent Dirichlet Allocation on a real-world corpus. Chapter 8 presents conclusions and suggestions for future research.

Chapter 2

Related Work

2.1 Document Analytics and Information Retrieval

Early foundational efforts in information retrieval relied primarily on vector representations of documents using simple word frequencies in a document or transformations of those frequencies such as *tf-idf* (Salton and McGill 1983). In *tf-idf*, term or word frequencies in each document are counted (tf) and then weighted by the inverse of the number of documents in the corpus (idf) to obtain the *tf-idf* measure. Frequently, term frequencies and inverse document frequencies are both normalized to avoid overemphasizing overly common terms or unusually rare terms. By representing a document as a vector of term frequencies, some amount of compression is achieved since the original text no longer needs to be retained. Comparisons of word frequencies and patterns can be performed directly on the vectorized representations. Indeed, one can consider suitably normalized term frequencies and document frequencies as empirical probability distributions of terms over a document or corpus. A corpus can be represented in matrix form as a term-document matrix where a corpus's vocabulary of terms are represented as rows and the documents in the corpus are represented as columns. One major drawback of this approach, however, is that minimal reduction is achieved in this representation, which poses problems for storage as well as speed during tasks such as information retrieval (Blei, Ng, and Jordan 2003; Salton and McGill 1983).

2.2 Latent Semantic Indexing

Latent semantic indexing (LSI) was an initial effort to improve document retrieval for query-based searches. Deerwester et al. (1990) noted that document retrieval solely based on term matching is unreliable for two primary reasons. First, query terms may not be contained in the document or its metadata; a relevant document without exact term matches will not be returned. This problem is known as *synonymy*. Second, *polysemy* – multiple meanings for a given word – can lead to the retrieval of irrelevant documents; these documents will contain matches to a query term, but the documents' terms may have entirely different meanings than the query's intent. Methods relying solely on the original term-document matrix are prone to suffering from both synonymy and polysemy; while exact matches of a query term may not exist in the corpus, synonymous words might. Similarly, polysemous words can inappropriately match query terms as exact matches when the meaning of the word in a document does not match the meaning in the query. Again, using the original document-term matrix, it is impossible to disambiguate multiple word senses. Projection of the document-term matrix into a lower-dimensional space can encode relationships among synonymous words and disambiguate word senses for polysemous words by embedding related words and multiple word senses in a subspace that captures relationships like synonymy and polysemy.

Seminal work by Deerwester et al. (1990) suggested searching for a latent space where projections of the term-document matrix into the latent space yielded a lower-dimensional representation. Furthermore, they proposed that such a projection should capture any assumed underlying semantic structure in the original term-document matrix. Their approach used singular-value decomposition on the term-document matrix to reduce the dimensionality from V to K such that $K < V$ where V is the size of the vocabulary of a corpus and K is the number of latent dimensions. Simultaneously, orthogonal linear projections of documents and terms are obtained by this matrix factorization. The complexity of the latent representation is controlled by K. The LSI approximation of the original term-document matrix can be shown to minimize the Frobenius norm and as such yields a rank-K optimal approximation of the original document-term matrix (Hofmann 1999).

Queries can be projected into the resulting latent space and document similarity is assessed by comparing the latent projection of the query to the latent representation of neighboring documents using vector-based similarity measures such as the inner product. Methods like factor analysis or clustering operate on document similarity matrices or term similarity matrices and are unable to capture relationships between terms and documents.

Unfortunately, LSI assumes strictly linear relationships between terms and documents and provides no probabilistic generative model of a corpus. Subsequent developments such as probabilistic LSA and latent Dirichlet allocation (LDA) provide such a framework.

2.3 Probabilistic Latent Semantic Indexing

A critical development in latent representation of discrete data extended latent semantic indexing (Deerwester et al. 1990)[LSI] by assuming a generative model of terms and topics for each document in a corpus. Rather than find a latent representation of a corpus using a geometric orthogonal norm as in LSI, probabilistic LSI (pLSI) fit a latent projection of the term-document matrix by maximum likelihood for the generative model (Hofmann 1999). From an application perspective, LSI is compelling since the singular value decomposition of the term-document matrix can scale well for large data sets. However, the use of generative probabilistic models for corpora tends to outperform the simplistic LSI model, though they may not scale as easily (Hofmann 1999).

Probabilistic LSI assumes a common generative model (shown in Figure 2.1) for each document in which a document d is chosen according to $p(d)$. A latent variable or topic z is then chosen according to $p(z|d)$. Formally, a topic is a distribution over the vocabulary of all words in the corpus. Finally, a term w in document d is chosen according to $p(w|z)$. Under the "bag-of-words" assumption, documents are assumed to be independent and words w in a document are assumed to be drawn independently given topic z. Hofmann proposed an expectation-maximization solution. Documents can then be described by document-specific distributions of topics $p(z|d)$ instead of distributions of the entire vocabulary.

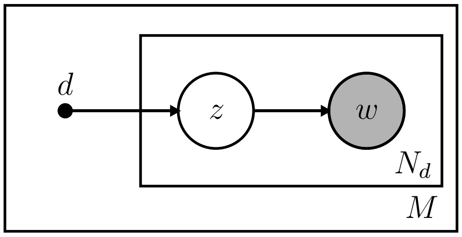

Figure 2.1: Graphical model of probabilistic Latent Semantic Indexing

The distribution of the V-dimensional vocabulary is multinomially distributed over M documents in the pLSI framework. Using the latent space representation, the words are multinomially distributed as a sub-simplex over $K < M$ topics. Therefore, information retrieval can be performed by identifying words in the document space where $p(w|d)$ gives the location of words w on documents $d, d \in \{1, \ldots, D\}$. Alternatively, a more efficient representation identifies words in the topic space where $p(w|z)$ gives the location of words w on topics $z, z \in \{1, \ldots, K\}$. Just as the latent vectors in LSI can be used for similarity comparisons in information retrieval, $p(w|z)$ can be used equivalently.

2.4 Latent Dirichlet Allocation

While pLSI represents documents as a set of the mixing proportions for the topics (i.e., a probability distribution on the fixed topics), it does not model the documents from a probabilistic model. As a result, there are $K(M + V)$ parameters to learn for K topics, M documents, and a vocabulary of V words. As a result, the number of parameters grows linearly with the corpus size. Furthermore, it is difficult to assign probabilities to documents outside the training set (Blei, Ng, and Jordan 2003), prohibiting generalization of the pLSI model outside the training corpus.

Latent Dirichlet allocation (LDA) resolves these limitations by providing a probabilistic generative model of the documents, resulting in $K + KV$ parameters to learn and further allowing for unseen documents to be classified. The "bag-of-words" assumption in pLSI is preserved in LDA such that the words are assumed to be independent given topics and the topics are independent conditioned on the Dirichlet random variable θ_d for a given document.

The generative model for LDA is similar to that of pLSI, except that LDA assumes a probabilistic model of the documents in addition to that of the topics and words. Latent Dirichlet Allocation assumes that the corpus contains M documents where each document $d \in \{1, \ldots, M\}$ is generated by the following process:

1. Draw the number of words in the document $N_d \sim \text{Poisson}(\psi)$
2. Draw $\theta_d \sim \text{Dirichlet}(\alpha)$
3. For each word w_n in document $d, n \in \{1, \ldots, N_d\}$
   a) Draw topic $z_n \sim \text{Multinomial}(\theta_d)$
   b) Draw word $w_n|z_n \sim \text{Multinomial}(\beta_{z_n})$

The topic distribution depends on the K-dimensional random variable θ_d which is drawn once for each document. The words are drawn independently from a multinomial parameterized by β where β is a $K \times V$ matrix of the conditional probabilities of word j given topic i, $p(w_j|z_i = 1)$. Parameters α and β are corpus-level parameters that are the same for all documents while θ is a document-level parameter. Note that symmetric prior distributions are used for θ_d where the parameters for the Dirichlet prior are all equal to α. The graphical model for the original LDA model (Blei, Ng, and Jordan 2003) is shown in Figure 2.2.

As a result of the structure of the generative model, all documents share a common set of topics, but the topics are expressed with different probabilities in each document. The joint

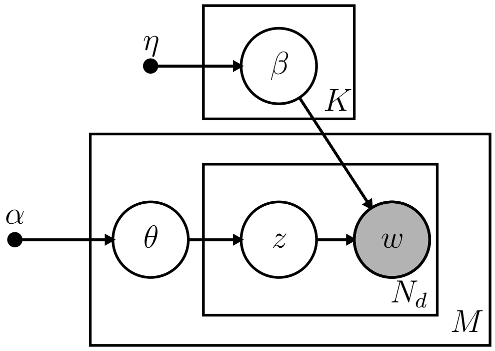

Figure 2.2: Graphical model of Latent Dirichlet Allocation

distribution of the topic mixture θ_d, the N_d topics $z_1, \ldots, z_{N_d}$, and N_d words $w_1, \ldots, w_{N_d}$ can be written as

$$p(\theta_d, z_1, \ldots, z_{N_d}, w_1, \ldots, w_{N_d}|\alpha, \beta) = p(\theta_d|\alpha)p(z_1, \ldots, z_{N_d}|\theta_d)p(w_1, \ldots, w_{N_d}|z_1, \ldots, z_{N_d}, \beta). \tag{2.1}$$

Taking advantage of the conditional independence of the N_d topics given θ_d, equation 2.1 can be rewritten as

$$p(\theta_d, z_1, \ldots, z_{N_d}, w_1, \ldots, w_{N_d}|\alpha, \beta) = p(\theta_d|\alpha) \prod_{n=1}^{N_d} p(z_n|\theta_d)p(w_n|z_n, \beta). \tag{2.2}$$

By marginalizing out the document Dirichlet parameter θ_d and the topics in equation 2.2, the marginal distribution of a document is

$$p(w_1, \ldots, w_{N_d}|\alpha, \beta) = \int_{\theta_d} p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_n} p(z_n|\theta_d)p(w_n|z_n, \beta) \right) d\theta_d. \tag{2.3}$$

Furthermore, the probability of a corpus $\mathbf{D}$ can be obtained simply since the documents $\mathbf{w_d}$ are assumed to be independent:

$$\begin{aligned} p(\mathbf{D}|\alpha,\beta) &= p(\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_M|\alpha,\beta) \\ &= \prod_{d=1}^{M} p(\mathbf{w}_d|\alpha,\beta) \\ &= \prod_{d=1}^{M} \int_{\theta_d} p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_n} p(z_n|\theta_d) p(w_n|z_n,\beta) \right) d\theta_d. \end{aligned} \tag{2.4}$$

Finally, the joint distribution of words and topics for a document can be obtained by marginalizing out θ_d:

$$p(w_1,\ldots,w_{N_d},z_1,\ldots,z_{N_d}|\alpha,\beta) = \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} p(z_n|\theta_d) p(w_n|z_n) \right) d\theta_d. \tag{2.5}$$

One simpler alternative to LDA, a mixture of unigrams (Nigam et al. 2000), represents documents as word distributions conditioned on a single topic (shown in Figure 2.3):

$$p(w_1,\ldots,w_{N_d}) = \sum_z p(z) \prod_{n=1}^{N_d} p(w_n|z). \tag{2.6}$$

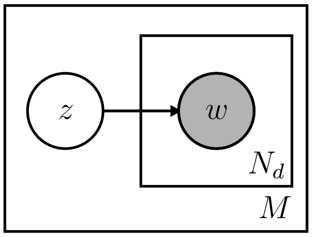

Figure 2.3: Graphical model of mixture of unigrams

This much simpler mixture model has been shown to inadequately represent large corpora (Blei, Ng, and Jordan 2003). However, it is informative to view LDA in the context of a mixture of unigrams since LDA can be considered as a continuous mixture of unigrams if the joint distribution of the words and topics conditioned on θ_d is marginalized over the topics:

$$p(w_1, \ldots, w_{N_d} | \theta_d, \beta) = \sum_{z_d} p(w_1, \ldots, w_{N_d} | z_1, \ldots, z_{N_d}, \beta) p(z_1, \ldots, z_{N_d} | \theta_d). \tag{2.7}$$

This results in a representation of documents from LDA as a continuous mixture:

$$p(w_1, \ldots, w_{N_d} | \alpha, \beta) = \int p(\theta_d | \alpha) \left(\prod_{n=1}^{N_d} p(w_n | \theta_d, \beta) \right) d\theta_d. \tag{2.8}$$

Here, $(w_n|\theta_d, \beta)$ is a random variable and $p(\theta_d|\alpha)$ defines mixture weights. Most interestingly, the LDA continuous mixture of unigrams only requires K parameters to estimate for $p(\theta_d|\alpha)$ instead of the $K - 1$ parameters needed for $p(z)$ in the simple mixture of unigrams model shown in Figure 2.3 while improving the quality of topic allocation substantially.

Blei, Ng, and Jordan (2003) used a variational Bayes algorithm to estimate the parameters and topics of the LDA model and demonstrated marked improvement over simple unigrams, mixtures of unigrams, and pLSI, establishing LDA as a standard for topic modeling. However, a fully Bayesian solution was developed using Gibbs sampling (Griffiths and Steyvers 2002, 2004). More recently, a faster collapsed Gibbs sampler for LDA was proposed by Porteous et al. (2008).

2.5 Departures from the “Bag-of-Words” Assumption

The popularity and success of LDA (Blei, Ng, and Jordan 2003) spurred an active body of research to extend LDA beyond its limiting assumptions. As mentioned in section 2.4, the hierarchical Bayesian model proposed for LDA assumes (unrealistically) that the words are exchangeable given a topic and that the documents are exchangeable within a corpus. When treating a topic as a collection of related words, it is reasonable to ignore the meaning created in natural language by the order of words in a sentence, the order of sentences in a paragraph, the order of paragraphs in a document, and even the order of documents in a corpus. Treating a corpus as an unordered collection of documents is reasonable in many settings, but may not be reasonable, for example, when studying the development of literature on document analysis over the 20th century. While the “bag-of-words” assumption is computationally convenient, improvements to topic modeling have generally been driven by efforts to relax the assumption of independent topics or the assumption of independent documents.

Correlated Topic Modeling

One approach to inducing a covariance structure in the LDA framework takes advantage of the hierarchical structure of the model. Blei and Lafferty (2005) introduced correlated topic models (CTM) to capture the covariance structure of topics. This allows for correlations as well as independence among topics. The crucial difference between LDA and CTM is the use of a logistic-normal distribution to model topic proportions instead of a Dirichlet distribution.

Much like LDA, the correlated topic model assumes that the corpus contains M documents where each document $d \in \{1, \ldots, M\}$ is generated by the following process:

1. Draw the number of words in the document $N_d \sim \text{Poisson}(\psi)$
2. Draw $\eta_d \sim \text{N}(\mu, \Sigma)$
3. For each word w_n in document d
   a) Draw topic $z_n | \eta_d \sim \text{Multinomial}(f(\eta_d))$
   b) Draw word $w_n | z_n \sim \text{Multinomial}(\beta_{z_n})$

where $f(\eta_d) = \frac{\exp \eta_i}{\sum_j \exp \eta_j}$. The CTM graphical model is shown in Figure 2.4

Blei and Lafferty (2005, 2007) proposed a variational algorithm to perform inference for CTM. Their results suggested that CTM provides a better fit to corpora than LDA and is better able to represent larger numbers of topics. Recent work has examined correlated topic modeling using a probit-normal model instead of the logistic-normal model of CTM (Yu and Fokoue 2014).

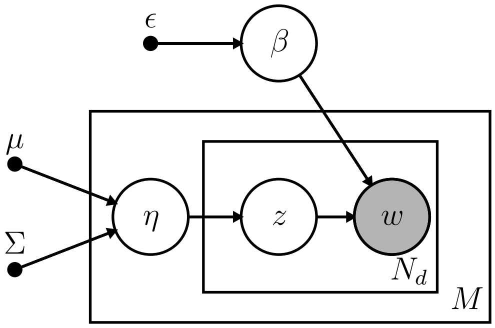

Figure 2.4: Graphical model of the correlated topic model

Dynamic Topic Modeling

While correlated topic modeling (CTM) allows for inference on the topic correlation structure to be performed, it does not allow for the evolution of topics over time. Blei and Lafferty (2006) proposed an alternative extension of LDA that, like the correlated topic model, relaxes the independence assumption for topics. Rather than model the correlation structure of topics, they assume that documents develop in a Gaussian time series and that topics in a given interval of the document time span follow another Gaussian time series. Topic proportions are assumed to follow a Dirichlet distribution as in LDA (Blei, Ng, and Jordan 2003) and CTM (Blei and Lafferty 2005). Inference is performed using variational approximation of the posterior distributions. Blei and Lafferty (2006) demonstrated that the dynamic topic model outperformed static LDA topic models. The development of the dynamic topic model was preceded by a simpler hidden Markov model approach to topic identification by Blei and Moreno (2001) that solely focused on unstructured streams of words.

Other attempts to relax the "bag-of-words" assumption have been proposed. Wang and McCallum (2006) developed a modified version of LDA that allows topics to develop over time. They place a beta distribution over a normalized continuous time index and learn the distribution of topics and time by Gibbs sampling. Wallach (2006) developed a hierarchical Bayesian model that combines the latent structure of LDA with aspects of a hierarchical

Dirichlet language model (MacKay and Peto 1995). In this model, the probability of a word w_i depends on the previous word w_{i-1} as well as the topics $\mathbf{z}$. This effectively extends LDA by imposing a first-order Markov process on the words in a document and relies on a Gibbs EM algorithm to perform inference. Performance was better using this hybrid model than both LDA and the hierarchical Dirichlet language model. A similar model was proposed within the cognitive science community by Griffiths, Steyvers, and Tenenbaum (2007). Other notable approaches to topic modeling include syntactic topic models (Boyd-Graber and Blei 2009), constrained topic assignments (Chen et al. 2009), network analysis (Zhang, Zhu, and Zhang 2013; Bouveyron, Latouche, and Zreik 2016), author-topic models (Rosen-Zvi et al. 2010), P'olya urn topic models (Mimno et al. 2011), spectral LDA (Anandkumar et al. 2012), neural network topic models (Wan, Zhu, and Fergus 2012), hidden stochastic automata (Andrews 2013), and augmented max-margin topic models (Zhu et al. 2014). For an accessible review of the development of topic modeling, see Blei (2012).

2.6 Hidden Topic Markov Modeling

Improved topic quality and predictive performance relative to latent Dirichlet allocation (LDA) have been achieved by using Markov modeling for the observed words (Wallach 2006, bigrams). Gruber, Rosen-Zvi, and Weiss (2007) noted that it would be reasonable to adopt a Markov process in the latent space since it is reasonable to assume that topics would change over time in a given document. Therefore, they modified LDA by introducing a hidden Markov model (HMM); this is similar to the work of Blei and Moreno (2001), though that model does not allow for a mixture of topics in a document or segment of text. The hidden topic Markov model (HTMM) of Gruber, Rosen-Zvi, and Weiss (2007) assumes that topics are likely to be contiguous throughout a document; this property is modeled with a first-order discrete Markov chain in the topic space. Specifically, the HTMM assumes that topics are fixed for a sentence so that all words in a sentence share a single topic. In the LDA model, topics are independent when conditioned on topic proportions θ_d and sentences can be composed of multiple topics. In the HTMM, topics in a document are dependent on θ_d and transition indicator variables $\psi_n, n \in \{1, \ldots, N_d\}$, where $\psi \in \{0, 1\}$. When $\psi_n = 1$, a new topic z_n is drawn according to θ_d and when $\psi_n = 0$, the topic is not changed so $z_n = z_{n-1}$. Since sentences are assumed to contain a single topic, the Markov chain is only allowed to change state at the first word of each sentence (i.e., $\psi_n = 0$ for words other than the first words in a sentence). The generative model of a document as shown in Figure 2.5 is described below:

```
1. For $z \in \{1, \ldots, K\}$
     Draw $\beta_z \sim \text{Dirichlet}(\eta)$
2. Draw $\theta_d \sim \text{Dirichlet}(\alpha)$
3. Set $\psi_1 = 1$
4. For each word $w_n$ in document $d$
     a) If $w_n$ begins a sentence
          Draw $\psi_n \sim \text{Binomial}(\epsilon)$
          Else $\psi_n = 0$
     b) For $n \in \{1, \ldots, N_D\}$
          i. If $\psi_n == 0$
               $z_n := z_{n-1}$
               Else $z_n \sim \text{Multinomial}(\theta_d)$
          ii. Draw word $w_n | z_n \sim \text{Multinomial}(\beta_{z_n})$
```

One disadvantage of the hidden topic Markov model (HTMM) is its storage requirements. While latent Dirichlet allocation (LDA) and other "bag-of-words" topic models use a term-document matrix as input, HTMM requires the entirety of each document. The cost of storing the entire corpus is balanced by allowing for more expressive representations of documents. Perhaps most notably, words are more likely to be drawn from multiple topics in a single document in HTMM than in LDA due to the Markov process which could allow

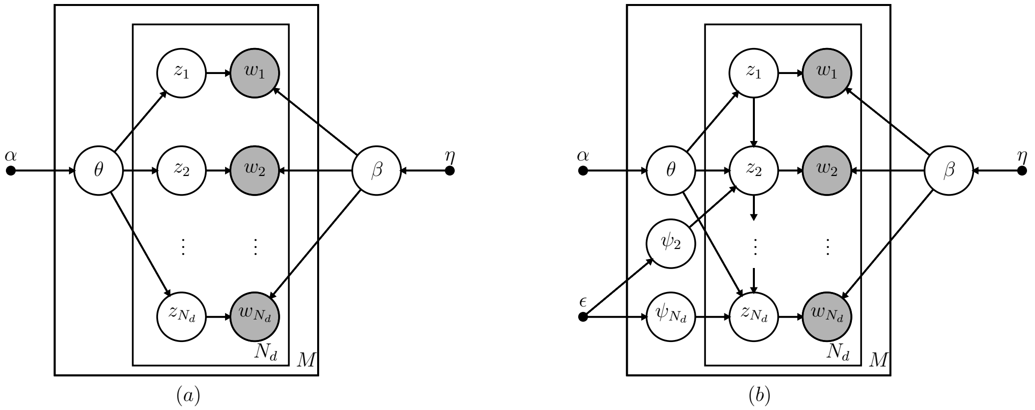

Figure 2.5: (a) Graphical model of the latent Dirichlet allocation topic model. (b) Graphical model of the hidden topic Markov model. Word generation is drawn explicitly to highlight the topic independence in LDA versus the topic Markov chain in HTMM.

for better disambiguation of polysemous words. LDA tends to assign a given word to a single or very few topics regardless of where the word occurs in a document or in a corpus. This is undesirable, for example, if a mathematical paper discussing *support* vector machines also referred to the *support* of a grant in its acknowledgments. Gruber, Rosen-Zvi, and Weiss (2007) showed that for this example, HTMM is capable of assigning the word *support* in the support vector machine context to a mathematical topic and the word *support* in the acknowledgements section to a document metadata section. The HTMM may be of particular interest for natural language processing due to its ability to better capture and disambiguate these different word senses.

Gruber, Rosen-Zvi, and Weiss (2007) make use of the well-studied Hidden Markov Model (HMM) to approximate the posterior probabilities. Conditioned on β and θ, the hidden topic Markov model is a form of HMM so the forward-backward algorithm and the EM algorithm can be easily used for parameter estimation. In this framework, latent variables z_n and driving variables ψ_n are drawn from $p(z_n, \psi_n | d, w_1, \ldots, w_{N_d}; \theta_d, \beta, \epsilon)$ where θ_d, β, and ϵ are considered parameters to be estimated. The joint conditional distribution of z_n and ψ_n is computed with the forward-backward algorithm for HMM and θ_d, β, ϵ are updated in the maximization step.

While the authors acknowledged that the EM algorithm may be less preferable than a Gibbs sampler since EM is known to converge to local optima instead of a global optimum, they argued that their EM algorithm was robust to various initializations. In this thesis, I derive a Gibbs sampler to provide a Bayesian alternative to the EM algorithm. Furthermore, I am interested in studying the structure of the resulting topic model which is better accomplished by approximating the joint posterior distribution of the model parameters by Gibbs sampling than by point estimates alone. Results from Gruber, Rosen-Zvi, and Weiss (2007) suggested that the HTMM provided lower perplexity scores than LDA which indicated that HTMM better predicted the words in a new corpus. Furthermore, qualitative analysis suggested that polysemy or word senses were better disambiguated using HTMM than LDA. Unfortunately, perhaps due to lack of space for publication, Gruber, Rosen-Zvi, and Weiss (2007) did not provide the derivation of the EM algorithm used for the HTMM. I derive their EM algorithm, a special forward-backward algorithm, and a special Viterbi algorithm and then derive a Gibbs sampling algorithm for inference and estimation. Finally, the performance of the HTMM are assessed in a simulation study and on a real-world corpus.

It is worth noting that the Hidden Markov Topic Model proposed by Andrews and Vigliocco (2010) is similar to the Hidden Topic Markov Model of Gruber, Rosen-Zvi, and Weiss (2007). Since Andrews and Vigliocco did not mention Gruber, Rosen-Zvi, and Weiss, it appears that the two approaches developed independently. This thesis focuses on the Hidden Topic Markov Model, but future work could consider a comparison of the two approaches.

Chapter 3

Gibbs Sampling and the Bayesian Framework

3.1 Bayesian Probability

To motivate the use of Bayesian methods for topic modeling, it is important to understand the philosophical framework of Bayesian probability. Classical or frequentist statistics consider probability as a long-term expectations. Methods such as maximum likelihood and the Expectation-Maximization algorithm consider probability in a frequentist sense.

For a set of n random variables $X = \{X_1, X_2, \ldots, X_n\}$, let $p_\theta(X|\theta)$ be the likelihood or joint probability of the data X given a parameter θ. Inference can be performed by seeking a value of θ that was most likely to have generated the observed data X by solving

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \left\{ p(X; \theta) \right\}.$$

This approach assumes that θ is fixed instead of being a random variable.

The Bayesian framework instead assumes that θ was drawn from a distribution known as a prior distribution $p(\theta)$. Using Bayes Theorem, the likelihood function and the prior distribution can be used to obtain a posterior distribution of $\theta|X$. The introduction of a prior distribution allows one to make probabilistic statements about θ given the available data. Different choices of prior distributions can be used to encode different *a priori* beliefs about θ before observing data. Inference can be performed using the posterior distribution instead of the likelihood function. The posterior distribution of the parameter(s) given the data is expressed as

$$\begin{aligned} p(\theta|X) &= \frac{p(X,\theta)}{p(X)} \qquad (3.1) \\ &= \frac{p(X|\theta)p(\theta)}{p(X)}. \qquad (3.2) \end{aligned}$$

Often, the value of $p(X)$ is not needed as it is only a normalizing constant. It is often sufficient to manipulate a distribution proportional to the posterior that is just the product of the likelihood and the prior and then normalize that distribution since $p(X)$ contains no information about θ;

$$p(\theta|X) \propto p(X|\theta)p(\theta).$$

This approach can be used to obtain the posterior distribution of the parameter given the data in addition to point estimates of the parameter. Furthermore, the use of explicit prior distributions make a priori hypotheses about the parameter space clear. Indeed, it should be straightforward to see that the use of an improper prior $p(\theta) \propto 1$ can be used to write the likelihood as a posterior distribution in which all values of the parameter θ are considered equally likely a priori. Notably, Bayesian estimates of parameters will converge to their maximum-likelihood counterparts if the size of the data n grows large. Such estimates also allow inference to be performed under conditions where maximum likelihood estimates are not tractable (e.g., when a model is underdetermined). This is made possible by the use of the prior distribution. Maximum-likelihood analogues can be obtained in the Bayesian framework through maximum-a-posteriori estimates of parameters. Commonly chosen estimators include the posterior mean, the posterior median, and the posterior mode, although one advantage of obtaining the posterior distribution is the ability to use the full distribution of $\theta|X$ to make probabilistic statements about $\theta|X$.

3.2 Markov Chain Monte Carlo

While direct analytical solutions can be determined in some cases for Bayesian formulations, it is quite common that alternative computational solutions are proposed to avoid intractable analytical problems. Markov chain Monte Carlo (MCMC) is a common general strategy for sampling from distributions whose complete form cannot be specified or directly sampled from. MCMC is used when a distribution $p(x)$ cannot be sampled from directly but can be evaluated up to some normalizing constant. An immediate use can be seen when considering sampling from an intractable posterior distribution which can be approached by using MCMC to sample from the posterior proportional to a normalizing constant instead of the posterior itself. The goal of MCMC algorithms is to generate a sample of size m by sampling $x^{(i)}, i = 1, \ldots, m$ from the state space of a Markov chain $\mathcal{X}$. By construction, MCMC samplers visit more probable locations in $\mathcal{X}$, facilitating construction of $p(x)$ without spending too much time in unimportant regions of $\mathcal{X}$ provided that the transition kernel of the chain is irreducible and aperiodic. Proper MCMC samplers are irreducible and aperiodic Markov chains that converge to the target distribution (e.g. Andrieu et al. 2003).

The use of Gibbs sampling is motivated by discussing its relation to the Metropolis-Hastings algorithm.

3.3 Metropolis-Hastings Algorithm

The Metropolis-Hastings (MH) algorithm is a general Markov Chain Monte Carlo (MCMC) sampler (Hastings and K. 1970; Metropolis et al. 1953). Each step of the MH sampler tries to sample from a target distribution $p(x)$ by sampling a candidate x^* from a proposal distribution $q(x^*|x)$ given the current value in the chain x. The chain moves to x^* according to the acceptance probability $\mathcal{A}(x, x^*) = \min\left\{1, \frac{p(x^*)q(x|x^*)}{p(x)q(x^*|x)}\right\}$ or else it remains at x:

```
Initialize $x^{(0)}$;
for $i = 0$ to $m - 1$ do
    sample $u \sim U(0, 1)$;
    sample $x^* \sim q(x^*|x^{(i)})$;
    if $u < \mathcal{A}(x^{(i)}) = \min\left\{1, \frac{p(x^*)q(x^{(i)}|x^*)}{p(x^{(i)})q(x^*|x^{(i)})}\right\}$ then
        $x^{(i+1)} = x^*$;
    else
        $x^{(i+1)} = x^{(i)}$;
    end
end
```

Figure 3.1: Metropolis-Hastings sampler

The successful convergence of the chain and its rate of convergence both depend on the construction of the proposal distribution $q(x^*|x)$. A poorly chosen proposal distribution will result in slow convergence and may even result in the Markov chain being stuck in an absorbing state (e.g. Andrieu et al. 2003). However, several key properties make the MH algorithm appealing. The target distribution $p(x)$ need not be fully specified and instead need only be known proportional to its normalizing constant. Furthermore, independent MH chains can be run in parallel, making the algorithm scaleable for large data. Of course, careful assessment of the final chain is critical to assess proper mixing while strategies such as thinning the chain can be used to decrease the correlation among samples. Furthermore, application of simulated annealing can be used to increase the rate of sampling near the global maxima of $p(x)$ (e.g. Andrieu et al. 2003). Finally, a very useful property of the MH algorithm is its utility as a component of an MCMC sampler that uses a mixture or cycle of several samplers. Therefore, large regions of a state space $\mathcal{X}$ can be explored using a global proposal sampler while more localized regions of $\mathcal{X}$ such as global maxima can be explored using local proposals. Popular examples of this mixed sampler include reversible jump MCMC (Green 1995) and block MCMC.

3.4 Gibbs Sampling

While Bayesian formulations are theoretically appealing, they have historically proven difficult to obtain computationally. They often required the use of highly problem-specific computational strategies and sophisticated analytical solutions (Gelfand and Smith 1990) such as the general Metropolis-Hastings algorithm (3.3). However, as Gelfand and Smith point out, the development of the Gibbs sampler and related substitution sampling schemes provided general-purpose, if slightly slower, computational solutions for a wide body of Bayesian problems. Indeed, Gelfand and Smith is widely considered to be the start of wider use of Markov Chain Monte Carlo methods for Bayesian inference (Andrieu et al. 2003; Cappé, Moulines, and Rydén 2005). The reason for the popularity of the Gibbs sampler is its relatively simple formulation of proposal distributions that relies only on the availability of full conditional distributions.

The Gibbs sampler can be derived from the Metropolis-Hastings sampler. Consider a p-dimensional probability vector x (i.e., x has a multivariate distribution over p random variables). Define the full conditionals $p(x_j|x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_p), j \in \{1, \ldots, p\}$. Recall that a Metropolis-Hastings algorithm requires a proposal distribution $q(x^*|x)$ that is proportional to the target distribution. Therefore, consider the proposal distribution for $j = 1, \ldots, p$

$$q(x^*|x^{(i)}) = \begin{cases} p(x_j^*|x_{-j}^{(i)}) & x = x_{-j}^{(i)} \\ 0 & \text{otherwise} \end{cases}$$

where x_{-j} denotes all components of x except x_j.

The acceptance probability used to update x is

$$\begin{aligned}
\mathcal{A}(x^{(i)}, x^*) &= \min\left\{1, \frac{p(x^*)q(x^{(i)}|x^*)}{p(x^{(i)})q(x^*|x^{(i)})}\right\} \\
&= \min\left\{1, \frac{p(x^*)p(x_j^{(i)}|x_{-j}^{(i)})}{p(x^{(i)})p(x_j^*|x_{-j}^*)}\right\} \\
&= \min\left\{1, \frac{p(x_{-j}^*)}{p(x_{-j}^{(i)})}\right\} \\
&= 1.
\end{aligned}$$

As a consequence, Algorithm 3.1 can be simplified to describe a generic Gibbs sampler:

```
Initialize $x_{1:p}^{(0)}$;
for $i = 0$ to $m - 1$ do
    sample $x_1^{(i+1)} \sim p(x_1^{(i)}|x_2^{(i)}, x_3^{(i)}, \dots, x_p^{(i)})$;
    sample $x_2^{(i+1)} \sim p(x_2^{(i)}|x_1^{(i)}, x_3^{(i)}, \dots, x_p^{(i)})$;
    $\vdots$
    sample $x_j^{(i+1)} \sim p(x_j^{(i)}|x_1^{(i)}, \dots, x_{j-1}^{(i)}, x_{j+1}^{(i)}, \dots, x_p^{(i)})$;
    $\vdots$
    sample $x_p^{(i+1)} \sim p(x_p^{(i)}|x_1^{(i)}, x_2^{(i)}, \dots, x_{p-1}^{(i)})$;
end
```

Figure 3.2: Gibbs sampler

3.5 A Gibbs Sampler for a Univariate Latent Variable Model

For a simple Gibbs sampling example, consider a model for a continuous random variable Y generated based on a continuous latent variable Z:

$$Y = \theta Z + \epsilon$$

where it is assumed that $\epsilon \sim N(0, \sigma^2)$, σ^2 is fixed and known, and $Z \sim N(0, 1)$. As a result,

$$(y_i|\theta, z_i) \sim N\left(\theta z_i, \sigma^2\right).$$

A Bayesian framework to learn the posterior distribution of θ and Z is implemented using a Gibbs sampler. Prior distributions must be specified for ϵ, Z, and θ:

$$\theta \sim N\left(\mu_0, \tau^2\right),$$

$$p(\epsilon, Z) = p(\epsilon)p(Z),$$

where μ_0 and τ^2 are hyperparameters. With this choice of conjugate priors, the joint posterior of θ and $z_i, i = 1, \ldots, n$ is:

$$p(\theta, z_i) \propto p(y_i|\theta, z_i)p(\theta, z_i).$$

To implement the Gibbs sampler, the full conditional distributions of θ and z_i are needed. First, the full conditional distribution of z_i is derived:

$$\begin{aligned}
p(z_i|y_i, \theta) &\propto p(y_i|z_i, \theta)p(z_i) \\
&= \exp\left\{-\frac{1}{2}(\sigma^2)^{-1}(y_i - \theta z_i)\right\} \cdot \exp\left\{-\frac{1}{2}z_i^2\right\} \\
&= \exp\left\{-\frac{1}{2}(y_i - \theta z_i)(\sigma^2)^{-1}(y_i - \theta z_i)^2\right\} \cdot \exp\left\{-\frac{1}{2}z_i^2\right\} \\
&= \exp\left\{-\frac{1}{2}\left[y_i(\sigma^2)^{-1}y_i - y_i(\sigma^2)^{-1}\theta z_i - \theta z_i(\sigma^2)^{-1}y_i + \theta z_i(\sigma^2)^{-1}\theta z_i\right]\right\} \cdot \exp\left\{-\frac{1}{2}z_i^2\right\}.
\end{aligned}$$

Recognizing that this expression is a convolution of two Gaussians with respect to z_i,

$$(z_i|y_i, \theta) \propto N\left(\frac{y_i}{\theta}, \frac{\sigma^2}{\theta^2}\right) \cdot N(0, 1).$$

Finally, the full conditional of $(z_i|y_i, \theta)$ is:

$$(z_i|y_i, \theta) \propto N\left(\frac{\theta y_i}{\theta^2 + \sigma^2}, \frac{\sigma^2}{\theta^2 + \sigma^2}\right).$$

Next, the full conditional of θ is derived:

$$\begin{aligned} p(\theta|y,z) &\propto p(y|\theta,z)\cdot p(\theta) \\ &= \exp\left\{-\frac{1}{2}(y-\theta z)^T(\sigma^2 I_n)^{-1}(y-\theta z)\right\}\cdot\exp\left\{-\frac{1}{2}(\theta-\mu_0)(\tau_0^2)^{-1}(\theta-\mu_0)\right\} \\ &= \exp\left\{-\frac{1}{2}\left[y^T(\sigma^2 I_n)^{-1}y - y^T(\sigma^2 I_n)^{-1}\theta z - z^T\theta^T(\sigma^2 I_n)^{-1}y + z^T\theta^T(\sigma^2 I_n)^{-1}\theta z\right]\right\} \\ &\times\exp\left\{-\frac{1}{2}(\theta-\mu_0)(\tau_0^2)^{-1}(\theta-\mu_0)\right\}. \end{aligned}$$

Recognizing that this expression is a convolution of two Gaussians with respect to θ,

$$\begin{aligned} (\theta|y,z) &\propto N\left(\left(z^T(\sigma^2 I_n)^{-1}z\right)^{-1}z^T(\sigma^2 I_n)^{-1}y, \left(z^T(\sigma^2 I_n)^{-1}z\right)^{-1}\right)\cdot N(\mu_0,\tau_0^2) \\ &= N\left(\frac{\left(z^T(\sigma^2 I_n)^{-1}z\right)^{-1}z^T(\sigma^2 I_n)^{-1}y\cdot\tau_0^2+\mu_0\cdot\left(z^T(\sigma^2 I_n)^{-1}z\right)^{-1}}{(z^T(\sigma^2 I_n)^{-1}z)^{-1}+\tau_0^2}, \frac{\left(z^T(\sigma^2 I_n)^{-1}z\right)^{-1}\cdot\tau_0^2}{(z^T(\sigma^2 I_n)^{-1}z)^{-1}+\tau_0^2}\right) \\ &= N\left(\frac{\sigma^2||z||^{-2}\cdot z^Ty(\sigma^2)^{-1}tau_0^2+\mu_0\sigma^2||z||^{-2}}{\sigma^2||z||^{-2}+\tau_0^2}, \frac{\sigma^2||z||^{-2}\tau_0^2}{\sigma^2||z||^{-2}+\tau_0^2}\right) \\ &= N\left(\frac{\tau_0^2 z^Ty+\mu_0\sigma^2}{||z||^2}\cdot\frac{||z||^2}{\sigma^2+\tau_0^2||z||^2}, \frac{\sigma^2\tau_0^2}{\sigma^2+||z||^2\tau_0^2}\right). \end{aligned}$$

Finally,

$$(\theta|y,z) \propto N\left(\frac{\tau_0^2 z^Ty+\mu_0\sigma^2}{\tau_0^2||z||^2+\sigma^2}, \frac{\sigma^2\tau_0^2}{\tau_0^2||z||^2+\sigma^2}\right).$$

Equipped with the full conditional distributions, the resulting Gibbs sampler is:

```
Initialize $\theta$;
for $t = 0$ to $T - 1$ do
    for $i = 1$ to $n$ do
        sample $z_i^{(t)} \sim N\left(\frac{\theta y_i}{\theta^2+\sigma^2}, \frac{\sigma^2}{\theta^2+\sigma^2}\right)$;
    end
    sample $\theta^{(t)} \sim N\left(\frac{\tau_0^2 z^T y+\mu_0\sigma^2}{\tau_0^2||z||^2+\sigma^2}, \frac{\sigma^2\tau_0^2}{\tau_0^2||z||^2+\sigma^2}\right)$;
end
```

Figure 3.3: Gibbs sampler for univariate latent variable model

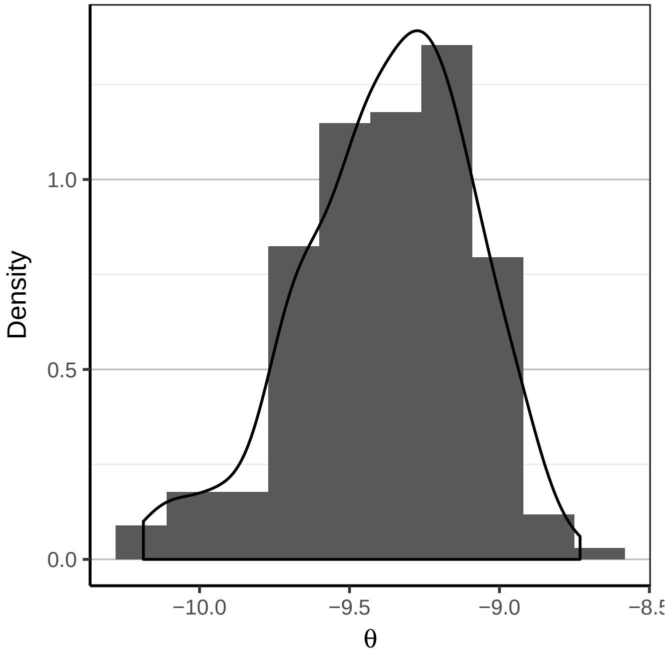

Figure 3.4: Empirical posterior distribution of theta

The posterior distribution for a sample of size $n = 200$ with $\sigma^2 = 1$ is shown in Figure 3.4. A data set of $n = 200$ observations were generated using $\theta = 10, \sigma^2 = 1$. The posterior is approximately normal-distributed as expected form the theoretical form of the posterior. Maximum *a posteriori* estimates of θ using the empirical mean and median are -9.3757 and -9.3411, respectively. The prior mean μ_0 for θ was initialized to $\mu_0 = 5$ to see if the sampler could recover the true θ with a biased prior. The posterior includes $\theta = 10$ although the influence of the prior is evident since the posterior distribution is centered above $\theta = 10$. Finally, inference on θ can be performed using a 95% Bayesian credible interval: (-9.7201, -9.4612). The Bayesian credible interval does not contain $\theta = 10$, which demonstrates the impact of choosing a prior for θ with a mean above the true θ.

3.6 A Gibbs Sampler for a Mixture of Two Gaussians

Next, consider a simple model for a continuous random variable and a discrete latent space. Suppose $Y \in \mathbb{R}$ where Y is assumed to be generated by a mixture of two univariate Gaussians:

$$p(y;\theta) = \pi\phi\left(y;\mu_1,\sigma_1^2\right) + (1-\pi)\phi\left(y;\mu_2,\sigma_2^2\right)$$

where $\theta = (\pi, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$.

Assume that membership in the two Guassians is represented by a latent class variable $Z \in \{0,1\}$ where $Z \sim$ Bernoulli(π)

$$p(z_i) = \pi_i^z(1-\pi)^{1-z_i}, i \in [n]$$

and assume priors:

$$(\pi|\eta) \sim \text{Beta}(\eta,\eta)$$

where $\pi|\eta$ is a symmetric distribution center about $\pi = \frac{1}{2}$,

$$(\tau_k|\alpha_k,\beta_k) \sim \text{Gamma}(\alpha_k,\beta_k)$$

where $\tau_k = \frac{1}{\sigma_k^2}$ is the precision and β_k is the rate parameter for a gamma distribution,

$$(\mu_k|\tau_k,\mu_{0k},\nu_k) \sim \text{N}\left(\mu_{0k},(\nu_k\tau_k)^{-1}\right)$$

where μ_{0k} is the prior mean for μ_k and ν_k is the number of pseudo-observations used to estimate μ_k. Note that it is assumed that the joint prior for μ_k and τ_k is

$$p(\mu_k,\tau_k) = p(\mu_k|\tau_k)\cdot p(\tau_k).$$

Finally, the forms of these prior distributions are given explicitly:

$$p(\pi|\eta) = \frac{\Gamma(2\eta)}{\Gamma(\eta)\Gamma(\eta)}\pi^{\eta-1}(1-\pi)^{\eta-1}, \pi \in [0,1], \eta \in \mathbb{R}^+.$$

$$p(\tau_k|\alpha_k,\beta_k) = \frac{\beta_k^{\alpha_k}}{\Gamma(\alpha_k)}\tau_k^{\alpha_k-1}\exp\left\{-\beta_k\tau_k\right\}, \tau_k,\alpha_k,\beta_k \in \mathbb{R}^+.$$

$$p(\mu_k|\tau_k, \mu_{0k}, \nu_k) = (2\pi)^{-\frac{1}{2}} (\nu_k\tau_k)^{\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mu_k - \mu_{0k})(\nu_k\tau_k)(\mu_k - \mu_{0k})\right\}.$$

The generative model can be represented as a graphical model as shown in 3.5

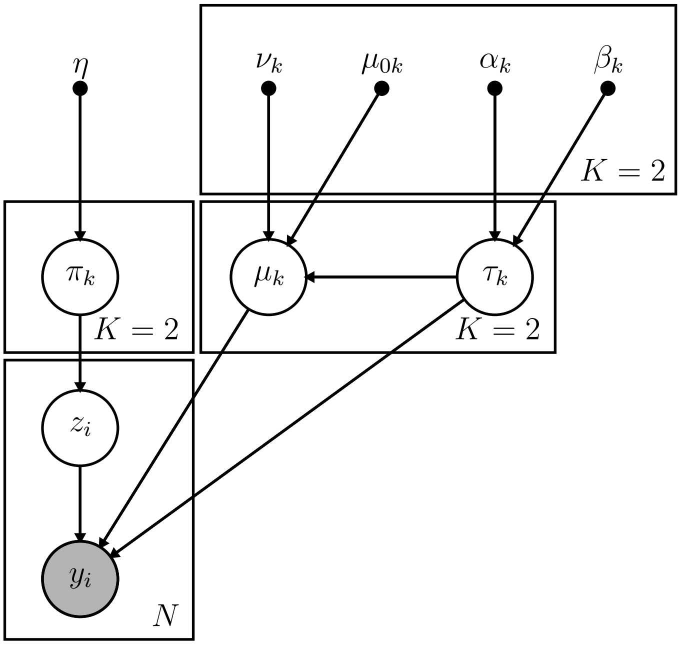

Figure 3.5: Graphical model of a mixture of two Gaussians

In order to implement a Gibbs sampler to obtain the joint posterior distribution $p(\theta, z|y)$, the full conditional distributions of θ and z_i are needed. First, the full conditional distribution of π is derived. Since $z \sim$ Bernoulli(π) and $\pi \sim$ Beta(η, η), conjugacy will yield a posterior where $(\pi|z) \sim$ Beta($\cdot, \cdot$).

$$\begin{aligned} p(\pi|z_i) &\propto p(z_i|\pi) \cdot p(\pi|\eta) \\ &= \pi^{z_i}(1-\pi)^{1-z_i} \cdot \pi^{\eta-1}(1-\pi)^{\eta-1} \\ &= \pi^{z_i+\eta-1}(1-\pi)^{-z_i+\eta}. \end{aligned}$$

It can be seen that the full conditional distribution of π is

$$(\pi|z_i) \sim \text{Beta}(z_i + \eta, -z_i + \eta + 1).$$

Extending the result for $\pi|z_i$ to $\pi|z$,

$$\begin{aligned} p(\pi|z) &\propto p(\pi|\eta) \cdot \prod_{i=1}^{n} p(z_i|\pi) \\ &= \pi^{\eta-1}(1-\pi)^{\eta-1}\pi^{\sum_{i=1}^{n} z_i}(1-\pi)^{n-\sum_{i=1}^{n} z_i} \\ &= \pi^{n_2+\eta-1}(1-\pi)^{n_1+\eta-1}, \end{aligned}$$

which yields

$$(\pi|z) \sim \text{Beta}(n_2 + \eta, n_1 + \eta).$$

Next, the full conditional distribution of z_i is derived:

$$\begin{aligned} p(z_i|y_i, \theta) &\propto p(y_i|z_i, \theta) \cdot p(z_i|\pi) \\ &= \left[\phi\left(y; \mu_1, \tau_1^{-1}\right)\right]^{1-z_i} \left[\phi\left(y; \mu_2, \tau_2^{-1}\right)\right]^{z_i} \cdot \pi^{z_i}(1-\pi)^{1-z_i} \\ p(z_i|y_i, \theta) &\propto \left[(1-\pi)\phi\left(y; \mu_1, \tau_1^{-1}\right)\right]^{1-z_i} \left[\pi\phi\left(y; \mu_2, \tau_2^{-1}\right)\right]^{z_i}. \end{aligned}$$

Next, the full conditional distribution of τ_k is derived. Since $y_{n_k} \sim \text{N}(\mu_k, \tau_k)$ and $\tau_k \sim \text{Gamma}(\alpha_k, \beta_k)$, conjugacy will yield a posterior where $(\tau_k|y_{n_k}) \sim \text{Gamma}(\cdot, \cdot)$:

$$\begin{aligned} p(\tau_k|y_{n_k}, z_{n_k}) &\propto \prod_{i=1}^{n_k} p(y_i|z_i, \tau_k) \cdot p(\tau_k|\alpha_k, \beta_k) \\ &= \tau_k^{\frac{n_k}{2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{n_k}(y_i - \mu_k)^2 \tau_k\right\} \cdot \tau_k^{\alpha_k - 1} \exp\left\{-\beta_k \tau_k\right\} \\ &= \tau_k^{\frac{n_k}{2} + \alpha_k - 1} \exp\left\{-\left[\frac{1}{2}\sum_{i=1}^{n_k}(y_i - \mu_k)^2 + \beta_k\right]\tau_k\right\} \end{aligned}$$

$$(\tau_k|y_{n_k}, z_{n_k}) \sim \text{Gamma}\left(\alpha_k + \frac{n_k}{2}, \beta_k + \frac{1}{2}\sum_{i=1}^{n_k}(y_i - \mu_k)^2\right)$$

Finally, the full conditional distribution of μ_k is derived. Since $y_{n_k} \sim \text{N}(\mu_k, \tau_k)$ and $\mu_k \sim \text{N}(\mu_{0k}, (\nu_k\tau_k)^{-1})$, conjugacy will yield a posterior where $(\mu_k|y_{n_k}, \tau_k) \sim \text{N}(\cdot, \cdot)$.

$$\begin{aligned}
p(\mu_k|\tau_k, y_{n_k}, z_{n_k}) &\propto \prod_{i=1}^{n_k} p(y_i|z_i, \mu_k, \tau_k) \cdot p(\mu_k|\tau_k, \nu_k) \\
&= \exp\left\{-\frac{1}{2}\sum_{i=1}^{n_k}(y_i - \mu_k^2\tau_k)\right\} \cdot \phi\left(\mu_k; \mu_{0k}, (\nu_k\tau_k)^{-1}\right) \\
&= \exp\left\{-\frac{1}{2}\sum_{i=1}^{n_k} y_i\tau_k y_i - 2\mu_k\tau_k\sum_{i=1}^{n_k} y_i + \mu_k n_k \tau_k \mu_k\right\} \cdot \phi\left(\mu_k; \mu_{0k}, (\nu_k\tau_k)^{-1}\right) \\
&= \phi\left(\mu_k; \bar{y}_{n_k}, (n_k\tau_k)^{-1}\right) \cdot \phi\left(\mu_k; \mu_{0k}, (\nu_k\tau_k)^{-1}\right) \\
&= \phi\left(\mu_k; \frac{\frac{\bar{y}_{n_k}}{\nu_k\tau_k} + \frac{\mu_{0k}}{n_k\tau_k}}{\frac{1}{n_k\tau_k} + \frac{1}{\nu_k\tau_k}}, \frac{\frac{1}{n_k\nu_k\tau_k^2}}{\frac{1}{n_k\tau_k} + \frac{1}{\nu_k\tau_k}}\right) \\
&= \phi\left(\mu_k; \frac{n_k\bar{y}_{n_k} + \nu_k\mu_{0k}}{\nu_k + n_k}, \frac{1}{\tau_k} \cdot \frac{1}{\nu_k + n_k}\right)
\end{aligned}$$

$$(\mu_k|\tau_k, y_{n_k}, z_{n_k}) \sim \text{N}\left(\frac{n_k\bar{y}_{n_k} + \nu_k\mu_{0k}}{\nu_k + n_k}, \frac{1}{\tau_k} \cdot \frac{1}{\nu_k + n_k}\right)$$

Using these full conditional distributions, a Gibbs sampling algorithm for this model is:

```
**Initialize** $\pi_k, \mu_k, \tau_k,$;
**for** $t = 0$ *to* $T-1$ **do**
    **for** $i = 1$ *to* $n$ **do**
        sample $z_i^{(t+1)} \sim \left[(1-\pi^{(t)})\phi\left(y; \mu_1^{(t)}, \frac{1}{\tau_1^{(t)}}\right)\right]^{1-z_i^{(t)}} \left[\pi^{(t)}\phi\left(y; \mu_2^{(t)}, \frac{1}{\tau_2^{(t)}}\right)\right]^{z_i^{(t)}}$;
    **end**
    update group sizes $n_1^{(t+1)}$ and $n_2^{(t+1)}$ and group means $\bar{y}_1^{(t+1)}$ and $\bar{y}_2^{(t+1)}$;
    sample $\pi^{(t+1)} \sim \text{Beta}(n_2^{(t+1)} + \eta, n_1^{(t+1)} + \eta)$;
    sample $\tau_k^{(t+1)} \sim \text{Gamma}\left(\alpha_k + \frac{n_k^{(t+1)}}{2}, \beta_k + \frac{1}{2}\sum_{i=1}^{n_k^{(t+1)}}\left(y_i - \mu_k^{(t)}\right)^2\right)$;
    sample $\mu_k^{(t+1)} \sim \text{N}\left(\frac{n_k^{(t+1)}\bar{y}_{n_k}^{(t+1)} + \nu_k\mu_{0k}}{\nu_k + n_k^{(t+1)}}, \frac{1}{\tau_k^{(t+1)}} \cdot \frac{1}{\nu_k + n_k^{(t+1)}}\right)$;
**end**
```

Figure 3.6: Gibbs sampler for mixture of two Gaussians

Data were generated by drawing 500 samples from $\text{N}(\mu_1 = -10, \sigma^2 = 9)$ and 500 samples from $\text{N}(\mu_2 = 10, \sigma^2 = 9)$. Assuming that $p(y;\theta) = 0.5 \cdot \phi\left(y; -10, 3^2\right) + 0.5 \cdot \phi\left(y; 10, 3^2\right)$, a sample of size $n = 200$ is generated after an initial burn-in period of 3000 iterations and thinning every 15 samples to decorrelate the Markov chain. The following initializations were used: $\mu_{01} = -5, \mu_{02} = 5, \alpha_1 = \alpha_2 = 1, \beta_1 = \beta_2 = 5, \eta = 0.5, \nu_1 = \nu_2 = 50$.

The posterior distributions are shown in Figure 3.7. Note that all five posterior distributions are relatively symmetric and unimodal. Maximum a posteriori estimates using the posterior means are: $\pi = 0.4991$, $\mu 1 = -9.4976$, $\tau_1 = 0.1139$, $\mu_2 = 9.5430$, $\tau_2 = 0.1133$. Finally, inference on θ can be performed using 95% Bayesian credible interval: π: $(0.4688, 0.5293)$; $\mu 1$: $(-9.7475, -9.2362)$; τ_1: $(0.0989, 0.1303)$; $\mu 2$: $(9.3139, 9.7757)$; τ_1: $(0.1013, 0.1252)$. While the Gibbs sampler is more complex than Algorithm 3.3, it successfully approximated the joint posterior of all five parameters.

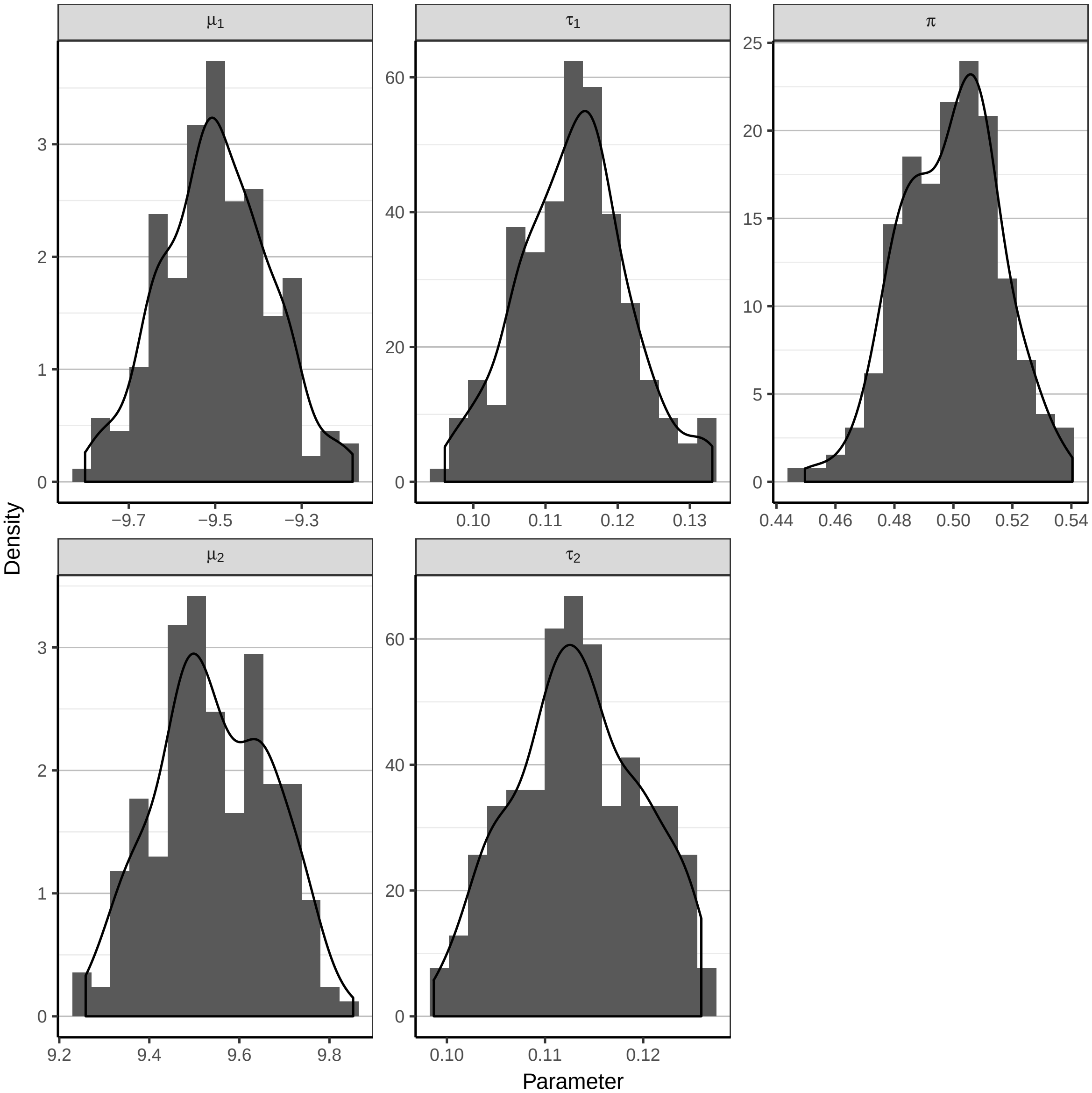

Figure 3.7: Empirical posterior distributions for a mixture of two Gaussians.

Chapter 4

Hidden Markov Models

4.1 The Generative Hidden Markov Model

The mixture model discussed in the previous chapter assumed that the latent variables were mutually independent, an assumption that eases computation significantly. While convenient, this assumption is not always reasonable. It is often more realistic in applications such as speech processing (e.g. Levinson, Rabiner, and Sondhi 1983) to abandon probabilistic models that rely on stationary distributions and instead attempt to model the non-stationary nature of data directly. This is particularly useful in speech and text applications since language is inherently temporal.

A hidden Markov model (HMM) assumes that randomly observed variables are generated by an unobserved finite-state Markov chain. The observed random variables are assumed to be generated by a different distribution for each latent state. More formally, a hidden Markov model consists of a discrete-time process $\{(X_t, Y_t)\}, t = 0, 1, \ldots, T-1$ where X_t denotes the state of the latent Markov chain at time t, Y_t denotes the observed value at time t, and T values are observed. It is assumed that Y_t depends on X_t, but that Y_t is independent of all other latent values $X_j, j \neq t$. While it is theoretically possible to consider any configuration of dependency in the latent space, it is most common for X_t to depend on X_{t-1}. The distribution of $Y_t|X_t$ can be discrete or continuous. For simplicity, only derivations for the discrete case are considered. However, extensions to the continuous case are straightforward. HMMs were first proposed in the literature in the 1960s (e.g. Baum and Petrie 1966) while the EM algorithm for HMMs was proposed by Baum et al. (1970).

First, consider the discrete observations case. Suppose that there are N possible latent states, $X \in Q = \{q_0, q_1, \ldots, q_{N-1}\}$ and M possible observed values, $Y \in \{0, 1, \ldots, M-1\}$. The Markov chain transitions from time t to time $t+1$ according to the $N \times N$ transition probability matrix

$$A = \{a_{ij}\}$$

where $a_{ij} = p(x_{t+1} = q_j | x_t = q_i)$ and A is row stochastic such that $\sum_j a_{ij} = 1, i = 1, \ldots, N$. The initial state probabilities are

$$\pi_{x_0} = \{p(x_0 = q_j)\}$$

where $\sum_j \pi_j = 1$. Observations are generated according to the $N \times M$ emission matrix

$$B = \{b_j(k)\}$$

where $b_j(k) = p(y_t = k | x_t = q_j)$, B is row stochastic, and $k \in \{0, \ldots, M-1\}$.

The model $\lambda = (\pi, A, B)$ fully define a discrete HMM and presents three distinct problems. First, it is of interest to determine the likelihood of a particular observed sequence $p(Y|\lambda)$. Second, one may seek to estimate the latent state sequence $\{X\}$ that generated the observed sequence $\{Y\}$. This problem can be solved using either the Viterbi algorithm (Viterbi 1967) – which seeks the sequence $\{X^*\}$ which maximizes $p(X|Y, \lambda)$ – or the forward-backward algorithm (Baum et al. 1970). Finally, estimation of the model λ can be performed by seeking the optimal model λ^* which maximizes the likelihood $p(Y|\lambda)$.

4.2 The Likelihood of the Discrete Hidden Markov Model

Our main concern is in optimizing λ by maximizing the likelihood, so first consider the process of obtaining the likelihood $p(Y|\lambda)$ where $Y = \{y_0, y_1, \dots, y_{T-1}\}$ is the observed sequence generated by the state sequence $X = \{x_0, x_1, \dots, x_{T-1}\}$.

Since the observations are independent given x_t,

$$p(Y|X,\lambda) = \prod_{t=0}^{T} p(y_t|x_t,\lambda).$$

By definition of the emission matrix B,

$$p(Y|X,\lambda) = b_{x_0}(y_0) \cdot b_{x_1}(y_1) \cdots b_{x_{T-1}}(y_{T-1}).$$

Next, note that using the definition of conditional probability,

$$p(Y|X,\lambda) = \frac{p(Y,X|\lambda)}{p(X|\lambda)}$$

can be rearranged to obtain

$$p(Y,X|\lambda) = p(Y|X,\lambda) \cdot p(X|\lambda).$$

Using the definitions of π and A,

$$\begin{aligned} p(X|\lambda) &= p(x_0|\lambda) \cdot p(x_1|x_0,\lambda) \cdots p(x_{T-1}|x_{T-2},\lambda) \\ &= \pi_{x_0} \cdot a_{x_0,x_1} \cdots a_{x_{T-2},x_{T-1}}. \end{aligned}$$

Therefore,

$$\begin{aligned} p(Y,X|\lambda) &= p(Y|X,\lambda) \cdot p(X|\lambda) \\ &= b_{x_0}(y_0) \cdot b_{x_1}(y_1) \cdots b_{x_{T-1}}(y_{T-1}) \cdot \pi_{x_0} \cdot a_{x_0,x_1} \cdots a_{x_{T-2},x_{T-1}}. \end{aligned} \tag{4.1}$$

Summing over the latent space,

$$
\begin{aligned}
p(Y|\lambda) &= \sum_{X \in \mathcal{X}} p(Y|X,\lambda) \cdot p(X|\lambda) \\
&= \sum_{X \in \mathcal{X}} b_{x_0}(y_0) \cdot b_{x_1}(y_1) \cdots b_{x_{T-1}}(y_{T-1}) \cdot \pi_{x_0} \cdot a_{x_0,x_1} \cdots a_{x_{T-2},x_{T-1}} \\
&= \sum_{X \in \mathcal{X}} \pi_{x_0} \cdot b_{x_0}(y_0) a_{x_0,x_1} \cdot b_{x_1}(y_1) \cdots a_{x_{T-2},x_{T-1}} b_{x_{T-1}}(y_{T-1}).
\end{aligned}
$$

Unfortunately, computing the likelihood in this manner requires approximately $2TN^T$ multiplications (Rabiner 1989). This becomes computationally prohibitive when the number of observations T and states N grows large.

4.3 The Forward-Backward Algorithm

One attractive feature of the HMM is the forward-backward algorithm of Baum et al. (1970) which makes the computation of the likelihood and the most probable latent state sequence much faster. Next, the forward algorithm (one half of the forward-backward algorithm) for determining the likelihood $p(Y|\lambda)$ is defined. Define $\alpha_t(i) = p(y_0, y_1, \cdots, y_t, x_t = q_i|\lambda)$ for $t = 0, 1, \cdots, T-1$ and $i = 0, 1, \cdots, N-1$ as the joint probability of the observation sequence up to time t and the latent state at time t. Let $\alpha_t(i)$ denote the forward variable for time t. $\alpha_t(i)$ can be obtained inductively:

1. $\alpha_0(i) = \pi_i \cdot b_i(y_0), i = 0, 1, \cdots, N-1$
2. $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(j) a_{ij}\right] b_j(y_t), 1 \leq t \leq T-1, 0 \leq, j \leq N-1$
3. $p(Y|\lambda) = \sum_{i=0}^{N-1} \alpha_{T-1}(i).$

An algorithm for computing the forward variables is provided in Figure 4.1.

```
/* Given $A$, $B$, $\pi$                                    */
$c_0 = 0$ ;
for $i = 0$ to $N-1$ do
    $\alpha_0(i) = \pi_i b_i(y_0)$ ;
    $c_0 = c_0 + \alpha_0(i)$ ;
end
$c_0 = \frac{1}{c_0}$ ;
for $i = 0$ to $N-1$ do
    $\alpha_0(i) = c_0 \alpha_0(i)$ ;
end
for $t = 1$ to $T-1$ do
    $c_t = 0$ ;
    for $i = 0$ to $N-1$ do
        $\alpha_t(i) = 0$ ;
        for $j = 0$ to $N-1$ do
            $\alpha_t(i) = \alpha_t(i) + \alpha_{t-1}(j) a_{ji}$ ;
        end
        $\alpha_t(i) = \alpha_t(i) b_i(y_t)$ ;
        $c_t = c_t + \alpha_t(i)$ ;
    end
    $c_t = \frac{1}{c_t}$ ;
    for $i = 0$ to $N-1$ do
        $\alpha_t(i) = c_t \alpha_t(i)$ ;
    end
end
```

Figure 4.1: Forward algorithm

Instead of the naive direct computation of the likelihood's required $2TN^T$ multiplications, the forward algorithm only involves approximately N^2T multiplications. While this will still take longer to compute as N and T increase, it is much faster.

The second half of the forward-backward algorithm yields the backward variable $\beta_t(i)$ which is defined as

$$\beta(i) = p(y_{t+1}, y_{t+2}, \ldots, y_{T-1} | x_t = q_i, \lambda).$$

Like the forward variables, the backward variables can be computed inductively:

1. $\beta_{T-1}(i) = 1, i = 0, 1, \ldots, N-1$

2. $\beta_t(i) = \sum_{j=0}^{N-1} a_{ij} b_j(y_{t+1}) \beta_{t+1}(j), t = T-2, T-3, \ldots, 0, i = 0, 1, \ldots, N-1.$

An algorithm for computing the backward variables is given in Figure 4.2.

```
/* Given $A$, $B$, $\pi$ */
for $i = 0$ to $N - 1$ do
    $\beta_{T-1}(i) = c_{T-1}$ ;
end
for $t = T - 2$ to $0$ do
    for $i = 0$ to $N - 1$ do
        $\beta_t(i) = 0$ ;
        for $j = 0$ to $N - 1$ do
            $\beta_t(i) = \beta_t(i) + a_{ij} b_j(y_{t+1} \beta_{t+1}(j)$ ;
        end
        $\beta_t(i) = c_t \beta_t(i)$ ;
    end
end
```

Figure 4.2: Backward algorithm

4.4 The Expectation-Maximization Algorithm for the Discrete Hidden Markov Model

Equipped with the forward algorithm of Baum et al. (1970), it is feasible to compute the likelihood $p(Y|\lambda)$ and therefore obtain a maximum likelihood estimator (MLE) of λ. Since there is no analytical solution for $\hat{\lambda}_{MLE}$, the EM algorithm is used to iteratively obtain $\hat{\lambda}_{MLE}$ corresponding to a (local) maximum of the likelihood function.

The rationale for this technique stems from Baum et al.'s proof using Jensen's inequality to show that the EM updates of the parameters maximize

$$Q(\lambda, \lambda^s) = \mathbb{E}_{x \in \mathcal{X}} \log p(Y, X|\lambda)$$

and that this is guaranteed to increase the likelihood since

$$\max_{\lambda^s} \{Q(\lambda, \lambda^s)\} \to p(Y|\lambda^s) \geq p(Y|\lambda).$$

Generally, the expectation step of the EM algorithm is

$$\begin{aligned} Q(\lambda, \lambda^s) &= \mathbb{E}_{x \in \mathcal{X}} \log p(Y, X|\lambda) \\ &= \sum_{x \in \mathcal{X}} \log\left[p(Y, X|\lambda)\right] \cdot p(X|\lambda^s). \end{aligned}$$

The maximization step is

$$\lambda^{s+1} = \arg\max_{\lambda} \left\{Q(\lambda, \lambda^s)\right\}.$$

The expectation and maximization steps are repeated until convergence. It is worth noting for convenience that it is equivalent to maximize

$$\sum_{x \in \mathcal{X}} \log\left[p(Y, X|\lambda)\right] \cdot p(X, Y|\lambda^s)$$

instead of

$$\sum_{x \in \mathcal{X}} \log\left[p(Y, X|\lambda)\right] \cdot p(X|\lambda^s).$$

Using Equation 4.1, the expectation step can be rewritten:

$$\begin{aligned} \hat{Q}(\lambda, \lambda^s) &= \sum_{x \in \mathcal{X}} \log\left[p(Y, X|\lambda)\right] \cdot p(X|\lambda^s) \\ &= \sum_{x \in \mathcal{X}} \log\left[\pi_{x_0} \cdot b_{x_0}(y_0) a_{x_0,x_1} \cdot \prod_{t=1}^{T-1} b_{x_t}(y_t)\right] \cdot p(X, Y|\lambda^s) \\ &= \sum_{x \in \mathcal{X}} \log\left[\pi_{x_0}\right] \cdot p(X, Y|\lambda^s) + \sum_{x \in \mathcal{X}} \sum_{t=1}^{T-1} \log\left[a_{x_{t-1},x_t}\right] \cdot p(X, Y|\lambda^s) + \\ & \sum_{x \in \mathcal{X}} \sum_{t=0}^{T-1} \log\left[b_{x_t}(y_t)\right] \cdot p(X, Y|\lambda^s). \end{aligned} \tag{4.2}$$

Since the rows of A and B and the vector π are stochastic, constraints must be added to 4.2 before maximizing with respect to π_i, a_{ij}, and $b_i(j)$. Therefore, define

$$\hat{L}(\lambda, \lambda^s) = \hat{Q}(\lambda, \lambda^s) - \gamma_\pi \left(\sum_{i=1}^{N} \pi_i - 1 \right) - \sum_{i=0}^{N-1} \gamma_{a_i} \left(\sum_{j=0}^{N-1} \gamma_{ij} - 1 \right) - \sum_{i=0}^{N-1} \gamma_{b_i} \left(\sum_{j=0}^{M-1} b_i(j) - 1 \right). \tag{4.3}$$

First, maximize Equation 4.3 with respect to π_i:

$$\begin{aligned} \frac{\delta \hat{L}(\lambda, \lambda^s)}{\delta \pi_i} &= \frac{\delta}{\delta \pi_i} \left(\sum_{x \in \mathcal{X}} \log [\pi_{x_0}] \cdot p(X, Y|\lambda^s) \right) - \gamma_\pi & &= 0 \\ &= \frac{\delta}{\delta \pi_i} \left(\sum_{j=0}^{N-1} \log [\pi_j] \cdot p(x_0 = j, Y|\lambda^s) \right) - \gamma_\pi & &= 0 \\ &= \frac{p(x_0 = i, Y)|\lambda^s)}{\pi_i} - \gamma_\pi & &= 0. \end{aligned} \tag{4.4}$$

Rearranging Equation 4.4

$$\pi_i = \frac{p(x_0 = i, Y)|\lambda^s)}{\gamma_\pi}$$

and using the constraint on π

$$\sum_{i=0}^{N-1} \pi_i = \frac{1}{\gamma_\pi} \sum_{i=0}^{N-1} p(x_0 = i, Y)|\lambda^s) = 1$$

yields

$$\gamma_\pi = p(Y|\lambda^s)$$

and our estimate for π_i is

$$\begin{aligned} \hat{\pi_i}^{(s+1)} &= \frac{p(x_0 = i, Y)|\lambda^s)}{p(Y|\lambda^s)} \\ &= p(x_0 = i|Y, \lambda^s). \end{aligned} \tag{4.5}$$

Using the forward and backward variables, Equation 4.5 can be rewritten:

$$\begin{aligned}\hat{\pi}_i{}^{(s+1)} &= p(x_0 = i|Y, \lambda^s)\\ &= \frac{\alpha_0(i)\beta_0(i)}{\sum_{i=0}^{N-1}\alpha_{T-1}(i)}.\end{aligned}$$

Defining

$$\begin{aligned}\omega(x_t = i) &= p(x_t|Y, \lambda)\\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=0}^{N-1}\alpha_{T-1}(i)},\end{aligned}$$

an alternative form of $\hat{\pi}_i{}^{(s+1)}$ is

$$\hat{\pi}_i{}^{(s+1)} = \omega(x_0 = i)$$

which is used later in the implementation of the EM algorithm.

Next, maximize Equation 4.3 with respect to a_{ij}:

$$\begin{aligned}\frac{\delta\hat{L}(\lambda, \lambda^s)}{\delta a_{ij}} &= \frac{\delta}{\delta a_{ij}}\left(\sum_{x\in\mathcal{X}}\sum_{t=1}^{T-1}\log\left[a_{x_{t-1},x_t}\right]\cdot p(X, Y|\lambda^s)\right) - \gamma_{a_i} &&= 0\\ &= \frac{\delta}{\delta a_{ij}}\left(\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}\sum_{t=1}^{T-1}\log\left[a_{jk}\right]p(x_{t-1} = j, x_t = k, Y|\lambda^s)\right) - \gamma_{a_i} &&= 0\\ &= \frac{1}{a_{ij}}\sum_{t=1}^{T-1}p(x_{t-1} = i, x_t = j, Y|\lambda^s) - \gamma_{a_i} &&= 0. \qquad (4.6)\end{aligned}$$

Rearranging Equation 4.6

$$a_{ij} = \frac{1}{\gamma_{a_i}}\sum_{t=1}^{T-1}p(x_{t-1} = i, x_t = j, Y|\lambda^s)$$

and using the constraint on a_{ij}

$$\sum_{j=0}^{N-1}a_{ij} = 1$$

yields

$$\begin{aligned}\gamma_{a_i} &= \sum_{j=0}^{N-1}\sum_{t=1}^{T-1} p(x_{t-1}=i, x_t=j, Y|\lambda^s)\\ &= \sum_{t=1}^{T-1} p(x_{t-1}=i, Y|\lambda^s)\end{aligned}$$

and our estimate for a_{ij} is

$$\begin{aligned}\hat{a_{ij}}^{(s+1)} &= \frac{\sum_{t=1}^{T-1} p(x_{t-1}=i, x_t=j, Y|\lambda^s)}{\sum_{t=1}^{T-1} p(x_{t-1}=i, Y|\lambda^s)}\\ &= \frac{\sum_{t=1}^{T-1} p(x_{t-1}=i, x_t=j|Y, \lambda^s)\cdot p(Y|\lambda^s)}{\sum_{t=1}^{T-1} p(x_{t-1}=i|Y, \lambda^s)\cdot p(Y|\lambda^s)}\\ &= \frac{\sum_{t=1}^{T-1} p(x_{t-1}=i, x_t=j|Y, \lambda^s)}{\sum_{t=1}^{T-1} p(x_{t-1}=i|Y, \lambda^s)}. \qquad (4.7)\end{aligned}$$

Defining

$$\begin{aligned}\omega(x_t=i, x_{t+1}=j) &= p(x_t=i, x_{t+1}=j|Y, \lambda)\\ &= \frac{\alpha_t(i)a_{ij}b_j(y_{t+1})\beta_{t+1}(j)}{\sum_{i=0}^{N-1}\alpha_{T-1}(i)},\end{aligned}$$

an alternative form of $\hat{a_{ij}}^{(s+1)}$ is

$$\hat{a_{ij}}^{(s+1)} = \frac{\sum_{t=0}^{T-2}\omega(x_t=i, x_{t+1}=j)}{\sum_{t=0}^{T-2}\omega(x_t=i)}$$

which is used later in the implementation of the EM algorithm.

An algorithm for efficiently computing $\omega_t(i,j)$ and $\omega_t(i)$ is provided in Figure 4.3.

```
/* Given $\alpha$, $\beta$, $A$, $B$, $\pi$                                      */
for $t = 0$ to $T - 2$ do
    denom $= 0$ ;
    for $i = 0$ to $N - 1$ do
        for $j = 0$ to $N - 1$ do
            denom $=$ denom $+ \alpha_t(i) a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)$ ;
        end
    end
    for $i = 0$ to $N - 1$ do
        $\omega_t(i) = 0$ ;
        for $j = 0$ to $N - 1$ do
            $\omega_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(y_{t+1} \beta_{t+1}(j))}{\text{denom}}$ ;
            $\omega_t(i) = \omega_t(i) + \omega_t(i, j)$
        end
    end
end
denom $= 0$ ;
for $i = 0$ to $N - 1$ do
    denom $=$ denom $+ \alpha_{T-1}(i)$ ;
end
for $i = 0$ to $N - 1$ do
    $\omega_{T-1}(i) = \frac{\alpha_{T-1}(i)}{\text{denom}}$ ;
end
```

Figure 4.3: Compute update probabilities

Finally, maximize Equation 4.3 with respect to $b_i(j)$:

$$\begin{aligned}
\frac{\delta \hat{L}(\lambda, \lambda^s)}{\delta b_i(j)} &= \frac{\delta}{\delta b_i(j)} \left(\sum_{x \in \mathcal{X}} \sum_{t=0}^{T-1} \log\left[b_{x_t}(y_t)\right] \cdot p(X, Y | \lambda^s) \right) - \gamma_{b_i} &&= 0 \\
&= \frac{\delta}{\delta b_i(j)} \left(\sum_{i=0}^{N-1} \sum_{t=0}^{T-1} \log\left[b_i(y_t)\right] \cdot p(x_t = i, Y | \lambda^s) \right) - \gamma_{b_i} &&= 0. \qquad (4.8)
\end{aligned}$$

Rearranging Equation 4.8

$$b_i(j) = \frac{1}{\gamma_{b_i}} \sum_{t=0}^{T-1} p(x_t = i, Y | \lambda^s) I(y_t = j)$$

and using the constraint on $b_i(j)$

$$\sum_{j=0}^{M-1} b_i(j) = 1$$

yields

$$\begin{aligned}\gamma_{b_i} &= \sum_{t=0}^{T-1}\sum_{j=0}^{M-1} p(x_t = i, Y|\lambda^s) I(y_t = j) \\ &= \sum_{t=0}^{T-1} p(x_t = i, Y|\lambda^s)\end{aligned}$$

and our estimate for $b_i(j)$ is

$$\begin{aligned}\hat{b_i(j)}^{(s+1)} &= \frac{\sum_{t=0}^{T-1} p(x_t = i, Y|\lambda^s) I(y_t = j)}{\sum_{t=0}^{T-1} p(x_t = i, Y|\lambda^s)} \\ &= \frac{\sum_{t=0}^{T-1} p(x_t = i|Y, \lambda^s) p(Y|\lambda^s) I(y_t = j)}{\sum_{t=0}^{T-1} p(x_t = i|Y, \lambda^s) p(Y|\lambda^s)} \\ &= \frac{\sum_{t=0}^{T-1} p(x_t = i|Y, \lambda^s) I(y_t = j)}{\sum_{t=0}^{T-1} p(x_t = i|Y, \lambda^s)} \\ &= \frac{\sum_{t=0}^{T-1} \omega(x_t = i) I(y_t = j)}{\sum_{t=0}^{T-1} \omega(x_t = i)}. \end{aligned} \tag{4.9}$$

Using the updates given in Equations 4.5, 4.7, 4.9, the full EM algorithm for the discrete HMM is given in Figure 4.4. Note that scaling is introduced to avoid underflow since the product of many probabilities will decrease toward 0 as the number of observations T increases. The scaling constant for observation t c_t is defined as

$$c_t = \frac{1}{\sum_{j=0}^{N-1} \alpha_t(j)}$$

.

```
Initialize $A^{(0)}, B^{(0)}, \pi^{(0)}$ ;
/* Run forward algorithm                                    */
/* Backward algorithm                                       */
/* Compute $\omega_t(i,j)$ and $\omega_t(i)$                */
/* Update $\pi$:                                            */
for $i = 0$ to $N-1$ do
    $\pi_i = \omega_0(i)$ ;
end
/* Update $A$:                                              */
for $i = 0$ to $N-1$ do
    for $j = 0$ to $N-1$ do
        numer = 0 ;
        denom = 0 ;
        for $t = 0$ to $T-2$ do
            numer = numer + $\omega_t(i,j)$ ;
            denom = denom + $\omega_t(i)$ ;
        end
        $a_{ij} = \frac{\text{numer}}{\text{denom}}$ ;
    end
end
/* Update $B$:                                              */
for $i = 0$ to $N-1$ do
    for $j = 0$ to $M-1$ do
        numer = 0 ;
        denom = 0 ;
        for $t = 0$ to $T-1$ do
            if $y_t == j$ then
                numer = numer + $\omega_t(i)$ ;
            else
                denom = denom + $\omega_t(i)$ ;
            end
        end
        $b_i(j) = \frac{\text{numer}}{\text{denom}}$ ;
    end
end
/* Update log-likelihood:                                   */
loglike = 0 ;
for $i = 0$ to $N-1$ do
    loglike = loglike + $\log(c_i)$ ;
end
loglike = $-$loglike ;
```

Figure 4.4: Expectation-Maximization algorithm for discrete hidden Markov model

4.5 An Example Application of the EM Algorithm for a Discrete Hidden Markov Model

Finally, the performance of the EM algorithm for learning the parameters of a discrete HMM is illustrated. Consider $10,000$ observations generated by a HMM $\lambda = (\pi, A, B)$ where

$$\pi = \begin{bmatrix} 0.8 & 0.2 \end{bmatrix},$$

$$A = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix},$$

and

$$B = \begin{bmatrix} 0.1 & 0.2 & 0.7 \\ 0.6 & 0.3 & 0.1 \end{bmatrix}$$

The HMM EM algorithm was initialized with

$$\hat{\pi}^{(0)} = \begin{bmatrix} 0.4934 & 0.5066 \end{bmatrix},$$

$$\hat{A}^{(0)} = \begin{bmatrix} 0.5274 & 0.4725 \\ 0.4826 & 0.5174 \end{bmatrix},$$

and

$$\hat{B}^{(0)} = \begin{bmatrix} 0.3626 & 0.3542 & 0.2832 \\ 0.3356 & 0.3270 & 0.3374 \end{bmatrix},$$

which corresponded to an initial log-likelihood $\ell(\hat{\lambda}^{(0)}) = -1111.2$ and allowed to run for a minimum of 10 iterations before converging in $S = 72$ iterations with a final log-likelihood, $\ell(\hat{\lambda}^{(S)}) = -1034$. Model parameters were estimated as

$$\hat{\pi}^{(S)} = \begin{bmatrix} 1.0000 & 0.0000 \end{bmatrix},$$

$$\hat{A}^{(S)} = \begin{bmatrix} 0.7625 & 0.2375 \\ 0.2752 & 0.7248 \end{bmatrix},$$

and

$$\hat{B}^{(S)} = \begin{bmatrix} 0.1001 & 0.1496 & 0.7503 \\ 0.5848 & 0.3128 & 0.1024 \end{bmatrix}.$$

The EM algorithm's estimate of λ is reasonably close despite the sensitivity of the EM algorithm to the complex set of maxima of the log-likelihood function. It is worth noting in earlier trials that the EM algorithm for this initialization converged in three iterations to a very poor solution that did not differ substantially from the initial values. Enforcing a higher minimum number of iterations or trying different initializations may allow the algorithm to explore beyond poor local maxima. For a review of the discrete hidden Markov model as well as proposed extensions and computational considerations, see Rabiner (1989). For an excellent comparison of the EM algorithm and Gibbs sampling for hidden Markov models, see Rydén (2008).

Chapter 5

Maximum a Posteriori Expectation-Maximization for Estimation of Hidden Topic Markov Models

5.1 The State Space for the Hidden Topic Markov Model

In order to perform estimation of the parameters of the Hidden Topic Markov Model, recall that Gruber, Rosen-Zvi, and Weiss (2007) used the well-known expectation-maximization (EM) algorithm for hidden Markov models (HMM) described in Chapter 4 to perform estimation of the HTMM parameters $(\theta_d, \beta, \epsilon)$, $d \in \{0, \ldots, D-1\}$, of the d-th document. This is possible because, conditioned on θ_d and β, the Hidden Topic Markov model is a form of HMM. Therefore, the EM algorithm can be used to approximate the posterior distribution $p(w_d, z_d, \psi_d, d|\theta_d, \beta, \epsilon)$ and derive either maximum-likelihood or maximum-a-posteriori estimates for $(\theta_d, \beta, \epsilon), \forall d$. In order to do so, the conditional distribution of the latent variables

$(z_{d,t}, \psi_{d,t}), t \in \{0, \ldots, N_d\}$ is needed for each document. It is shown that this distribution can be estimated using the forward-backward algorithm for HMMs for use in the expectation step. With this distribution available, closed form updates for $(\theta_d, \beta, \epsilon), \forall d$ can be derived in the maximization step. It is assumed that hyperparameters α and η are fixed and not estimated.

It is not obvious in Gruber, Rosen-Zvi, and Weiss (2007) that the state space needed for an EM approach is distinctly different than a dual space of both topics z_d and driving variables ψ_d. In fact, considering such a state space would lead to a naive approach in which the transition dynamics of the state space would have $2(K^2 - K)$ free parameters to be estimated: One could consider two transition matrices A_d and A'_d where A_d is a $K \times K$ matrix of transitions from topic $z_{d,t-1} = i$ to $z_{d,t} = j$ when $\psi_{d,t} = 1$ and A'_d is a $K \times K$ matrix of transitions from topic $z_{d,t-1} = i$ to $z_{d,t} = j$ when $\psi_{d,t} = 0$. However, we show that the mechanics of the HTMM model allow for estimation of a much smaller set of K free parameters rather than $K^2 - K$ free parameters. This was originally sketched out in an unpublished and incomplete technical note included in the open source code for HTMM (Gruber and Popat 2007), but the derivation of the forward-backward algorithm and EM algorithm was not provided in that note or the original HTMM paper (Gruber, Rosen-Zvi, and Weiss 2007).

Instead of a dual state space of topics $z_{d,t}$ controlled by $\psi_{d,t}, t \in 0, \ldots, N_d - 1$, define a state space $s_{d,t} = (z_{d,t}, \psi_{d,t})$ at word t such that

$$s_{d,t} = \begin{cases} z_{d,t} = z_{d,t-1}, & \psi_{d,t} = 0 \\ z_{d,t} \sim \text{Multinomial}(\theta_d), & \psi_{d,t} = 1. \end{cases} \tag{5.1}$$

Equation 5.1 can be encoded concisely by defining

$$s_{d,t} = z_{d,t} + K(1 - \psi_{d,t}), \tag{5.2}$$

where $s_{d,t} \in \{0, \ldots, 2K - 1\}$. Clearly, if $\psi_{d,t} = 1$, $s_{d,t} \in \{0, \ldots, K - 1\}$ corresponds to drawing a new topic for $z_{d,t}$. If instead $\psi_{d,t} = 0$, then $s_{d,t} \in \{K, \ldots, 2K-1\}$ corresponds to setting $z_{d,t} = z_{d,t-1}$. As a result, the new state space has $2K$ possible states that encode both the topic $z_{d,t}$ and transition indicator $\psi_{d,t}$ at word t.

The transition matrix of the Markov chain governing the behavior of $s_{d,t}$ is $C_d = \{c_{d,ij}\} = p(s_{d,t} = j | s_{d,t-1} = i; \theta_d, \beta, \epsilon)$ where

$$\begin{cases} \epsilon\theta_{d,j}, & 0 \leq j < K \\ 1-\epsilon, & K \leq j < 2K-1, i \in \{j-K, j\}. \end{cases} \tag{5.3}$$

The transition matrix C_d is as follows,

$$C_d = \left[\begin{array}{cccc|cccc} \vert & \vert & & \vert & 1-\epsilon & & & \\ \epsilon\theta_{d,0} & \epsilon\theta_{d,1} & \cdots & \epsilon\theta_{d,K-1} & & 1-\epsilon & & \\ \vert & \vert & & \vert & & & \ddots & \\ & & & & & & & 1-\epsilon \\ \hline \vert & \vert & & \vert & 1-\epsilon & & & \\ \epsilon\theta_{d,0} & \epsilon\theta_{d,1} & \cdots & \epsilon\theta_{d,K-1} & & 1-\epsilon & & \\ \vert & \vert & & \vert & & & \ddots & \\ & & & & & & & 1-\epsilon \end{array}\right], \tag{5.4}$$

where the Markov chain transitions from $s_{d,t-1} = i$ to $s_{d,t} = j, j \in \{0, \ldots, K-1\}$ with probability $p(s_{d,t} = j | p_s d, t-1) = \epsilon\theta_{d,j}$ regardless of the previous state $s_{d,t-1}$. However, if $j \in \{K, \ldots, 2K-1\}$, a new topic is not drawn so the chain can only transition to one possible state with non-zero probability $p(s_{d,t} = j | s_{d,t-1} = i) = 1-\epsilon, i \in \{j-K, j\}$ where topic $z_{d,t}$ is the same as topic $z_{d,t-1}$ deterministically (i.e., $\psi_{d,t} = 0$) or stochastically (i.e., $\psi_{d,t} = 1$). No other transitions are possible. Therefore, the transition matrix of the Markov chain has $K+1$ parameters in a given document: $(\theta_{d,0}, \ldots, \theta_{d,K-1}, \epsilon)$, only K of which are free parameters due to the constraint that $\sum_{i=0}^{K-1} \theta_{d,i} = 1, d \in \{0, \ldots, D-1\}$. It can be easily confirmed that each row of C_d is row stochastic. As a result, the transition dynamics of the

Markov process governing the topics in a document can be obtained by simply estimating K parameters instead of the $K^2 - K + 1$ required by a naive formulation. Perhaps most interesting is that the transition matrix is learned by performing estimation for the mixture components of a document θ_d and a measure of the dependency in the topic space ϵ.

5.2 The Forward-Backward Algorithm for the Hidden Topic Markov Model

Given the special state space for this model, a straightforward adaptation of the forward-backward algorithm is derived to assist with the E-step in the EM algorithm. First, a prior distribution for the initial state $s_{d,0} \sim \pi_d$ is proposed where

$$\pi_{d,i} = p(s_{d,0} = i; \theta_d) = \begin{cases} \theta_{d,i}, & 0 \leq i < K \\ 0, & K \leq i < 2K. \end{cases} \tag{5.5}$$

In effect, this is equivalent to a simple multinomial distribution with parameter θ_d that initializes the topic for word $w_{d,0}$. Since $\psi_{d,0} = 1, \forall d$ by assumption, no probability is assigned to states where $\psi_{d,0} = 0$.

Emission probabilities for $w_{d,t}|s_{d,t}$ are represented by an emission matrix B,

$$B = \{b_j(k)\} = p(w_{d,t} = k|s_{d,t} = j) \text{ s.t.} \sum_{k=0}^{V-1} b_j(k) = 1 \tag{5.6}$$

where

$$p(w_{d,t}|s_{d,t} = j) = \begin{cases} p'(w_{d,t}|z_{d,t} = j) & 0 \leq j < K \\ p'(w_{d,t}|z_{d,t} = j - K) & K \leq j < 2K. \end{cases} \tag{5.7}$$

From the model, it is assumed that $p(w_{d,t} = k|z_{d,t} = j) = \beta_{j,k}$.

Having fully identified the initial state distribution, transition matrix, and emission matrix for a given document, the standard formulation of the forward and backward variables described in Chapter 4 can be used.

For document d, and parameters $\lambda_d = (\theta_d, \beta, \epsilon)$, we define the forward variable $\alpha_{d,t}(i) = p(w_{d,0}, \ldots, w_{d,t}, s_{d,t} = i; \lambda)$. For $t = 0$,

$$\alpha_{d,0}(i) = p(s_{d,0} = i; \lambda) \cdot p(w_{d,0} | s_{d,0} = i; \lambda)$$

and

$$\begin{aligned}\alpha_{d,t}(j) = \left[\sum_{i=0}^{2K-1} \alpha_{d,t-1}(i) \cdot p(s_{d,t} = j | s_{d,t-1} = i; \lambda)\right] \times \\ p(w_{d,t} | s_{d,t} = j; \lambda), t \in \{1, \ldots, N_d - 1\}.\end{aligned} \tag{5.8}$$

Note that $p(w_{d,0}, \ldots, w_{d,N_d-1}; \lambda) = \sum_{i=0}^{2K-1} \alpha_{d,N_d-1}(i)$.

Define the backward variable $\rho_{d,t}(i) = p(w_{d,t+1}, \ldots, w_{d,N_d-1} | s_{d,t} = i; \lambda)$. For $t = N_d - 1$,

$$\rho_{d,N_d-1}(i) = 1, i \in \{0, \ldots, 2K-1\}$$

and

$$\rho_{d,t}(i) = \sum_{j=0}^{2K-1} c_{d,ij} b_j(w_{d,t+1}) \rho_{d,t+1}(j), t \in \{0, \ldots, N_d - 2\}. \tag{5.9}$$

Equipped with the forward and backward variables, it is possible to use the standard EM estimates for $\pi_{d,i}, c_{d,ij}$, and $b_j(k)$ given in Chapter 4. However, it is of more interest to estimate the parameters of the HTMM $(\theta_d, \beta, \epsilon), \forall d$ to study the structure of the corpus. Furthermore, estimation of the latter set of parameters allows easy estimation of the former HMM parameters. Therefore, the EM algorithm is used to estimate $\lambda = (\theta_d, \beta, \epsilon), \forall d$.

First, the expression of the forward variables $\alpha_{d,t}(i)$ can be simplified for faster computation due to the special structure of the transition matrix C_d. Using Equation 5.8 and the assumed model structure,

$$\alpha_{d,0}(i) = \begin{cases} \theta_{d,i}\beta_{i,w_t}, & i \in \{0,\ldots,K-1\} \\ 0, & i \in \{K,\ldots,2K-1\}. \end{cases} \tag{5.10}$$

$\alpha_{d,t}(j)$ can be further simplified. Recall,

$$\alpha_{d,t}(j) = \left[\sum_{i=0}^{2K-1} \alpha_{d,t-1}(i) \cdot p(s_{d,t} = j | s_{d,t-1} = i; \lambda)\right] \cdot p(w_{d,t} | s_{d,t} = j; \lambda), t \in \{1,\ldots,N_d - 1\} \tag{5.11}$$

and that

$$c_{d,ij} = \begin{cases} \epsilon\theta_{d,j}, & 0 \le j < K \\ 1-\epsilon, & K \le j < 2K, i \in \{j-K, j\} \\ 0, & \text{otherwise.} \end{cases} \tag{5.12}$$

Therefore,

$$\begin{aligned} \alpha_{d,t}(j) &= \begin{cases} \left[\sum_{i=0}^{2K-1} \alpha_{d,t-1}(i) \cdot \epsilon\theta_{d,j}\right] \cdot \beta_{j,w_{d,t}}, & 0 \le j < K \\ \left[\sum_{i=0}^{2K-1} \alpha_{d,t-1}(i) \cdot (1-\epsilon) I(i \in \{j-K, j\})\right] \cdot \beta_{j-K,w_{d,t}}, & K \le j < 2K \end{cases} \\ &= \begin{cases} \left[\sum_{i=0}^{2K-1} \alpha_{d,t-1}(i)\right] \cdot \epsilon\theta_{d,j} \cdot \beta_{j,w_{d,t}}, & 0 \le j < K \\ \left[\alpha_{d,t-1}(j-K) + \alpha_{d,t-1}(j)\right] \cdot (1-\epsilon) \cdot \beta_{j-K,w_{d,t}}, & K \le j < 2K \end{cases} \end{aligned} \tag{5.13}$$

where $t \in \{1,\ldots,N_d - 1\}$

Conveniently, if $\alpha_{d,t-1}$ was normalized such that $\sum_{i=0}^{2K-1} \alpha_{d,t-1}(i) = 1$, then Equation 5.13 simplifies to

$$\alpha_{d,t}(j) = \begin{cases} \epsilon\theta_{d,j}\beta_{j,w_{d,t}}, & 0 \le j < K \\ \left[\alpha_{d,t-1}(j-K) + \alpha_{d,t-1}(j)\right](1-\epsilon)\beta_{j-K,w_{d,t}}, & K \le j < 2K \end{cases} \tag{5.14}$$

and the forward variables for document d can be computed using the algorithm in Figure 5.1.

```
/* Given $\theta_d$, $\beta$, $\epsilon$, $K$                                   */
/* $norm_d$ is a vector of length $N_d - 1$                                   */
$norm_{d,0} = 0$ ;
for $i = 0$ to $K - 1$ do
    $\alpha_{d,0}(i) = \theta_{d,i}\beta_{i,w_{d,t}}$ ;
    $\alpha_{d,0}(i + K) = 0$;
    $norm_{d,0} = norm_{d,0} + \alpha_{d,0}(i) + \alpha_{d,0}(i + K)$ ;
end
for $i = 0$ to $K - 1$ do
    $\alpha_{d,0}(i) = \alpha_{d,0}(i)/norm_{d,0}$ ;
end
for $t = 1$ to $N_d - 1$ do
    $norm_{d,t} = 0$ ;
    if $w_t$ is the beginning of a sentence then
        for $j = 0$ to $K - 1$ do
            $\alpha_{d,t}(j) = \epsilon\theta_{d,j}\beta_{j,w_{d,t}}$ ;
            $\alpha_{d,t}(j + K) = [\alpha_{d,t-1}(j) + \alpha_{d,t-1}(j + K)]\,(1 - \epsilon)\beta_{j,w_{d,t}}$ ;
            $norm_{d,t} = norm_{d,t} + \alpha_{d,t}(j) + \alpha_{d,t}(j + K)$ ;
        end
    end
    else
        for $j = 0$ to $K - 1$ do
            $\alpha_{d,t}(j) = 0$ ;
            $\alpha_{d,t}(j + K) = [\alpha_{d,t-1}(j) + \alpha_{d,t-1}(j + K)]\,\beta_{j,w_{d,t}}$ ;
            $norm_{d,t} = norm_{d,t} + \alpha_{d,t}(j + K)$ ;
        end
    end
    for $i = 0$ to $2K - 1$ do
        $\alpha_{d,t}(i) = \alpha_{d,t}(i)/norm_{d,t}$ ;
    end
end
```

Figure 5.1: Forward algorithm for Hidden Topic Markov Model

Similarly, the computation of the backward variables $\rho_{d,t}(i)$ can be simplified. Recall that

$$\rho_{d,t}(i) = p(w_{d,t+1}, \ldots, w_{d,N_d-1}|s_{d,t} = i; \lambda)$$

and $\rho_{d,N_{d-1}}(i) = 1, \forall i$.

Using the assumed model structure,

$$\begin{aligned}
\rho_{d,t}(i) &= \sum_{j=0}^{2K-1} c_{d,ij} b_j(w_{d,t+1}) \rho_{d,t+1}(j) \\
&= \sum_{j=0}^{K-1} c_{d,ij} b_j(w_{d,t+1}) \rho_{d,t+1}(j) + \\
&\quad \sum_{j=K}^{2K-1} c_{d,ij} b_{j-K}(w_{d,t+1}) \rho_{d,t+1}(j) \\
&= \sum_{j=0}^{K-1} \epsilon \theta_{d,j} \beta_{j,w_{d,t+1}} \rho_{d,t+1}(j) + \\
&\quad \sum_{j=K}^{2K-1} (1-\epsilon) \beta_{j-K,w_{d,t+1}} \rho_{d,t+1}(j) I(i \in \{j-K, j\}). \qquad (5.15)
\end{aligned}$$

Since $\rho_{d,t}(j) = \rho_{d,t}(j+K), j \in \{0, \ldots, K-1\}$,

$$\begin{aligned}
\rho_{d,t}(i) &= \sum_{j=0}^{K-1} \epsilon \theta_{d,j} \beta_{j,w_{d,t+1}} \rho_{d,t+1}(j) + \sum_{j=0}^{K-1} (1-\epsilon) \beta_{j,w_{d,t+1}} \rho_{d,t+1}(j) I(i = j) \\
&= \sum_{j=0}^{K-1} \epsilon \theta_{d,j} \beta_{j,w_{d,t+1}} \rho_{d,t+1}(j) + (1-\epsilon) \beta_{i,w_{d,t+1}} \rho_{d,t+1}(i), \qquad (5.16)
\end{aligned}$$

and the backward variables for document d can be computed using the algorithm in Figure 5.2.

```
/* Given $\theta_d$, $\beta$, $\epsilon$, $K$, $norm_d$                */
for $i = 0$ to $K - 1$ do
    $\rho_{N_d-1}(i) = norm_{d,N_d-1}$ ;
end
for $t = N_d - 2$ to $0$ do
    $d = 0$;
    if $w_{t+1}$ is the beginning of a sentence then
        for $j = 0$ to $K - 1$ do
            $\rho_{d,t}(j) = 0$ ;
            $d = d + \epsilon\theta_{d,j}\beta_{j,w_{d,t+1}}\rho_{d,t+1}(j)$ ;
        end
        for $i = 0$ to $K - 1$ do
            $\rho_{d,t}(i) = d + (1 - \epsilon)\beta_{i,w_{d,t+1}}\rho_{d,t+1}(i)$ ;
            $\rho_{d,t}(i) = \rho_{d,t}(i)/norm_{d,t}$ ;
            $\rho_{d,t}(i + K) = \rho_{d,t}(i)$ ;
        end
    end
    else
        for $j = 0$ to $K - 1$ do
            $\rho_{d,t}(j) = \beta_{j,w_{d,t+1}}\rho_{d,t+1}(j)$ ;
            $\rho_{d,t}(j) = \rho_{d,t}(j)/norm_{d,t}$ ;
            $\rho_{d,t}(j + K) = \rho_{d,t}(j)$ ;
        end
    end
end
```

Figure 5.2: Backward algorithm for Hidden Topic Markov Model

5.3 Maximum A Posteriori Expectation-Maximization

The forward-backward algorithm computes the conditional distribution of the state space of a document $p(s_d|d, w_d; \lambda)$ and allows for the computation of the expectation step in an EM algorithm.

The objective function $R(\lambda, \lambda^{(q)})$ at iteration q is

$$R(\lambda, \lambda^{(q)}) = Q(\lambda, \lambda^{(q)}) + log\left[p(\lambda)\right], \tag{5.17}$$

where $p(\lambda)$ is the prior distribution of λ and

$$\begin{aligned} Q(\lambda, \lambda^{(q)}) &= \mathbb{E}_{s_d|d,w_d,\lambda^{(q)}}\Big[\log\Big[\prod_{d=0}^{D-1} p(w_d, s_d; \lambda^{(q)})\Big]\Big] \\ &= \sum_{s\in\mathcal{S}}\sum_{d=0}^{D-1} \log\big[p(w_d, s_d; \lambda^{(q)})\big] \cdot p(s_d|d, w_d; \lambda^{(q)}). \end{aligned}$$

Next, the log-prior $\log\left[p(\lambda)\right]$ is written using

$$\theta_d \sim \text{Dirichlet}(\alpha), \forall d \in \{0, \ldots, D-1\}, \theta_d \in [0,1]^K, \tag{5.18}$$

$$\beta_j \sim \text{Dirichlet}(\eta), \forall j \in \{0, \ldots, K-1\}, \beta_j \in [0,1]^V, \tag{5.19}$$

where V is the size of the corpus vocabulary.

While Gruber, Rosen-Zvi, and Weiss (2007) did not use a prior distribution for ϵ, it would be straightforward to use a Beta prior for ϵ. This strategy is used for the Gibbs sampler proposed in Chapter 6.

Since $p(\theta_d)$ and $p(\beta_j)$ are assumed independent,

$$\begin{aligned} \log\left[p(\lambda)\right] &= \log\Big[\prod_{d=0}^{D-1} p(\theta_d) \prod_{j=0}^{K-1} p(\beta_j)\Big] \\ &\propto \log\Big[\prod_{d=0}^{D-1}\prod_{j=0}^{K-1} \theta_{d,j}^{\alpha-1} \prod_{j=0}^{K-1}\prod_{k=0}^{V-1} \beta_{j,k}^{\eta-1}\Big]. \end{aligned}$$

For convenience, define

$$\widehat{\log\left[p(\lambda)\right]} = (\alpha-1)\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\log(\theta_{d,j}) + (\eta-1)\sum_{j=0}^{K-1}\sum_{k=0}^{V-1}\log(\beta_{j,k}) \tag{5.20}$$

and note that

$$\arg\max_{\lambda}\left\{R(\lambda,\lambda^{(q)})\right\} = \arg\max_{\lambda}\left\{\hat{R}(\lambda,\lambda^{(q))}\right\}$$

where

$$\hat{R}(\lambda,\lambda^{(q)}) = Q(\lambda,\lambda^{(q)}) + \widehat{\log\left[p(\lambda)\right]}. \tag{5.21}$$

Next, Lagrangian constraints are introduced to Equation 5.21

$$\hat{L}(\lambda,\lambda^{(q)}) = \hat{R}(\lambda,\lambda^{(q)}) - \sum_{d=0}^{D-1}\gamma_{\theta_d}\left(\sum_{i=0}^{K-1}\theta_{d,i} - 1\right) - \sum_{i=0}^{K-1}\gamma_{\beta_i}\left(\sum_{k=0}^{V-1}\beta_{i,k} - 1\right). \tag{5.22}$$

Maximization of $\hat{L}(\lambda,\lambda^{(q)})$ with respect to λ is performed in the M- step. First, we maximize $\hat{L}(\lambda,\lambda^{(q)})$ with respect to $\theta_{d,i}$:

$$\frac{\delta\hat{L}(\lambda,\lambda^{(q)})}{\delta\theta_{d,i}} = \frac{\delta Q(\lambda,\lambda^{(q)})}{\delta\theta_{d,i}} + \frac{\delta\widehat{\log\left[p(\lambda)\right]}}{\delta\theta_{d,i}} - \frac{\delta\left[\sum_{d=0}^{D-1}\gamma_{\theta_d}\left(\sum_{i=0}^{K-1}\theta_{d,i} - 1\right)\right]}{\delta\theta_{d,i}}. \tag{5.23}$$

First,

$$\begin{aligned}
\frac{\delta Q(\lambda,\lambda^{(q)})}{\delta\theta_{d,i}} =& \frac{\delta}{\delta\theta_{d,i}}\left[\sum_{d=0}^{D-1}\sum_{j=0}^{2K-1}\log\left[p(w_d,s_d=j;\lambda^{(q)})\right]\cdot p(s_d=j|d,w_d;\lambda^{(q)})\right] \\
=& \frac{\delta}{\delta\theta_{d,i}}\left[\sum_{d=0}^{D-1}\sum_{j=0}^{2K-1}\log\left[p(s_{d,0}=j;\lambda^{(q)})\right]\cdot p(s_{d,0}=j|d,w_d;\lambda^{(q)})\right]+ \\
& \frac{\delta}{\delta\theta_{d,i}}\left[\sum_{d=0}^{D-1}\sum_{j=0}^{2K-1}\sum_{t=1}^{N_d-1}\log\left[p(s_{d,t}=j|s_{d,t-1};\lambda^{(q)})\right]\cdot p(s_{d,t}=j|d,w_d;\lambda^{(q)})\right]+ \\
& \frac{\delta}{\delta\theta_{d,i}}\left[\sum_{d=0}^{D-1}\sum_{j=0}^{2K-1}\sum_{t=1}^{N_d-1}\log\left[p(w_{d,t}|s_{d,t}=j;\lambda^{(q)})\right]\cdot p(s_{d,t}=j|d,w_d;\lambda^{(q)})\right] \\
=& \frac{\delta}{\delta\theta_{d,i}}\left[\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\log\left[\theta_{d,j}\right]\cdot p(s_{d,0}=j|d,w_d;\lambda^{(q)})\right]+ \\
& \frac{\delta}{\delta\theta_{d,i}}\left[\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\sum_{t=1}^{N_d-1}\log\left[\epsilon\theta_{d,j}\right]\cdot p(s_{d,t}=j|d,w_d;\lambda^{(q)})\right] \\
=& \frac{\delta}{\delta\theta_{d,i}}\left[\sum_{d=0}^{D-1}\log\left[\theta_{d,i}\right]\cdot p(s_{d,0}=i|d,w_d;\lambda^{(q)})\right]+ \\
& \frac{\delta}{\delta\theta_{d,i}}\left[\sum_{d=0}^{D-1}\sum_{t=1}^{N_d-1}\log\left[\epsilon\theta_{d,i}\right]\cdot p(s_{d,t}=i|d,w_d;\lambda^{(q)})\right] \\
\frac{\delta Q(\lambda,\lambda^{(q)})}{\delta\theta_{d,i}} =& \frac{\sum_{t=0}^{N_d-1}p(s_{d,t}=i|d,w_d;\lambda^{(q)})}{\theta_{d,i}}.
\end{aligned} \tag{5.24}$$

Next,

$$\begin{aligned}
\frac{\delta\widehat{\log\left[p(\lambda)\right]}}{\delta\theta_{d,i}} &= \frac{\delta}{\delta\theta_{d,i}}\left[(\alpha-1)\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\log(\theta_{d,j})+(\eta-1)\sum_{j=0}^{K-1}\sum_{k=0}^{V-1}\log(\beta_{j,k})\right] \\
\frac{\delta\widehat{\log\left[p(\lambda)\right]}}{\delta\theta_{d,i}} &= \frac{\alpha-1}{\theta_{d,i}}.
\end{aligned} \tag{5.25}$$

Finally,

$$\frac{\delta\left[\sum_{d=0}^{D-1}\gamma_{\theta_d}\left(\sum_{i=0}^{K-1}\theta_{d,i}-1\right)\right]}{\delta\theta_{d,i}} = \gamma_{\theta_d}. \tag{5.26}$$

Using Equations 5.24, 5.25, and 5.26, Equation 5.23 becomes

$$\frac{\delta \hat{L}(\lambda, \lambda^{(q)})}{\delta \theta_{d,i}} = \frac{\sum_{t=0}^{N_d-1} p(s_{d,t} = i | d, w_d; \lambda^{(q)}) + \alpha - 1}{\theta_{d,i}} - \gamma_{\theta_d}. \tag{5.27}$$

Setting $\frac{\delta \hat{L}(\lambda, \lambda^{(q)})}{\delta \theta_{d,i}} := 0$,

$$\theta_{d,i} = \frac{\sum_{t=0}^{N_d-1} p(s_{d,t} = i | d, w_d; \lambda^{(q)}) + \alpha - 1}{\gamma_{\theta_d}}. \tag{5.28}$$

Using the constraint that $\sum_{i=0}^{K-1} \theta_{d,i} = 1, \forall d \in \{0, \ldots, D-1\}$,

$$\sum_{i=0}^{K-1} \theta_{d,i} = \frac{\sum_{i=0}^{K-1} \sum_{t=0}^{N_d-1} p(s_{d,t} = i | d, w_d; \lambda^{(q)}) + K\alpha - K}{\gamma_{\theta_d}} = 1.$$

Therefore,

$$\gamma_{\theta_d} = \sum_{i=0}^{K-1} \sum_{t=0}^{N_d-1} p(s_{d,t} = i | d, w_d; \lambda^{(q)}) + K\alpha - K. \tag{5.29}$$

The M-step update for $\theta_{d,i}$ is

$$\hat{\theta}_{d,i} = \frac{\sum_{t=0}^{N_d-1} p(s_{d,t} = i | d, w_d; \lambda^{(q)}) + \alpha - 1}{\sum_{i=0}^{K-1} \sum_{t=0}^{N_d-1} p(s_{d,t} = i | d, w_d; \lambda^{(q)}) + K\alpha - K}. \tag{5.30}$$

From this full update, the proportional update given by Gruber, Rosen-Zvi, and Weiss (2007) is easily obtained since

$$\begin{aligned}\hat{\theta}_{d,i} &\propto \sum_{t=0}^{N_d-1} p(s_{d,t} = i | d, w_d; \lambda^{(q)}) + \alpha - 1 \\ &= \sum_{t=0}^{N_d-1} p(z_{d,t} = i, \psi_{d,t} = 1 | d, w_d; \lambda^{(q)}) + \alpha - 1 \end{aligned} \tag{5.31}$$

assuming that $\hat{\theta}_{d,i}$ is normalized. Equation 5.31 reveals that the i-th topic proportions of document d is estimated as the prior modal estimate plus the average number of words in document d that were assigned to topic i.

The M-step update for $\beta_{j,k}$ is derived by maximizing $\hat{L}(\lambda, \lambda^{(q)})$ with respect to $\beta_{j,k}$:

$$\frac{\delta \hat{L}(\lambda, \lambda^{(q)})}{\delta \beta_{j,k}} = \frac{\delta Q(\lambda, \lambda^{(q)})}{\delta \beta_{j,k}} + \frac{\delta \widehat{\log\left[p(\lambda)\right]}}{\delta \beta_{j,k}} - \frac{\delta\left[\sum_{i=0}^{K-1} \gamma_{\beta_i}\left(\sum_{m=0}^{V-1} \beta_{i,m} - 1\right)\right]}{\delta \beta_{j,k}}. \tag{5.32}$$

First,

$$\begin{aligned}
\frac{\delta Q(\lambda, \lambda^{(q)})}{\delta \beta_{j,k}} =& \frac{\delta}{\delta \beta_{j,k}}\left[\sum_{d=0}^{D-1}\sum_{i=0}^{2K-1} \log\left[p(w_d, s_d = i; \lambda^{(q)})\right] \cdot p(s_d = i | d, w_d; \lambda^{(q)})\right] \\
=& \frac{\delta}{\delta \beta_{j,k}}\left[\sum_{d=0}^{D-1}\sum_{i=0}^{2K-1}\sum_{t=0}^{N_d-1} \log\left[p(w_{d,t} | s_{d,t} = i; \lambda^{(q)})\right] \cdot p(s_{d,t} = i | d, w_d; \lambda^{(q)})\right] \\
=& \frac{\delta}{\delta \beta_{j,k}}\left[\sum_{d=0}^{D-1}\sum_{i=0}^{K-1}\sum_{t=0}^{N_d-1} \log\left[\beta_{i,k}^{I(w_{d,t}=k)}\right] \cdot p(s_{d,t} = i | d, w_d; \lambda^{(q)})\right] + \\
& \frac{\delta}{\delta \beta_{j,k}}\left[\sum_{d=0}^{D-1}\sum_{i=K}^{2K-1}\sum_{t=0}^{N_d-1} \log\left[\beta_{i-K,k}^{I(w_{d,t}=k)}\right] \cdot p(s_{d,t} = i | d, w_d; \lambda^{(q)})\right] \\
=& \frac{\delta}{\delta \beta_{j,k}}\left[\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} \log\left[\beta_{j,k}^{I(w_{d,t}=k)}\right] \cdot p(s_{d,t} = j | d, w_d; \lambda^{(q)})\right] + \\
& \frac{\delta}{\delta \beta_{j,k}}\left[\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} \log\left[\beta_{j,k}^{I(w_{d,t}=k)}\right] \cdot p(s_{d,t} = j + K | d, w_d; \lambda^{(q)})\right] \\
\frac{\delta Q(\lambda, \lambda^{(q)})}{\delta \beta_{j,k}} =& \frac{\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} p(s_{d,t} = j | d, w_d; \lambda^{(q)}) I(w_{d,t} = k)}{\beta_{j,k}} + \\
& \frac{\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} p(s_{d,t} = j + K | d, w_d; \lambda^{(q)}) I(w_{d,t} = k)}{\beta_{j,k}}.
\end{aligned} \tag{5.33}$$

Next,

$$\begin{aligned}
\frac{\delta \widehat{\log\left[p(\lambda)\right]}}{\delta \beta_{j,k}} &= \frac{\delta}{\delta \beta_{j,k}}\left[(\alpha - 1)\sum_{d=0}^{D-1}\sum_{i=0}^{K-1} \log(\theta_{d,i}) + (\eta - 1)\sum_{i=0}^{K-1}\sum_{k=0}^{V-1} \log(\beta_{i,k})\right] \\
\frac{\delta \widehat{\log\left[p(\lambda)\right]}}{\delta \beta_{j,k}} &= \frac{\eta - 1}{\beta_{j,k}}.
\end{aligned} \tag{5.34}$$

Finally,

$$\frac{\delta\left[\sum_{i=0}^{K-1}\gamma_{\beta_i}\left(\sum_{m=0}^{V-1}\beta_{i,m}-1\right)\right]}{\delta\beta_{j,k}}=\gamma_{\beta_j}. \tag{5.35}$$

Using Equations 5.33, 5.34, and 5.35, Equation 5.32 becomes

$$\begin{aligned}\frac{\delta\hat{L}(\lambda,\lambda^{(q)})}{\delta\beta_{j,k}}=&\frac{\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1}p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(w_{d,t}=k)}{\beta_{j,k}}+\\&\frac{\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1}p(s_{d,t}=j+K|d,w_d;\lambda^{(q)})I(w_{d,t}=k)}{\beta_{j,k}}+\\&\frac{\eta-1}{\beta_{j,k}}-\gamma_{\beta_j}.\end{aligned} \tag{5.36}$$

Setting $\frac{\delta\hat{L}(\lambda,\lambda^{(q)})}{\delta\beta_{j,k}}:=0$,

$$\begin{aligned}\beta_{j,k}=&\frac{\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1}p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(w_{d,t}=k)}{\gamma_{\beta_j}}+\\&\frac{\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1}p(s_{d,t}=j+K|d,w_d;\lambda^{(q)})I(w_{d,t}=k)}{\gamma_{\beta_j}}+\\&\frac{\eta-1}{\gamma_{\beta_j}}.\end{aligned} \tag{5.37}$$

Using the constraint that $\sum_{m=0}^{V-1}\beta_{j,m}=1,\forall j\in\{0,\ldots,K-1\}$,

$$\begin{aligned}\sum_{m=0}^{V-1}\beta_{j,m}=&\frac{\sum_{m=0}^{V-1}\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1}p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(w_{d,t}=m)}{\gamma_{\beta_j}}+\\&\frac{\sum_{m=0}^{V-1}\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1}p(s_{d,t}=j+K|d,w_d;\lambda^{(q)})I(w_{d,t}=m)}{\gamma_{\beta_j}}+\\&\frac{V\eta-V}{\gamma_{\beta_j}}=1.\end{aligned}$$

Therefore,

$$\begin{aligned}\gamma_{\beta_j} =&\sum_{m=0}^{V-1}\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(w_{d,t}=m)+\\ &\sum_{m=0}^{V-1}\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} p(s_{d,t}=j+K|d,w_d;\lambda^{(q)})I(w_{d,t}=m)+\\ &V\eta - V. \end{aligned} \tag{5.38}$$

The M-step update for $\beta_{j,k}$ is

$$\begin{aligned}\hat{\beta}_{j,k} =&\frac{\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(w_{d,t}=k)}{\gamma_{\beta_j}}+\\ &\frac{\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} p(s_{d,t}=j+K|d,w_d;\lambda^{(q)})I(w_{d,t}=k)}{\gamma_{\beta_j}}+\\ &\frac{\eta-1}{\gamma_{\beta_j}}. \end{aligned} \tag{5.39}$$

From this full update, the proportional update given by Gruber, Rosen-Zvi, and Weiss (2007) is easily obtained since

$$\begin{aligned}\hat{\beta}_{j,k} \propto&\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(w_{d,t}=k)+\\ &\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} p(s_{d,t}=j+K|d,w_d;\lambda^{(q)})I(w_{d,t}=k)+\\ &\eta-1\\ =&\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1} p(z_{d,t}=j,w_{d,t}=k|d,w_d;\lambda^{(q)})+\eta-1 \end{aligned} \tag{5.40}$$

assuming that $\hat{\beta}_{j,k}$ is normalized. Equation 5.40 reveals that the k-th word proportions of topic j is estimated as the prior modal estimate plus the average number of times word k was assigned to topic j in the corpus.

Finally, the M-step update for ϵ is obtained by maximizing $Q(\lambda, \lambda^{(q)})$ with respect to ϵ:

$$\frac{\delta \hat{L}(\lambda, \lambda^{(q)})}{\delta \epsilon} = \frac{\delta Q(\lambda, \lambda^{(q)})}{\delta \epsilon} + \frac{\delta \widehat{\log [p(\lambda)]}}{\delta \epsilon}. \tag{5.41}$$

Define the indicator variable

$$I(ss) = \begin{cases} 1, & \text{word } w_t \text{ is first in a sentence} \\ 0, & \text{otherwise.} \end{cases}$$

First,

$$\begin{aligned}
\frac{\delta Q(\lambda,\lambda^{(q)})}{\delta\epsilon} =& \frac{\delta}{\delta\epsilon}\left[\sum_{d=0}^{D-1}\sum_{i=0}^{2K-1}\log\left[p(w_d, s_d = i;\lambda^{(q)})\right]\cdot p(s_d = i|d, w_d;\lambda^{(q)})\right] \\
=& \frac{\delta}{\delta\epsilon}\left[\sum_{d=0}^{D-1}\sum_{i=0}^{2K-1}\sum_{j=0}^{2K-1}\sum_{t=1}^{N_d-1}\log\left[p(s_{d,t} = j|s_{d,t-1} = i;\lambda^{(q)})\right]\cdot p(s_{d,t} = j|d, w_d;\lambda^{(q)})\right] \\
=& \frac{\delta}{\delta\epsilon}\left[\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\sum_{t=1}^{N_d-1}\log\left[(\epsilon\theta_{d,j})^{I(ss)}\right]\cdot p(s_{d,t} = j|d, w_d;\lambda^{(q)})\right] + \\
& \frac{\delta}{\delta\epsilon}\left[\sum_{d=0}^{D-1}\sum_{i=0}^{2K-1}\sum_{j=K}^{2K-1}\sum_{t=1}^{N_d-1}\log\left[(1-\epsilon)^{I(ss,i\in\{j-K,j\})}\right]\right]\cdot \\
& p(s_{d,t} = j, s_{d,t-1} = i|d, w_d;\lambda^{(q)}) \\
=& \frac{\delta}{\delta\epsilon}\left[\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\sum_{t=1}^{N_d-1}\log\left[\epsilon^{I(ss)}\right]\cdot p(s_{d,t} = j|d, w_d;\lambda^{(q)})\right] + \\
& \frac{\delta}{\delta\epsilon}\left[\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\sum_{t=1}^{N_d-1}\log\left[\theta_{d,j}^{I(ss)}\right]\cdot p(s_{d,t} = j|d, w_d;\lambda^{(q)})\right] + \\
& \frac{\delta}{\delta\epsilon}\left[\sum_{d=0}^{D-1}\sum_{i=0}^{2K-1}\sum_{j=K}^{2K-1}\sum_{t=1}^{N_d-1}\log\left[(1-\epsilon)^{I(ss,i\in\{j-K,j\})}\right]\right]\cdot \\
& p(s_{d,t} = j, s_{d,t-1} = i|d, w_d;\lambda^{(q)}) \\
=& \frac{1}{\epsilon}\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\sum_{t=1}^{N_d-1} p(s_{d,t} = j|d, w_d;\lambda^{(q)})I(ss) - \\
& \frac{1}{1-\epsilon}\sum_{d=0}^{D-1}\sum_{j=K}^{2K-1}\sum_{t=1}^{N_d-1} p(s_{d,t} = j|d, w_d;\lambda^{(q)})I(ss). \qquad (5.42)
\end{aligned}$$

Setting $\frac{\delta Q(\lambda,\lambda^{(q)})}{\delta\epsilon} := 0$,

$$\begin{aligned}
\frac{1-\epsilon}{\epsilon} &= \frac{\sum_{d=0}^{D-1}\sum_{j=K}^{2K-1}\sum_{t=1}^{N_d-1} p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(ss)}{\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\sum_{t=1}^{N_d-1} p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(ss)}\\
\frac{1}{\epsilon} &= \frac{\sum_{d=0}^{D-1}\sum_{t=0}^{N_d-1}\sum_{j=0}^{2K-1} p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(ss)}{\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\sum_{t=1}^{N_d-1} p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(ss)}\\
&= \frac{\sum_{d=0}^{D-1} N_{d,sen} - D}{\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\sum_{t=1}^{N_d-1} p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(ss)}.
\end{aligned}$$

The M-step update for ϵ is

$$\hat{\epsilon} = \frac{\sum_{d=0}^{D-1}\sum_{j=0}^{K-1}\sum_{t=1}^{N_d-1} p(s_{d,t}=j|d,w_d;\lambda^{(q)})I(ss)}{\sum_{d=0}^{D-1} N_{d,sen} - D} \tag{5.43}$$

and can be rewritten equivalently as given in Gruber, Rosen-Zvi, and Weiss (2007) as

$$\hat{\epsilon} = \frac{\sum_{d=0}^{D-1}\sum_{t=1}^{N_d-1} p(\psi_{d,t}=1|d,w_d;\lambda^{(q)})I(ss)}{\sum_{d=0}^{D-1} N_{d,sen} - D} \tag{5.44}$$

which can be interpreted as the average number of times in the corpus that the topic for a sentence switched (excluding the first sentence of each document) relative to the number of sentences in the corpus (excluding the first sentence of each document).

5.4 A Viterbi Algorithm for the Hidden Topic Markov Model

Since the EM algorithm does not provide "hard" state assignments for the state space, the Viterbi algorithm is extended to the HTMM framework. The Viterbi algorithm seeks the most likely sequence of states rather than the most likely state for each observation. For a given document, let

$$\delta_t(i) = \max_{s_0,\ldots,s_{t-1}} \left\{p(s_0,\ldots,s_{t-1},s_t=i,w_0,\ldots,w_t)\right\}. \tag{5.45}$$

where $i \in \{0,\ldots,2K-1\}$ and $t \in \{0,\ldots,N_d-1\}$. Let

$$\delta_{t+1}(j) = \left[\max_i \left\{\delta_t(i)p(s_t = j|s_{t-1} = i)\right\}\right] p(w_{t+1}|s_{t+1} = j). \tag{5.46}$$

Define the index variable

$$\psi_{t+1}(j) = i, \tag{5.47}$$

which holds the previous state that maximizes $\delta_t(i)p(s_{t+1} = j|s_t = i)$. For the first word in a document, let

$$\begin{aligned}\delta_0(i) &= p(s_0 = i)p(w_0|s_0 = i)\\ &= \begin{cases}\theta_i\beta_i, k^{I(w_0=k)}, & i = 0, \ldots, K-1\\ 0, & i = K, \ldots, 2K-1\end{cases},\end{aligned} \tag{5.48}$$

and $\psi_0(i) = -1$. For $t = 1$ to $t = N_d - 1$, compute $\delta_t(j)$ and $\psi_t(j)$ as follows. If w_t is the first word in a sentence:

$$\begin{aligned}\delta_t(j) &= \max_i \left\{\delta_{t-1}(i)p(s_t = j|s_{t-1} = i)\right\} p(w_t|s_t = j)\\ &= \left[\max_i \begin{cases}\delta_{t-1}(i)\epsilon\theta_j, & j = 0, \ldots, K-1\\ \delta_{t-1}(i)(1-\epsilon), & i \in \{j-K, j\}\\ 0, & \text{otherwise}\end{cases}\right] \beta_j^{I(w_t)},\end{aligned} \tag{5.49}$$

and

$$\psi_t(j) = \arg\max_i \begin{cases}\delta_{t-1}(i)\epsilon\theta_j, & j = 0, \ldots, K-1\\ \delta_{t-1}(i)(1-\epsilon), & i \in \{j-K, j\}\\ 0, & \text{otherwise}\end{cases}. \tag{5.50}$$

If w_t is not the first word in a sentence, then transitions are deterministic rather than stochastic, and instead,

$$\delta_t(j) = \max_i \{\delta_{t-1}(i)\} \begin{cases} \beta_j^{I(w_t)}, & j = \begin{cases} i+K, & i \in \{0,\dots,K-1\} \\ i, & i \in \{K,\dots,2K-1\} \end{cases}, \\ 0, & \text{otherwise} \end{cases} \tag{5.51}$$

and

$$\begin{aligned} \psi_t(j) &= \arg\max_i \{\delta_{t-1}(i) p(s_t = j | s_{t-1} = i)\} \\ &= \arg\max_i \{\delta_{t-1}(i)\}. \end{aligned} \tag{5.52}$$

In order to obtain the optimal state sequence, compute

$$s^*_{N_d-1} = \arg\max_i \{\delta_{N_d-1}(i)\} \tag{5.53}$$

and then for $t = N_d - 1$ to $t = 0$, find

$$s^*_t = \psi_{t+1}(s^*_{t+1}) \tag{5.54}$$

to obtain resulting highest probability sequence of states s^*.

Chapter 6

Posterior Approximation with Gibbs Sampling for Inference in Hidden Topic Markov Models

6.1 Derivation of Full Conditional Distributions for a Gibbs Sampler

Recall that the parameters of interest for inference are $\lambda = (\theta, \beta, \epsilon)$. Since the words w_n are the only observed data, we seek to design a Gibbs sampling algorithm that can approximate draws from the joint posterior distribution

$$p(\theta_d, \beta, \epsilon, s_d | d, w_d; \alpha, \eta). \tag{6.1}$$

We start by noting that the joint posterior distribution can be written proportionally

$$\begin{aligned} p(\theta_d, \beta, \epsilon, s_d | d, w_d; \alpha, \eta) \propto & p(\theta_d; \alpha) \cdot p(\beta; \eta) \cdot p(s_{d,0}|\theta_d) \cdot p(w_{d,0}|s_{d,t}, \beta) \cdot \\ & \prod_{t=1}^{N_d - 1} p(s_{d,t}|s_{d,t-1}, \theta_d, \epsilon) \cdot p(w_{d,t}|s_{d,t}, \beta). \end{aligned} \tag{6.2}$$

We can use a Gibbs sampling algorithm to sample from this posterior by sampling from each full conditional distribution separately. First, we consider $p(\theta_d|d, s_d; \alpha)$. We define the indicator variable $I(ss)$ to be equal to 1 if word $w_{d,t}$ is the first word in a sentence and 0 otherwise.

$$\begin{aligned} p(\theta_d|d, s_d; \alpha) \propto & p(\theta_d; \alpha) \cdot p(s_{d,0}|\theta_d) \cdot \prod_{t=1}^{N_d - 1} p(s_{dt}|s_{d,t-1}, \theta_d, \epsilon) \\ = & \prod_{k=0}^{K-1} \theta_{dk}^{\alpha - 1} \cdot \prod_{k=0}^{K-1} \theta_{dk}^{I(s_{d0}=k)} \cdot \prod_{t=1}^{N_d - 1} p(s_{dt} = j | s_{d,t-1} = i, \theta_d, \epsilon) \\ = & \prod_{k=0}^{K-1} \theta_{dk}^{\alpha + I(s_{d0}=k) - 1} \cdot \prod_{t=1}^{N_d - 1} p(s_{dt} = j | s_{d,t-1} = i, \theta_d, \epsilon). \end{aligned} \tag{6.3}$$

Since the Markov chain of the state space either transitions to a new topic j with probability $\epsilon\theta_j$ or does not transition to a new topic with probability $1 - \epsilon$, Equation 6.3 can be written as

$$\begin{aligned}
p(\theta_d|d, s_d; \alpha) &\propto \prod_{k=0}^{K-1} \theta_{dk}^{\alpha+I(s_{d0}=k)-1} \cdot \prod_{t=1}^{N_d-1} (\epsilon\theta_{dj})^{I(s_{dt}\in\{0,\ldots,K-1\},ss=1)} \cdot \\
&\quad (1-\epsilon)^{I(s_{dt}\in\{K,\ldots,2K-1\},s_{d,t-1}\in\{j-K,j\},ss=1)} \\
&\propto \prod_{k=0}^{K-1} \theta_{dk}^{\alpha+I(s_{d0}=k)-1} \cdot \prod_{t=1}^{N_d-1} \theta_{dk} I(s_{dt}=k, k\in\{0,\ldots,K-1\}, ss=1) \\
&= \prod_{k=0}^{K-1} \theta_{dk}^{\alpha+I(s_{d0}=k)+\sum_{t=1}^{N_d-1} I(s_{dt}=k,k\in\{0,\ldots,K-1\},ss=1)-1} \\
&= \prod_{k=0}^{K-1} \theta_{dk}^{\alpha+\sum_{t=0}^{N_d-1} I(s_{dt}=k,k\in\{0,\ldots,K-1\},ss=1)-1} \\
p(\theta_d|d, s_d; \alpha) &\propto \prod_{k=0}^{K-1} \theta_{dk}^{\alpha+\sum_{t=0}^{N_d-1} I(z_{dt}=k,\psi_{dt}=1,ss=1)-1}. \qquad (6.4)
\end{aligned}$$

This is recognizable as a Dirichlet distribution. Defining $n_{dz=i}$ as the number of times a sentence in document d was assigned to topic i while $\psi_d = 1$,

$$(\theta_d|z_d; \alpha) \sim \text{Dirichlet}(\alpha + n_{dz=0}, \ldots, \alpha + n_{dz=K-1}), d \in \{0, \ldots, D-1\}. \qquad (6.5)$$

This is nicely interpretable since the parameters of the prior Dirichlet distribution for θ_d are updated for a given topic proportion θ_{dk} by the number of new draws of topic k in the document.

Similarly, we can determine the full conditional distribution of β_j for topic j.

$$\begin{aligned} p(\beta_j|d, s_d; \alpha) \propto & p(\beta_j; \eta) \cdot p(w_d|s_d, \beta_j) \\ &= \prod_{k=0}^{V-1} \beta_{jk}^{\eta-1} \cdot \prod_{d=0}^{D-1} \prod_{t=0}^{N_d-1} p(w_{dt} = k|s_{dt} = j) \\ &= \prod_{k=0}^{V-1} \beta_{jk}^{\eta-1} \cdot \prod_{d=0}^{D-1} \prod_{t=0}^{N_d-1} \beta_{jk}^{I(s_{dt} \in \{j+K, j\}, w_{dt}=k)} \\ &= \prod_{k=0}^{V-1} \beta_{jk}^{\eta + \sum_{d=0}^{D-1} \sum_{t=0}^{N_d-1} I(s_{dt} \in \{j+K, j\}, w_{dt}=k) - 1} \\ &= \prod_{k=0}^{V-1} \beta_{jk}^{\eta + \sum_{d=0}^{D-1} \sum_{t=0}^{N_d-1} I(z_{dt}=j, w_{dt}=k) - 1}. \end{aligned} \tag{6.6}$$

This is recognizable as another Dirichlet distribution. Defining $n_{z=j,w=k}$ as the number of times that topic j and word k co-occur in the corpus,

$$(\beta_j|z, w; \eta) \sim \text{Dirichlet}(\eta + n_{z=j,w=0}, \ldots, \eta + n_{z=j,w=V-1}), j \in \{0, \ldots, K-1\}. \tag{6.7}$$

Next, we consider a full conditional distribution for ϵ. While Gruber, Rosen-Zvi, and Weiss (2007) did not introduce a prior for ϵ, we propose a Beta(ζ, ζ) prior distribution with equal shape parameters ζ to incorporate prior belief about topic coherence and to allow for updating estimates of ϵ in the presence of new data (i.e., using the posterior distribution of ϵ from one model as the prior distribution for ϵ for a new model of new data). Using identical parameters yields symmetric priors for ϵ of the form

$$p(\epsilon; \zeta) = \frac{\Gamma(2\zeta)}{\Gamma(\zeta)\Gamma(\zeta)} \epsilon^{\zeta-1}(1-\epsilon)^{\zeta-1}. \tag{6.8}$$

We now derive the full conditional distribution of ϵ,

$$
\begin{aligned}
p(\epsilon|s;\zeta) \propto & p(\epsilon;\zeta) \cdot \prod_{d=0}^{D-1} \prod_{t=1}^{N_d-1} p(s_{dt}|s_{d,t-1,\theta_d}, \epsilon) \\
= & \epsilon^{\zeta-1}(1-\epsilon)^{\zeta-1} \cdot \\
& \prod_{d=0}^{D-1} \prod_{t=1}^{N_d-1} \prod_{i=0}^{2K-1} \prod_{j=0}^{2K-1} \left[(\epsilon\theta_{dj})^{I(s_{dt}=j, j\in\{0,\ldots,K-1\}, ss=1)} \right] \\
& \left[(1-\epsilon)^{I(s_{dt}=j, j\in\{K,\ldots,2K-1\}, s_{d,t-1}=i, i\in\{j-K,j\}, ss=1)} \right] \\
= & \epsilon^{\zeta-1}(1-\epsilon)^{\zeta-1} \prod_{d=0}^{D-1} \prod_{t=1}^{N_d-1} \Big[\prod_{i=0}^{2K-1} \prod_{j=0}^{K-1} \epsilon\theta_{dj} \Big]^{I(s_{dt}=j, ss=1)} \cdot \\
& \Big[\prod_{i\in\{j-K,j\}} \prod_{j=K}^{2K-1} (1-\epsilon) \Big]^{I(s_{dt}=j, s_{d,t-1}=i, ss=1)} \Big] \\
\propto & \epsilon^{\zeta + \sum_{d=0}^{D-1} \sum_{t=1}^{N_d-1} \sum_{i=0}^{2K-1} \sum_{j=0}^{K-1} I(s_{dt}=j, ss=1) - 1} \cdot \\
& (1-\epsilon)^{\zeta + \sum_{d=0}^{D-1} \sum_{t=1}^{N_d-1} \sum_{i\in\{j-K,j\}} \sum_{j=K}^{2K-1} I(s_{dt}=j, s_{d,t-1}=i, ss=1) - 1} \\
p(\epsilon|s;\zeta) \propto & \epsilon^{\zeta + \sum_{d=0}^{D-1} \sum_{t=1}^{N_d-1} I(\psi_{dt}=1, ss=1) - 1} (1-\epsilon)^{\zeta + \sum_{d=0}^{D-1} \sum_{t=1}^{N_d-1} I(\psi_{dt}=0, ss=1) - 1}.
\end{aligned}
\tag{6.9}
$$

We define $n_{\psi=1}$ as the number of sentences in a corpus where $\psi_{dt} = 1$ (i.e., the number of sentences for which a new topic was drawn). We then define n_{sen} as the number of sentences in the corpus and recall that D is the number of documents in the corpus.

We can rewrite the full conditional posterior distribution of ϵ as

$$p(\epsilon|s;\zeta) \propto \epsilon^{\zeta+n_{\psi=1}-1}(1-\epsilon)^{\zeta+n_{sen}-D-n_{\psi=1}-1}, \tag{6.10}$$

Therefore,

$$(\epsilon|s;\zeta) \sim \text{Beta}(\zeta + n_{\psi=1}, \zeta + n_{sen} - D - n_{\psi=1}). \tag{6.11}$$

Finally, we need to learn the posterior distribution of the Markov chain in the state space, so we derive the full conditional distribution of the initial state in a document s_{d0} and later derive the full conditional distribution of state t in a document s_{dt}.

For the first state in a document

$$\begin{aligned} p(s_{d0}|w_d,\theta_d) &\propto p(s_{d0}|\theta_d)p(w_d|s_{d0},\beta) \\ &= p(s_{d0}|\theta_d)p(w_{d0}|s_{d0},\beta)p(w_{d1},\dots,w_{d,N_d-1}|s_{d0},\beta). \end{aligned} \tag{6.12}$$

It should be clear that $p(w_{d1},\dots,w_{d,N_d-1}|s_{d0},\beta)$ is simply the backward variable ρ_{d0} defined in Chapter 6. Therefore,

$$p(s_{d0}=i|w_d,\theta_d) \propto \theta_{di}\beta_{i,w_0}\rho_{d0}(i). \tag{6.13}$$

Similarly, we can derive the full conditional distribution of state t in document d,

$$\begin{aligned} p(s_{dt}|s_{d,t-1},w_{dt},\dots,w_{d,N_d-1},\theta_d) &\propto p(s_{dt}|s_{d,t-1},\theta_d,\epsilon)p(w_{dt},\dots,w_{d,N_d-1}|s_{dt},\beta) \\ &= p(s_{dt}|s_{d,t-1},\theta_d,\epsilon)p(w_{dt}|s_{dt})p(w_{d,t+1},\dots,w_{d,N_d-1}|s_{dt},\beta) \end{aligned} \tag{6.14}$$

Taking advantage of the model structure,

$$p(s_{dt}=j|s_{d,t-1}=i,w_{dt:(N_d-1)},\theta_d) \propto \begin{cases} \epsilon\theta_j\beta_{j,w_{dt}}\rho_{dt}(j)\ , & j\in\{0,\dots,K-1\} \\ (1-\epsilon)\beta_{j-K,w_{dt}}\rho_{dt}(j)\ , & j\in\{K,\dots,2K-1\}\ , \\ & i\in\{j-K,j\} \end{cases} \tag{6.15}$$

6.2 A Gibbs Sampling Algorithm for HTMM

Equipped with the full conditional distributions of ϵ, θ, β, and s, a Gibbs sampling algorithm for the Hidden Topic Markov Model is shown in Figure 6.1 where T is the number of iterations or sweeps of the sampler and K is the number of topics.

```
Initialize $s$, $\alpha$, $\eta$, $\zeta_1$, $\zeta_2$;
for $r = 0$ to $T - 1$ do
    for $d = 0$ to $D - 1$ do
        for $z = 0$ to $z = K - 1$ do
            Compute $n_{dz}$ ;
        end
    end
    for $k = 0$ to $k = V - 1$ do
        for $z = 0$ to $z = K - 1$ do
            Compute $n_{z=j,w=k}$ ;
        end
    end
    Compute $n_{\psi=1}$ ;
    for $j = 0$ to $j = K - 1$ do
        Sample $\beta_j \sim \text{Dirichlet}(\eta + n_{z=j,w=0}, \ldots, \eta + n_{z=j,w=V-1})$ ;
    end
    Sample $\epsilon \sim \text{Beta}(\zeta + n_{\psi=1}, \zeta + n_{sen} - D - n_{\psi=1})$ ;
    for $d = 0$ to $D - 1$ do
        Sample $\theta_d \sim \text{Dirichlet}(\alpha + n_{dz=0}, \ldots, \alpha + n_{dz=K-1})$ ;
    end
    Compute backward variables $\rho_d$ ;
    for $i = 0$ to $i = K - 1$ do
        Sample $s_{d0}(i) \sim \theta_{di}\beta_{i,w_0}\rho_{d0}(i)$ ;
        Set $s_{d0}(i + K) := 0$ ;
    end
    for $t = 1$ to $t = N_d - 1$ do
        for $j = 0$ to $j = 2K - 1$ do
            if $j \in \{0, \ldots, K - 1\}$ then
                Sample $s_{dt}(j)|s_{d,t-1} = i \sim \epsilon\theta_j\beta_{j,w_{dt}}\rho_{dt}(j)$ ;
            end
            else if $i \in \{j - K, j\}$ then
                Sample $s_{dt}(j)|s_{d,t-1} = i \sim (1 - \epsilon)\beta_{j-K,w_{dt}}\rho_{dt}(j)$ ;
            end
        end
    end
end
```

Figure 6.1: Gibbs sampler for the Hidden Topic Markov Model

Chapter 7

Evaluation of the Gibbs Sampler and EM Algorithm for Hidden Topic Markov Models

7.1 Simulation Study of HTMM

In order to assess the performance of the expectation-maximization algorithm and the Gibbs sampler proposed for inference for the hidden topic Markov model (HTMM), a simulation study was conducted. Twelve data sets were simulated from the generative model assumed for the HTMM. Each data set contained 600 documents and was generated from a vocabulary of $V = 1000$ words. I assumed that each sentence would contain an average of 20 words, so the number of words per sentence N_s was drawn according to $N_w \sim \text{Poisson}(\lambda = 20)$. The average number of sentences N_s was drawn from either $N_s \sim \text{Poisson}(\lambda = 10)$ or $N_s \sim \text{Poisson}(\lambda = 250)$ to approximate the average number of sentences that might be expected in an abstract and a scientific journal article, respectively. The number of topics used to generate the corpora was either $K = 2$ or $K = 10$ topics. Finally, ϵ was set to be in $\epsilon \in \{0.1, 0.5, 0.9\}$. I set α for each document to be a random permutation of $\left(\frac{1}{K}, \frac{2}{K}, \ldots, \frac{K}{K}\right)$.

Similarly, I set η for each topic to be a random permutation of $\left(\frac{1}{V}, \frac{2}{V}, \ldots, \frac{V}{V}\right)$. The generative attributes of the 12 synthetic data sets are shown in Table 7.1.

Data	D	V	N_w	N_s	K	ϵ
1	600	1000	20	10	2	0.1
2	600	1000	20	10	2	0.5
3	600	1000	20	10	2	0.9
4	600	1000	20	10	10	0.1
5	600	1000	20	10	10	0.5
6	600	1000	20	10	10	0.9
7	600	1000	20	250	2	0.1
8	600	1000	20	250	2	0.5
9	600	1000	20	250	2	0.9
10	600	1000	20	250	10	0.1
11	600	1000	20	250	10	0.5
12	600	1000	20	250	10	0.9

Table 7.1: Attributes of data sets simulated from the generative HTMM

For each data set, both the EM and Gibbs sampling algorithms were trained on 500 of the documents from a given data set. The EM algorithm was run until convergence where convergence was defined to be a change in log-likelihood of magnitude less than 0.01. The Gibbs samplers were run for a burn-in period of 1000 iterations with a thinning rate of 10. A final sample of $n = 100$ was obtained for each data set. Both the EM and Gibbs models used hyperparameters $\alpha = 1 + 50/K$ where K was set to match the known generative K for a given data set and $\eta = 1.01$ following Gruber, Rosen-Zvi, and Weiss (2007). Furthermore, the hyperparameter for ϵ was set to $\zeta = 1$ such that $\epsilon \sim \text{Beta}(1, 1)$ was an non-informative uniform prior, $\epsilon \sim \text{Uniform}(0, 1)$.

Results for EM and Gibbs inference are given in Tables 7.1 and 7.1. Estimates of ϵ and the absolute relative error of $\hat{\epsilon}$ are reported. The estimation error of θ and β were calculated using the L1 norm of the difference between the true parameter matrix and the estimated matrix relative to the number of entries in the difference matrix. Finally, topic recovery accuracies are provided.

Data	N_s	K	ϵ	$\hat{\epsilon}$	$\frac{\lvert\hat{\epsilon}-\epsilon\rvert}{\epsilon}$	$\frac{\lvert\lvert\hat{\theta}-\theta\rvert\rvert_1}{DK}$	$\frac{\lvert\lvert\hat{\beta}-\beta\rvert\rvert_1}{KV}$	Topic Recovery Accuracy
1	10	2	0.1	0.085	0.150	0.197	0.000103	0.780
2	10	2	0.5	0.394	0.212	0.195	0.000103	0.739
3	10	2	0.9	0.678	0.247	0.186	0.000101	0.736
4	10	10	0.1	0.102	0.020	0.060	0.000900	0.365
5	10	10	0.5	0.490	0.019	0.060	0.000810	0.282
6	10	10	0.9	0.889	0.012	0.059	0.000700	0.276
7	250	2	0.1	0.082	0.181	0.166	0.000021	0.832
8	250	2	0.5	0.462	0.075	0.086	0.000021	0.753
9	250	2	0.9	0.877	0.026	0.051	0.000021	0.764
10	250	10	0.1	0.096	0.039	0.061	0.000696	0.376
11	250	10	0.5	0.492	0.015	0.073	0.000873	0.277
12	250	10	0.9	0.898	0.002	0.078	0.000872	0.276

Table 7.2: Performance of HTMM-EM on simulated data

As shown in Table 7.1, the HTMM EM algorithm recovers ϵ well. For the twelfth data set, the lowest estimation error for $\hat{\epsilon}$ is only 0.2%, while the largest estimation error on the third data set is 24.7%. These results suggest that the accuracy of the EM algorithm estimates for ϵ are poorest when the number of sentences per document is relatively small and the number of topics is small regardless of the true value of ϵ. Since the number of sentences in a corpus drive the estimate of ϵ, it is reasonable to expect that a corpus of shorter documents (e.g., scientific abstracts) would provide less information about the Markovian dynamics of topics when using sentences as the smallest unit of topic assignment. Conversely, the EM algorithm is most precise in its estimation of ϵ when the average number of sentences per document is large and the number of topics is large. This is also reasonable since the large number of sentences in the corpus and the larger number of topics allow for more variable topical dynamics. One reasonable hypothesis for future research is that the quality of $\hat{\epsilon}$ depends on its degrees of freedom (i.e., the number of sentences in the corpus and the number of topics considered).

A similar pattern of results emerged for the accuracy of the EM algorithm when estimating the document topic proportions $\theta = (\theta_0, \ldots, \theta_{D-1})$. Larger estimation errors for θ

were observed for smaller data sets where the average number of sentences per document ($N_s = 10$) was small and the number of topics ($K = 2$) was smallest regardless of ϵ. Estimation errors for θ were relatively low and similar for larger average numbers of sentences and topics. Future work could consider examining whether the quality of $\hat{\theta}$ depends on the number of sentences in the corpus and the number of topics considered.

An interesting exception to these results were the estimates obtained for the seventh data set ($N_s = 250$, $K = 2$, $\epsilon = 0.1$). While estimates of ϵ and θ were otherwise better for larger N_s, the error in estimates of both ϵ and θ for this data set were comparable with errors in estimation of ϵ and θ for the first three data sets where $N_s = 10$ and $K = 2$.

The EM algorithm seemed to do universally well at estimating the topic-word probability matrix β for all data sets. The errors in estimation were an order of magnitude smaller for the seventh, eighth, and ninth data sets than the others. These three data sets had a large average number of sentences per document ($N_s = 250$) but only two topics. It is not surprising that the topic-word probabilities were estimated more precisely for these data sets since word frequencies were greater and the number of topic-word associations to be estimated were small, resulting in an optimal ratio of observed words to parameters for inference.

Finally, the accuracy of word-to-topic assignments was computed using a modification of the standard Viterbi algorithm that took into account the structure of the HTMM. Label switching was addressed by creating equivalencies between the true topics and inferred topics using the simple argmax of the true and inferred topic co-occurrences. Topic assignment accuracies were universally better when the number of topics was small ($K = 2$) than when the number of topics was large ($K = 10$). Regardless of the number of topics and the average number of sentences per document, topic assignment accuracy was slightly higher when $\epsilon = 0.1$ than when $\epsilon = 0.5$ or $\epsilon = 0.9$. Performance degraded dramatically as the number of topics increased, which suggests that the Viterbi algorithm is not a particularly accurate method for predicting topics for simulated data.

Another potential source of error in state assignments is label switching. The labeling of learned topics from the EM algorithm and especially from the Gibbs sampler do not have to correspond to the true labels in the simulated data since the likelihood in EM and the posterior distributions in the Gibbs sampler are invariant to permutations of topics (Jasra, Holmes, and Stephens 2005). More sophisticated methods of handling label switching could be considered to remedy this limitation, particularly if the EM algorithm is used.

Data	N_s	K	ϵ	$\hat{\epsilon}$	$\frac{\lvert\hat{\epsilon}-\epsilon\rvert}{\epsilon}$	$\frac{\lvert\lvert\hat{\theta}-\theta\rvert\rvert_1}{DK}$	$\frac{\lvert\lvert\hat{\beta}-\beta\rvert\rvert_1}{KV}$	Topic Recovery Accuracy
1	10	2	0.1	0.086 (0.077, 0.093)	0.137	0.197	0.000102	0.998
2	10	2	0.5	0.390 (0.377, 0.406)	0.219	0.196	0.000103	0.993
3	10	2	0.9	0.669 (0.655, 0.683)	0.257	0.186	0.000101	0.992
4	10	10	0.1	0.092 (0.083, 0.100)	0.085	0.060	0.000804	0.992
5	10	10	0.5	0.483 (0.467, 0.496)	0.033	0.060	0.000741	0.960
6	10	10	0.9	0.872 (0.862, 0.882)	0.031	0.060	0.000815	0.935
7	250	2	0.1	0.082 (0.080, 0.083)	0.184	0.168	0.000021	0.999
8	250	2	0.5	0.459 (0.457, 0.462)	0.081	0.088	0.000021	0.994
9	250	2	0.9	0.872 (0.870, 0.874)	0.031	0.052	0.000021	0.991
10	250	10	0.1	0.096 (0.094, 0.098)	0.036	0.064	0.000867	0.996
11	250	10	0.5	0.490 (0.487, 0.492)	0.021	0.069	0.000781	0.972
12	250	10	0.9	0.894 (0.892, 0.896)	0.006	0.071	0.000789	0.954

Table 7.3: Performance of HTMM-Gibbs on simulated data

As shown in Table 7.1, the HTMM Gibbs sampler recovers ϵ well. $\hat{epsilon}$ is the mean of the sample path of ϵ and is accompanied by a 95% Bayesian credible interval. Paralleling the results for the EM algorithm, for the twelfth data set, the lowest estimation error for $\hat{\epsilon}$ is only 0.6%, while the largest estimation error on the third data set is 25.7%. These results suggest that the accuracy of the Gibbs sampler for ϵ is, like the accuracy of the EM algorithm, poorest when the number of sentences per document is relatively small and the number of topics is small regardless of the true value of ϵ. Conversely, both the Gibbs sampler and EM algorithm are most precise in their estimation of ϵ when the average number of sentences per document is large and number of topics is large. One reasonable hypothesis for future research is that the quality of $\hat{\epsilon}$ depends on its degrees of freedom (i.e., the number of sentences in the corpus and the number of topics considered). Since ϵ is tied to the number of topics such that ϵ

increases as a function of the number of topics (Gruber, Rosen-Zvi, and Weiss 2007), the most important factor in the estimation of ϵ is likely the number of sentences in the corpus.

The accuracy of the Gibbs sampler and the EM algorithm when estimating the document topic proportions $\theta = (\theta_0, \ldots, \theta_{D-1})$ were virtually indistinguishable. Critically, errors for both methods were larger when the number of sentences in the corpus was smaller. The Gibbs sampler and EM algorithm were quite similar in their estimation of the topic-word probability matrix β for all data sets.

Finally, the accuracy of word-to-topic assignments was computed. Inferred topic assignments were determined using the median topic for each word in a document from the sample path. Label switching was resolved in the same manner described above for the EM algorithm. These accuracies were incredibly high and ranged from 93.5% to 99.9%. All accuracies were substantially better than those obtained using the Viterbi algorithm with the EM algorithm. As noticed with the EM-Viterbi approach, topic assignment accuracies were better when the number of topics was small ($K = 2$) than when the number of topics was large ($K = 10$) and slightly higher when $\epsilon = 0.1$ than when $\epsilon = 0.5$ or $\epsilon = 0.9$. However, the relative drop in accuracy was nearly negligible using Gibbs sampling.

Despite using a simply approach to handling label switching, the Gibbs sampler for HTMM recovered the underlying latent structure nearly perfectly across the twelve different data sets while the EM and Viterbi algorithms struggled to recover the underlying latent structure. At the same time, model parameter estimates errors were nearly equivalent using both methods. One advantage of the Gibbs sampling approach for inference is the ability to form Bayesian credible intervals and give distributional information about the model instead of the point estimates provided by the EM algorithm.

For small data sets, both algorithms are relatively fast and memory efficient. For large numbers of topics, vocabulary, documents, and words, Gibbs sampling becomes far slower and memory-expensive than the EM algorithm. This is a common limitation of Gibbs sampling and Markov Chain Monte Carlo (MCMC) in general, but is not necessarily insurmountable. Given the substantial gains in the accuracy of topic recovery using Gibbs sampling, future research on more efficient MCMC algorithms for the HTMM would be of interest. From experience, the HTMM EM algorithm can converge quickly and does not seem to be overly sensitive to initializations. However, the algorithm can also take a long time to converge, particularly for large data sets, and is not guaranteed to converge to a global maximum on the log-likelihood. There are advantages and disadvantages to both algorithms that warrant careful consideration in application, not unlike the choice between variational EM (Blei, Ng, and Jordan 2003) and Gibbs sampling (Griffiths and Steyvers 2004).

Finally, the rate of convergence of both methods is considered. For the sake of brevity, convergence is assessed for the EM and Gibbs algorithms on data set 1 where $N_s = 10$, $K = 2$, and $\epsilon = 0.1$. The quick convergence of the EM algorithm is evident in the asymptotic behavior of both the log-likelihood and $\hat{\epsilon}$ shown in Figure 7.1. It is unknown whether the algorithm converged to a global maximum or a local maximum, but empirically, convergence is typically swift and reasonably robust to variable initializations.

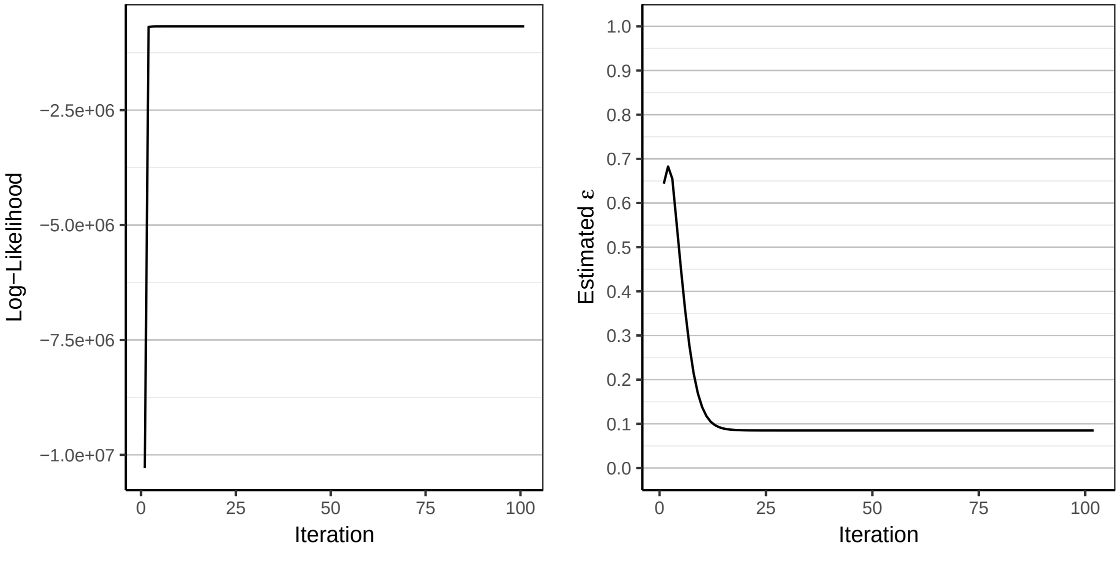

Figure 7.1: Convergence of log-likelihood for HTMM-EM algorithm (left). Convergence of estimated transition probability (right).

Assessment of the convergence of the Gibbs sampler is evident by examining the sample path for ϵ shown in Figure 7.2. Although ϵ is randomly initialized far from its true value, the sampler quickly moves away from this initialization and mixes. It is evident that the sampler explores the parameter space for ϵ as some draws explore regions above and below $\epsilon = 0.9$, but the chain stabilizes quickly around $\epsilon = 0.09$.

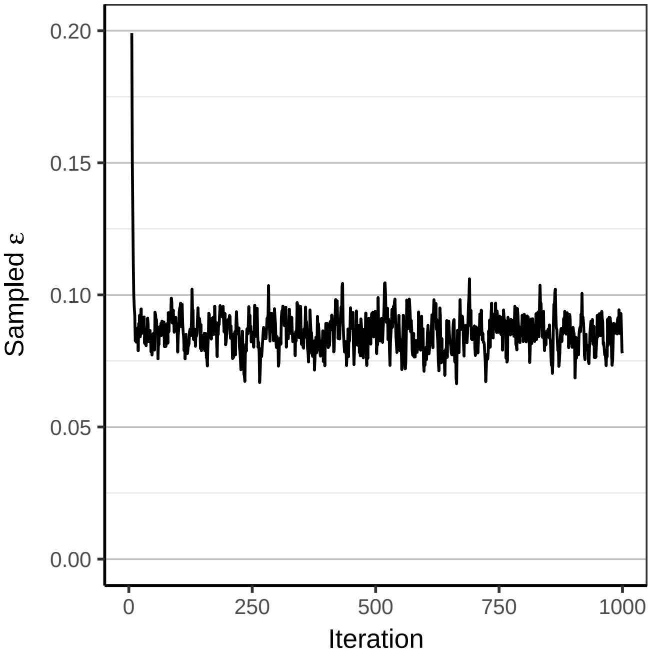

Figure 7.2: Sample path of transition probability from simulated data set 1.

The posterior distribution shown in Figure 7.3 of ϵ is unimodal with little variability and centered just below 0.1. While it's mode falls below the true parameter $\epsilon = 0.1$, the posterior distribution does include the true parameter value.

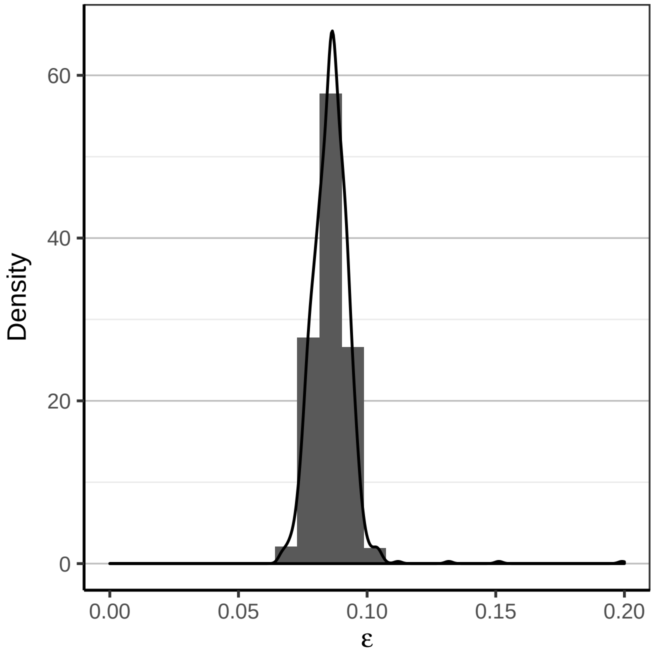

Figure 7.3: Posterior distribution of transition probability from simulated data set 1.

Inspection of the sample path in Figure 7.4 for the topic proportions θ_0 in the first document reveal while the sampler does explore smaller and larger values of θ_0, it struggles to converge in the absence of sufficient data (recall that there were only an average of 10 sentences per document in data set 1).

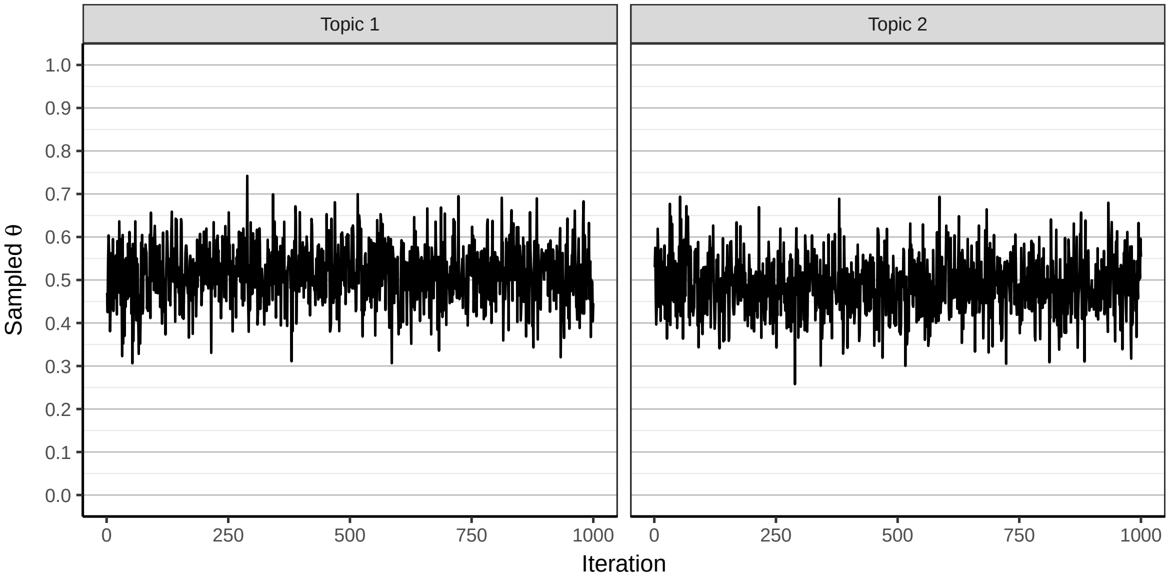

Figure 7.4: Sample path of topic proportions in the first document of simulated data set 1.

Furthermore, the sampler's estimate of θ_0 reflects the prior placed on θ_0 which was centered and heavily concentrated at 0.5 with a concentration parameter $\alpha = 26$; given the small amount of data available, is heavily influencing the sample path of θ_0 to remain near 0.5. This is typical of posterior samples when there is minimal information regarding a parameter in the data. The posterior distributions shown in Figure 7.5 for the topic proportions are symmetric and unimodal with a mode near the prior mean.

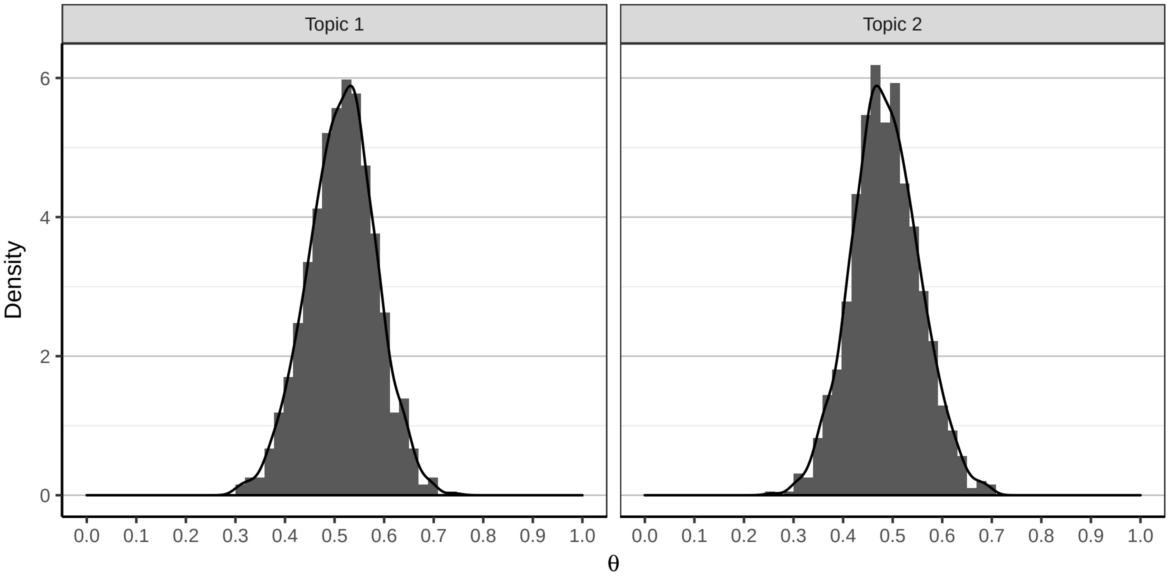

Figure 7.5: Posterior distribution of topic proportions in the first document from simulated data set 1.

The sample path of β in Figure 7.6 for topic-word probabilities of the first word for topics 1 and 2 reveals that the sample path for topic 1 favors draws for larger values of β_{11} while the sample path for topic 2 favors draws for values of β_{21} near the prior mean of 0.001.

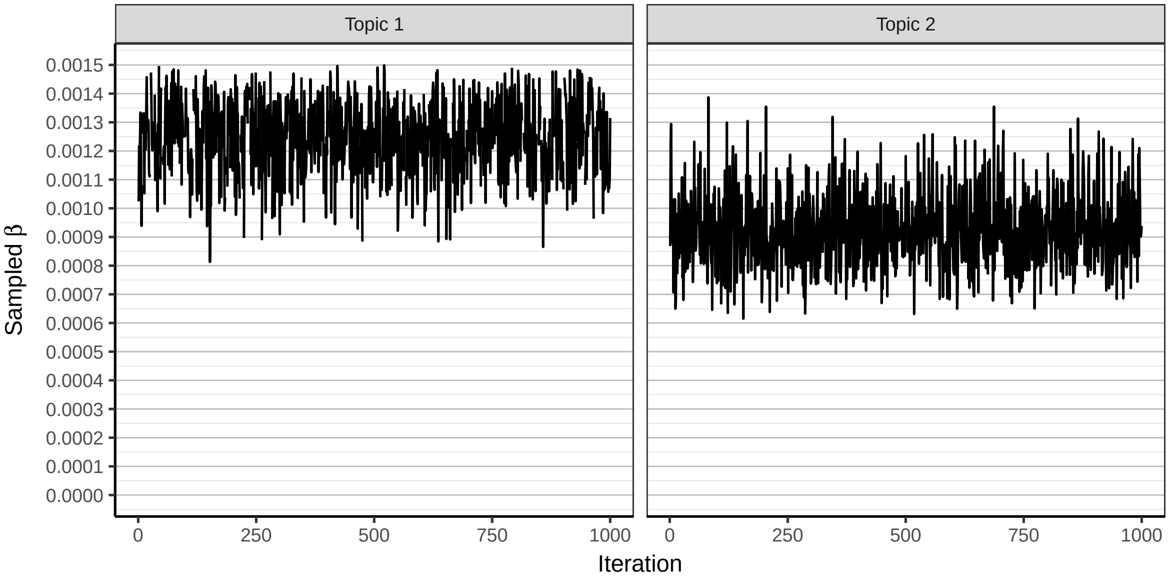

Figure 7.6: Sample path of topic-word probabilities for the first word from simulated data set 1.

The posterior distributions shown in Figure 7.7 for both components of β are unimodal and symmetric, but the distribution for topic 2 is centered at the prior mean of 0.001 while the distribution for topic 1 is shifted to the right of the prior mean. This suggests that the first word was drawn from the first topic, but not from the second topic. However, there is a fair amount of variability for both distributions which reflects the small size of the data set.

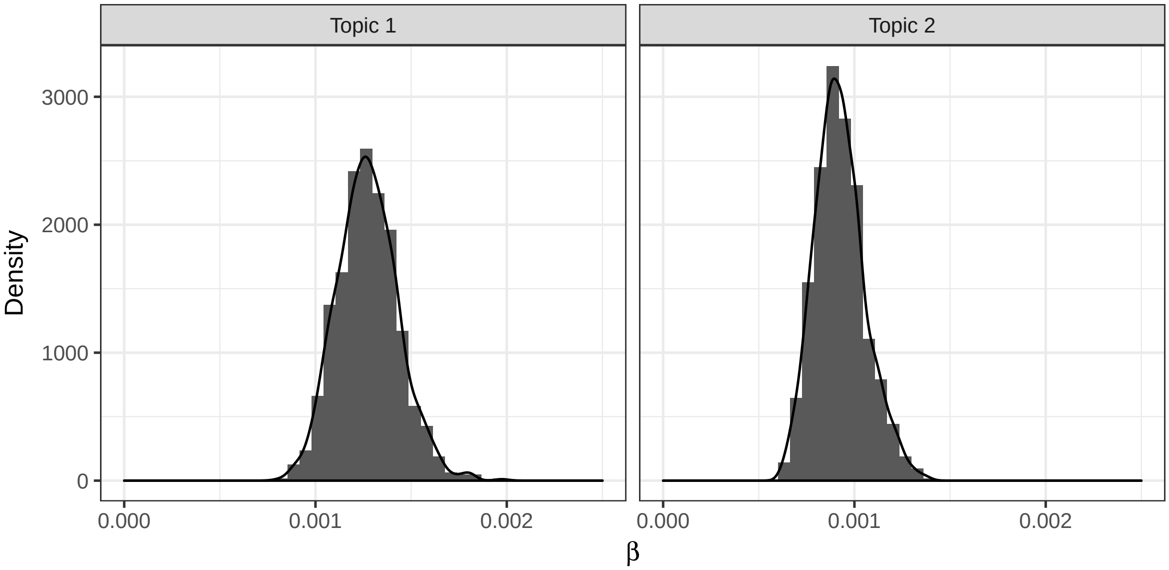

Figure 7.7: Posterior distributions of topic-word probabilities for the first word from simulated data set 1.

To assess the importance of the corpus size, a similar assessment of the model parameters inferred by Gibbs sampling is performed for data set 7. The only difference between data set 7 and data set 1 is that there was an average of 250 sentences per document in the former and an average of only 10 sentences per document in the latter.

The sample path for ϵ after burn-in and thinning in Figure 7.8 shows that the sampler converged for ϵ near 0.8 and remained stable.

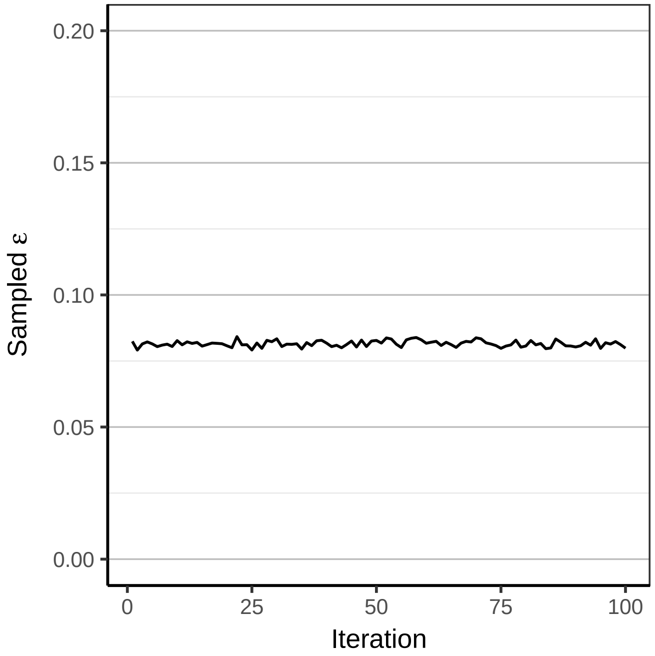

Figure 7.8: Sample path of transition probability from simulated data set 7.

The posterior distribution of ϵ is unimodal with less variability than the posterior distribution of ϵ for data set 1. The posterior for data set 7 is centered near 0.8 as shown in Figure 7.9.

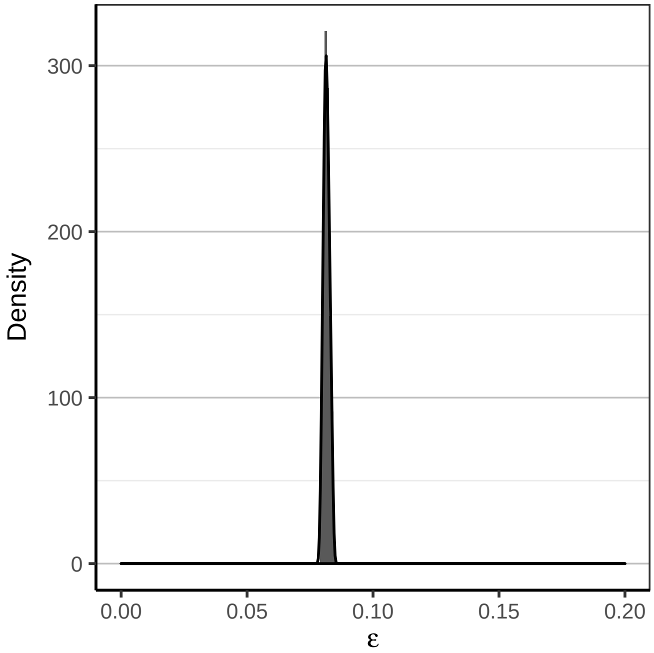

Figure 7.9: Posterior distribution of transition probability from simulated data set 1.

The sample path of θ_0 in Figure 7.10 in this data set provides clearer insight into a common dilemma in Bayesian mixture modeling: label switching. Here, only the thinned sample after burn-in is shown for clarity. The behavior of the sample path for the two topic proportions clearly shows label switching; the magnitudes of the two parameters switch back and forth between approximately 0.4 and 0.6 and this is reflected as multimodality in the posterior distributions shown in Figure 7.10. Two common approaches have been proposed for other Bayesian mixture models to address this phenomenon. First, ordered constraints can be placed on the parameters to enforce identifiability. While simple, this approach has been criticized for biasing the resulting parameter estimates. Second, relabeling algorithms have been proposed to choose optimal labeling schemes (Jasra, Holmes, and Stephens 2005; Stephens 2000). Post-processing of the MCMC samples was used in this thesis rather than ordered constraints since ordered constraints would force a handful of topics to dominate a given corpus and yielded excellent topic recovery as shown in Table 7.1.

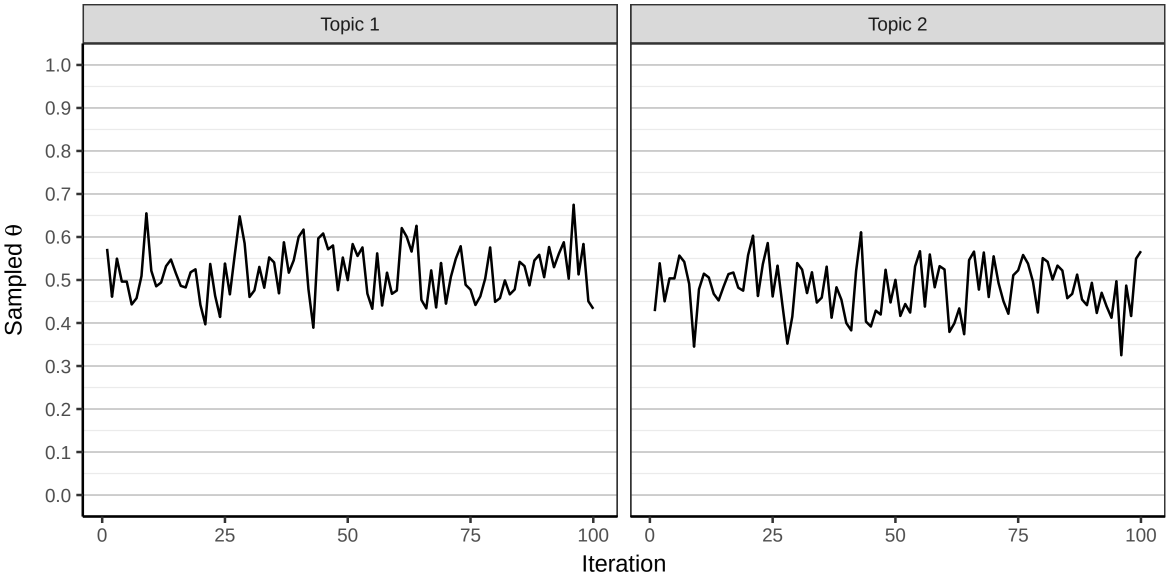

Figure 7.10: Sample path of topic proportions in the first document of simulated data set 7.

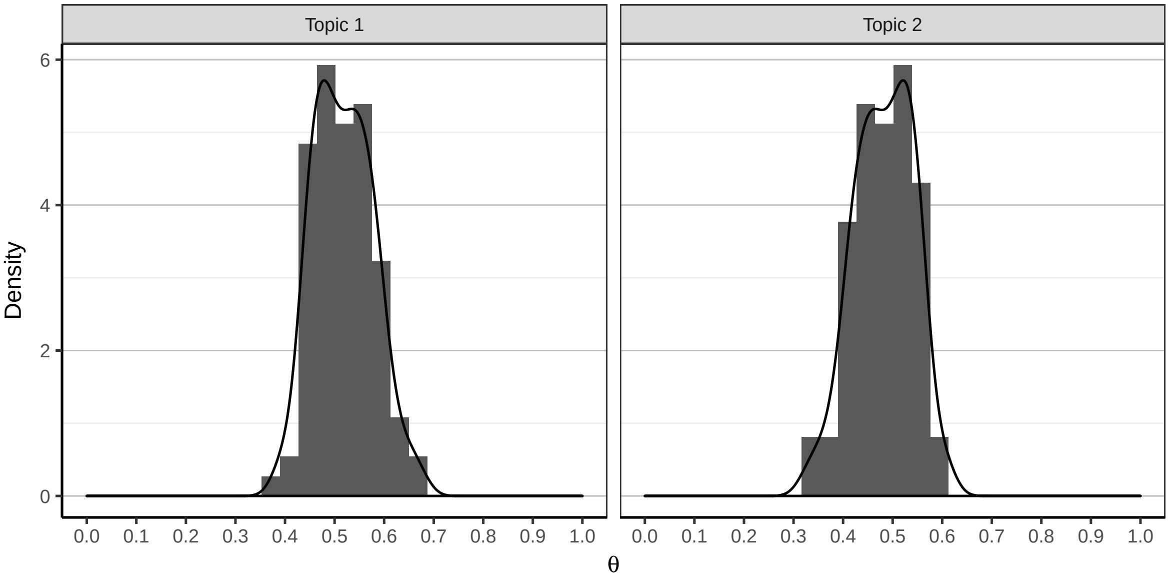

Figure 7.11: Posterior distribution of topic proportions in the first document from simulated data set 7.

The sample path of β in Figure 7.12 for topic-word probabilities of the first word for topics 1 and 2 reveals that the larger size of the corpus in data set 7 relative to data set has allowed the sampler to converge. It is clear the sample path for topic 1 favors draws for small values of β_{11} near 0.00045 while the sample path for topic 2 favors draws for values of β_{21} near the prior mean of 0.001.

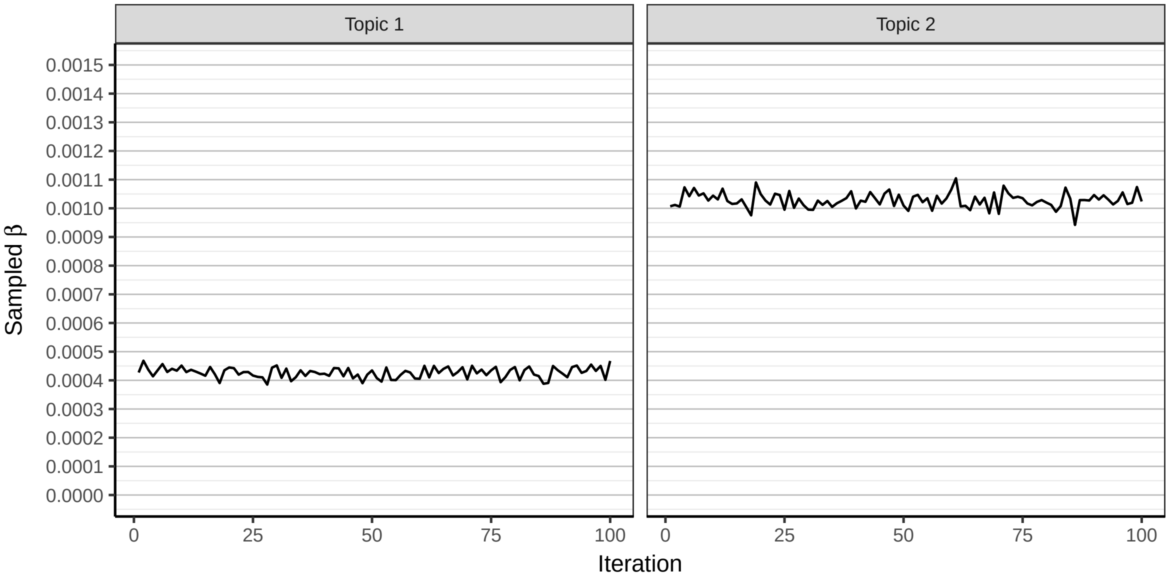

Figure 7.12: Sample path of topic-word probabilities for the first word from simulated data set 7.

The posterior distributions shown in Figure 7.13 for both components of β are both unimodal, but the distribution for topic 2 is centered at the prior mean of 0.001 while the distribution for topic 1 is shifted to the left of the prior mean near 0.0005. This suggests that the first word was drawn from the first topic very rarely, but was more likely to be drawn from the second topic. Compared to the posterior distributions for these parameters in data set 1, there is very little variability for both distributions which reflects the larger amount of information available in the data set.

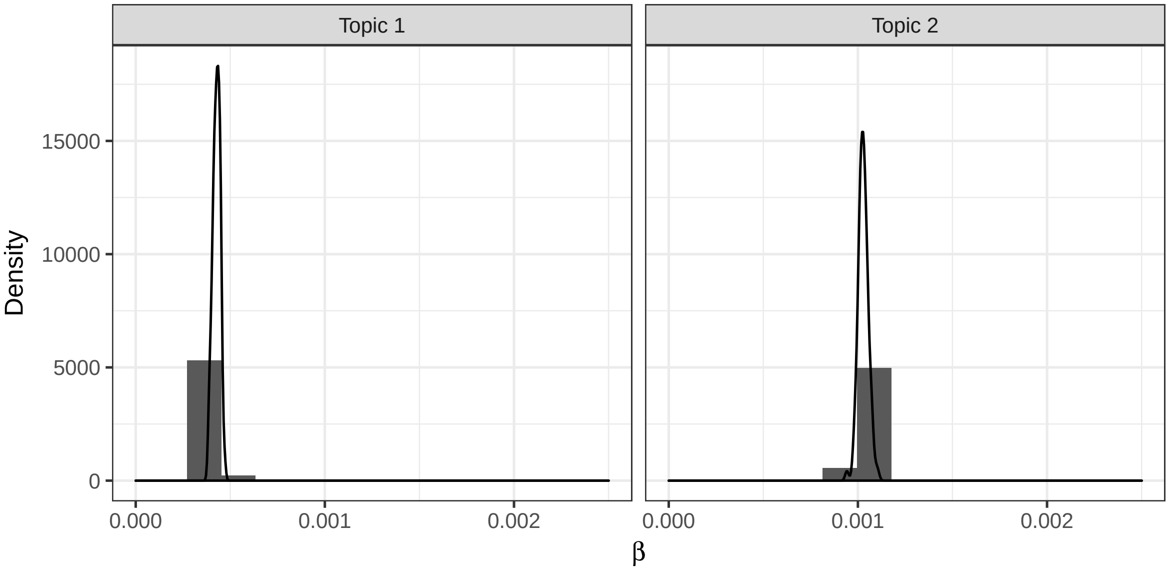

Figure 7.13: Posterior distributions of topic-word probabilities for the first word from simulated data set 7.

7.2 A Comparison of LDA and HTMM on the NIPS Corpus

The performance of the Hidden Topic Markov Model was contrasted with Latent Dirichlet Allocation (LDA) on a corpus of NIPS (Neural Information Processing Systems) conference proceedings. Following Gruber, Rosen-Zvi, and Weiss (2007), $K = 100$ topics were considered with priors on $\theta_d \sim$ Dirichlet(1.5) for $d \in \{0, 1, \ldots, 1557\}$, $\beta_k \sim$ Dirichlet(1.01) for $k \in \{0, 1, \ldots, 99\}$, and $\epsilon \sim$ Beta(1, 1) on a corpus of $D_{train} = 1557$ randomly selected journal articles with a vocabulary of $V = 12113$ words after removing stop words. The text was very simply tokenized into sentences using ".", "?", "!", and ";" as delimiters. All appearances of "e.g." and "i.e." were also removed due to their frequent use in journal articles. A randomly selected test set of $D_{test} = 180$ was held out for use in evaluating perplexity.

The EM HTMM algorithm was run on the training set for 1000 iterations, although convergence was reached within 10 iterations as shown in Figure 7.14. An estimated $\hat{\epsilon} = 0.291$ suggests that the topics in the training set were relatively contiguous since $\hat{\epsilon}$, the estimated probability of a topic transition, was substantially smaller than 0.5. Recall that if $\epsilon = 1$, topics transition on every token, which, if the token chosen is a word, yields the LDA model. If $\epsilon = 0$, topics never transition and documents are represented by a single topic as in the mixture of unigrams model. A variational EM algorithm for LDA was run until convergence at 37 iterations.

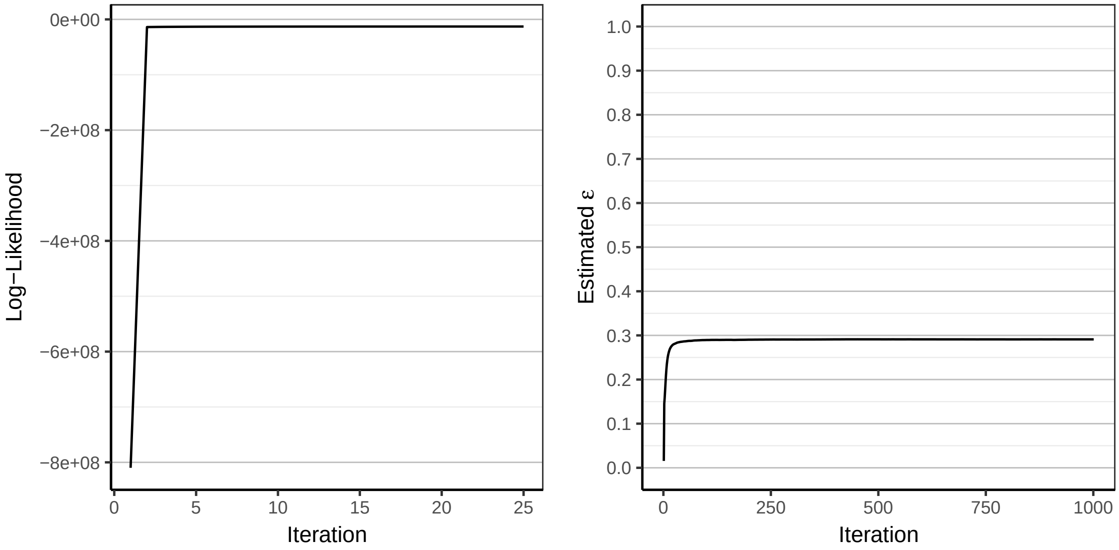

Figure 7.14: Convergence of log-likelihood for HTMM-EM algorithm (left). Convergence of estimated epsilon parameter (right)

A sample of the ten most probably words for four topics obtained from the LDA model are shown in Figure 7.15. These topics are nicely interpretable and seem to represent hidden Markov models, neuroscience, function approximation, and circuit design (clockwise from top left). While these topics are coherent, there are some words like “figure”, “data”, and “networks” that are not as thematically coherent with the interpretation of the topics. This is particularly undesirable because LDA models do not typically favor assignment of a word to multiple topics. Since the three words highlighted above are germane to practically any article in the corpus, their high probabilities in these four topics will likely result in these topics being assigned to these words in documents and sections of documents that are not discussing hidden Markov models, neuroscience, kernel functions, and circuit design.

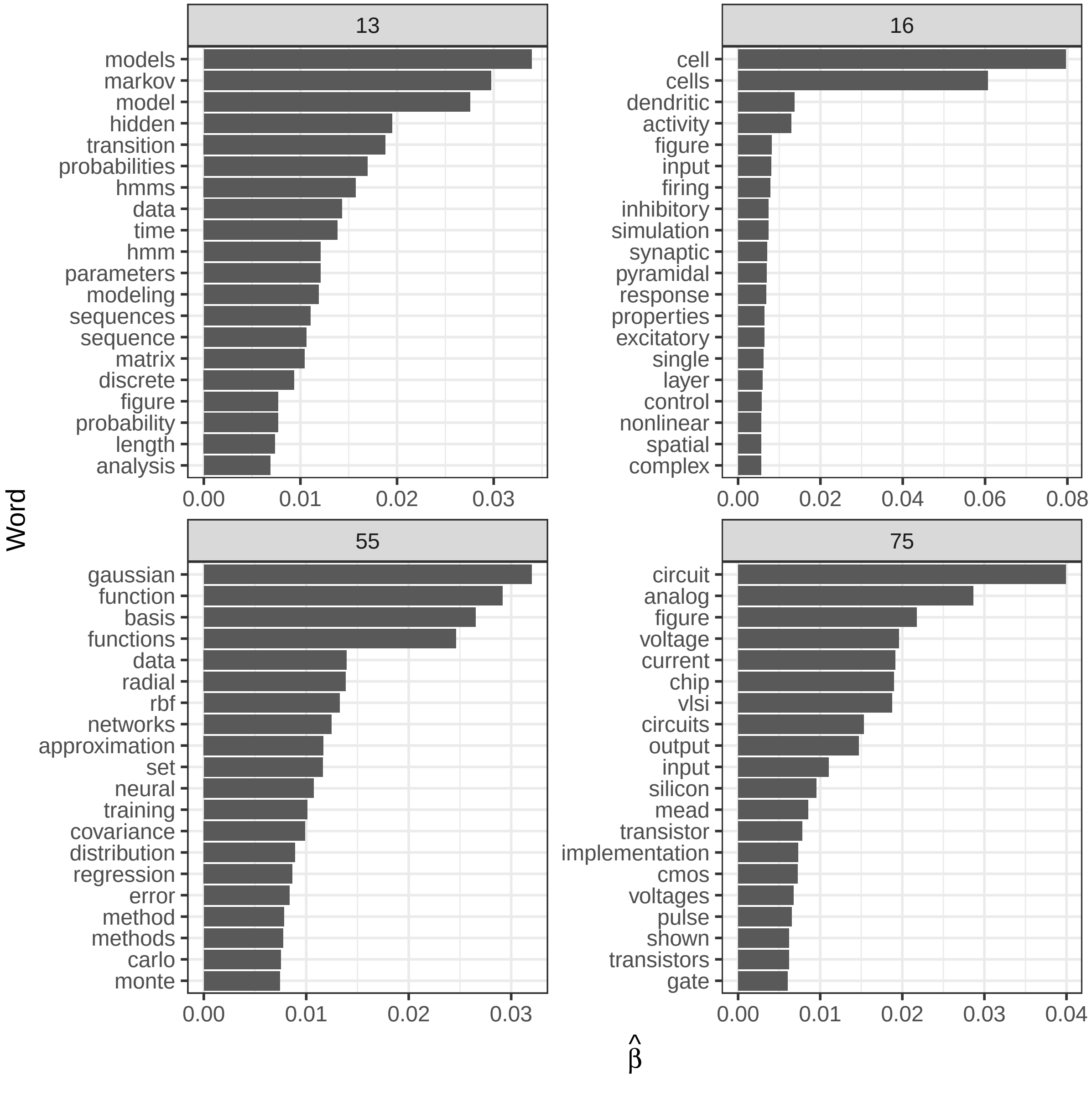

Figure 7.15: The ten most probable words for topics 13, 16, 55, and 75 from the NIPS LDA model

For comparison, a sample of the ten most probable words from four topics obtained from the HTMM are shown in Figure 7.16. Words within a sentence are now constrained to a single topic now, whereas LDA induced multiple topics within a single sentence. These topics are also clearly interpretable. Topic 7 can be interpreted as a neural network topic, topic 51 can be interpreted as a support vector machine topic (SVM), and topic 63 has captured reference and acknowledgement sections. Examination of topic 51 and topic 63 demonstrates the disambiguation of two word senses for *support*: *support* vector machine and funding *support*. Topic 79 is very interesting since it has captured professional affiliations of the authors; examination of the 100 LDA topics did not reveal a similar topic. Instead, many topics were incoherent and featured some professional affiliation words among other topical words. This suggests that an HTMM is capable of learning more semantically coherent topics than LDA and, furthermore, is capable of generating contiguous sequences of topics in a document; this result is less common using LDA.

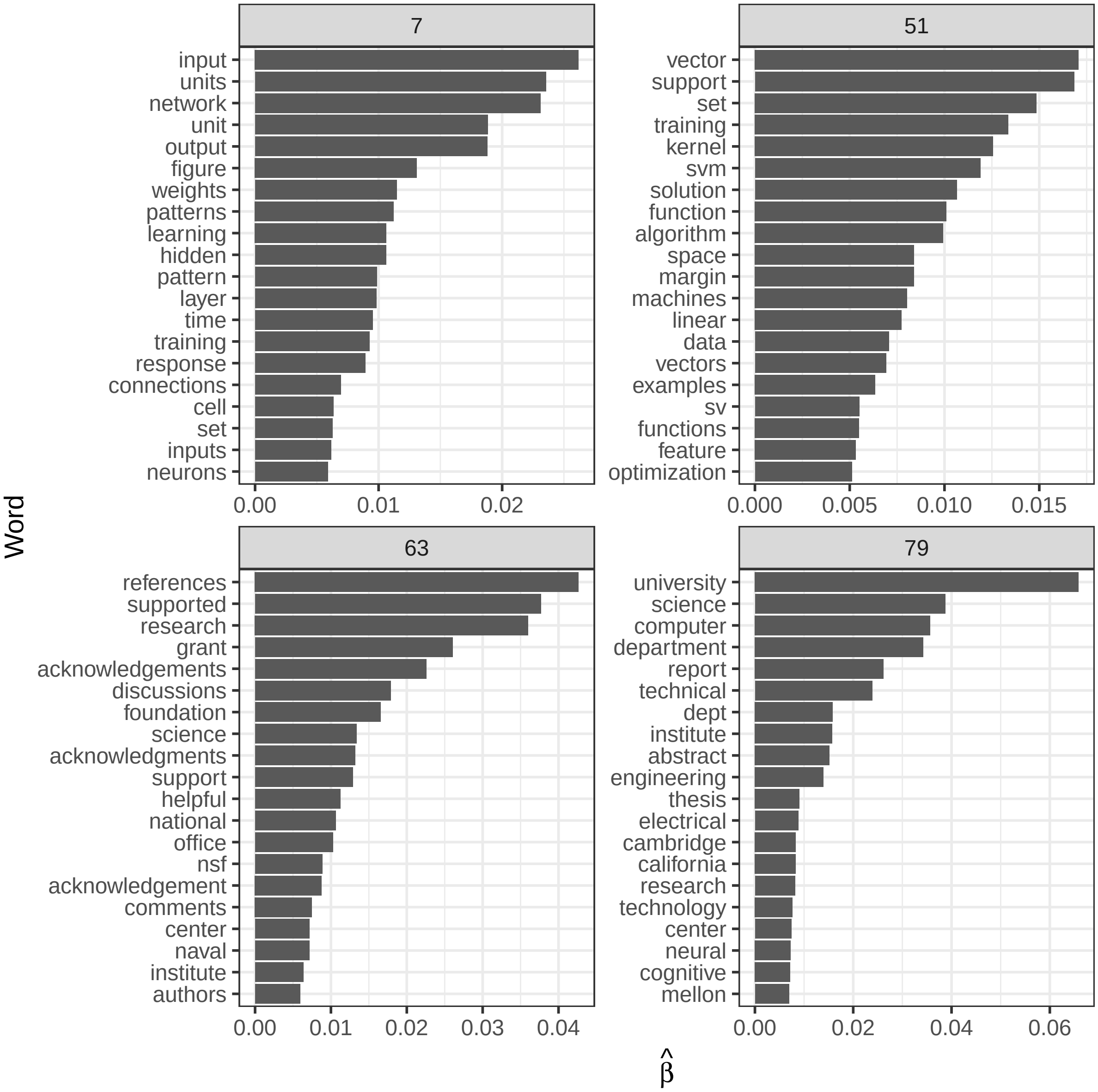

Figure 7.16: Sample topics from NIPS HTMM model

The documents can also be represented by their distribution over the 100 topics. As shown in Figure 7.17, LDA represents some documents using only a few topics, while other documents are represented with a larger set of topics.

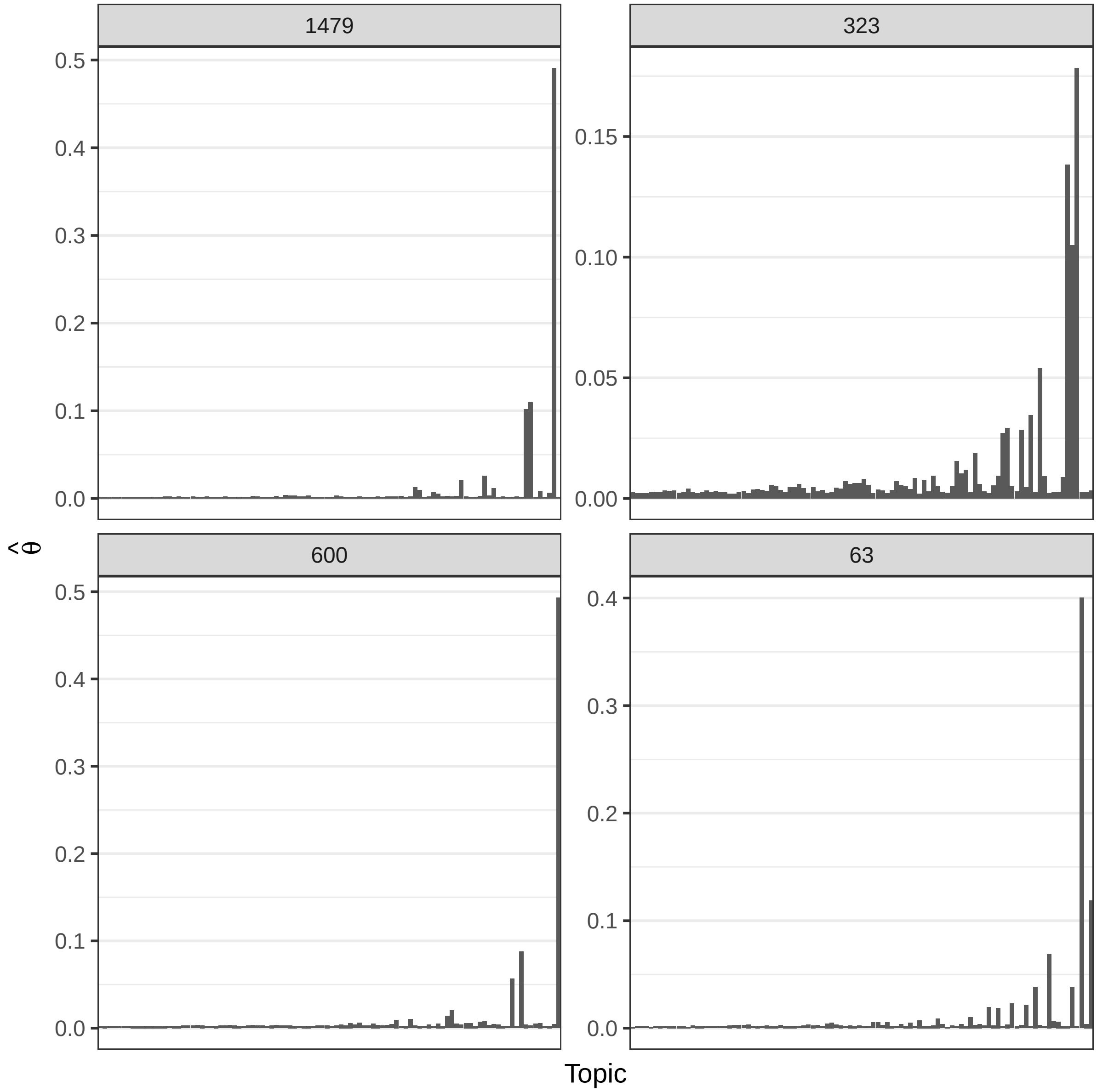

Figure 7.17: Sample topic proportions from NIPS LDA model

Interestingly, the same documents are represented with a richer set of topics using HTMM. As seen in Figure 7.18, documents 63, 600, and 1479 in particular are composed of many more topics with HTMM than with LDA while HTMM assigned document 323 fewer topics than LDA did.

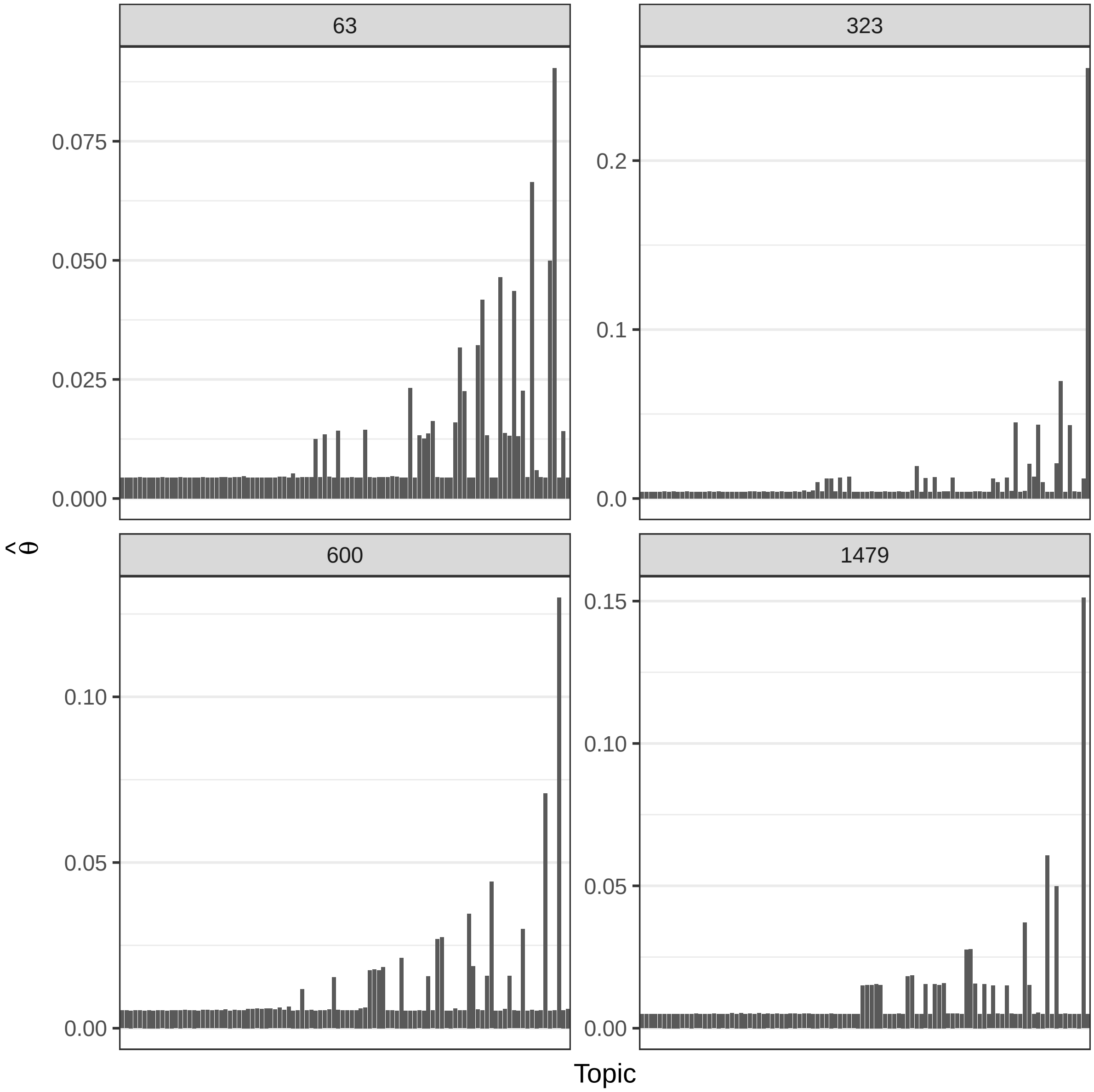

Figure 7.18: Sample topic proportions from NIPS LDA model

Finally, the predictive performance of LDA and HTMM were compared using perplexity. Perplexity of a corpus $\mathcal{D}$ of M documents was computed as

$$P(\mathcal{D}) = \exp\left\{ \frac{-\sum_{d=0}^{M-1} \log p(w_d; \lambda)}{\sum_{d=0}^{M-1} N_d} \right\} \tag{7.1}$$

where w_d is the vector of words in document d, N_d is the number of words in document d, and λ is a vector of model parameters. The perplexity of a corpus decreases monotonically as the likelihood of a corpus increases. Therefore, lower perplexity is an indicator of better generalization error. LDA had a perplexity of 1460.1 on the training set and a perplexity of 1952.7 on the test set. In comparison, HTMM had a perplexity of 1159.8 on the training set and 1157.3 on the test set. The perplexity for HTMM was lower than that of LDA for both the training and test sets which suggests that HTMM provided a better fit to the NIPS corpus than LDA through its use of Markovian dynamics. It was surprising that HTMM achieved very similar perplexity scores on both the training and testing set. This suggests that the HTMM model was not over-fit and that the model performed not only better than LDA for both training and testing sets, but equally well with new data as with the training data. The choice of a 90%/10% train-test split of the corpus was made in keeping with similar decisions made by other researchers such as Blei, Ng, and Jordan (2003) and Gruber, Rosen-Zvi, and Weiss (2007). It is possible that the use of such a large proportion of the corpus for training is responsible for the performance of HTMM on the test set, although this was not the case for LDA. Further evaluation with different ratios of training and test data would help understand these results.

Chapter 8

Conclusions

This thesis successfully extended the Hidden Topic Markov Model (HTMM) proposed by Gruber, Rosen-Zvi, and Weiss (2007) from a purely frequentist framework into a fully Bayesian framework. First, the state space used by the hidden Markov model embedded in the HTMM was elucidated to facilitate the first published derivation of the expectation-maximization (EM) algorithm used by Gruber, Rosen-Zvi, and Weiss. The necessary forward-backward algorithm was derived for the first time, filling in a crucial missing piece required for both frequentist and Bayesian inference that was never formally derived. In order to perform inference for the state space when using the EM algorithm, a modification of the Viterbi algorithm proposed by Gruber and Popat (2007) was derived that properly respects state transition restrictions when using sentences as a topical unit since the algorithm proposed by Gruber and Popat is only appropriate when the topical unit is a word.

Using the forward-backward algorithm derived in Chapter 5, full conditional distributions for the HTMM parameters ϵ, θ, β, and latent states s were derived. Equipped with these distributions, a Gibbs sampling algorithm was proposed in Chapter 6.

Both the EM and Gibbs sampling algorithms were implemented in the `R` and `C++` programming languages. The performance of both algorithms was assessed on twelve simulated data sets in a study of the impact of the number of sentences, number of topics, and transition probability ϵ on the accuracy of parameter estimation and topic recovery. The results of the simulation study suggested that both algorithms perform comparably in estimating the model parameters, but the Gibbs sampling algorithm dramatically outperformed the combination of the EM and Viterbi algorithms in recovering the topic space. Furthermore, both algorithms were shown to converge relatively quickly in the presence of moderately sized data sets. Assessment of the Gibbs sampling algorithm revealed that the prior distributions are dominant for small data sets, but that larger data sets yield precise posterior distributions that capture the true generative parameters well. Label switching was observed with both the EM and Gibbs algorithms and future work to address this phenomenon would be worthwhile.

Finally, the Hidden Topic Markov Model (HTMM) was compared with the popular Latent Dirichlet Allocation (LDA) on a corpus of published proceedings from the Conference on Neural Information Processing Systems (NIPS). Topical assignments were interpretable using both models, but the topic assignments extracted by the HTMM were more contiguous and subjectively appeared to be more coherent than those obtained by LDA. Predictive performance was assessed using perplexity: lower perplexity was observed for both training and test corpora for the HTMM than for LDA. Furthermore, HTMM demonstrated nearly identical perplexity on the test corpus as the training corpus, suggesting that the HTMM model generalized very well to unseen documents. Future work to optimize the speed and memory usage of the HTMM EM and Gibbs sampling algorithms would be worthwhile as both algorithms require longer computing times for large corpora and large quantities of memory in the case of the Gibbs sampler. A theoretical study of the HTMM model is recommended to better understand the quality of inference in the presence of small corpora and asymptotic properties as the number of documents, topics, and the vocabulary size grow large. Nonparametric extensions to learn the number of topics could be considered. Finally, human evaluation of the quality of topics extracted by the HTMM and other topic models like LDA is vitally important to assess the interpretability and linguistic utility of these models.

Bibliography

Anandkumar, Anima, et al. 2012. “A spectral algorithm for Latent Dirichlet Allocation”. *Advances in Neural Information Processing Systems* 25:917–925.

Andrews, Mark. 2013. “Probabilistic language modeling with hidden stochastic automata”. In *Proceedings of the 35th Annual Meeting of the Cognitive Science Society*, 1750–1755.

Andrews, Mark, and Gabriella Vigliocco. 2010. “The hidden Markov topic model: A probabilistic model of semantic representation”. *Topics in Cognitive Science* 2 (1): 101–113. doi:10.1111/j.1756-8765.2009.01074.x.

Andrieu, Christophe, et al. 2003. “An introduction to MCMC for machine learning”. *Machine Learning* 50:5–43.

Baum, Leonard E., and Ted Petrie. 1966. “Statistical inference for probabilistic functions of finite state Markov chains”. *The Annals of Mathematical Statistics* 37:1554–1563.

Baum, Leonard E., et al. 1970. “A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains”. *The Annals of Mathematical Statistics* 41 (1): 164–171. http://www.jstor.org/stable/2239727.

Blei, David. 2012. “Probabilistic topic models”. *Communications of the ACM* 55 (4): 77–84. doi:10.1145/2133806.2133826.

Blei, David M., and John D. Lafferty. 2007. “A correlated topic model of Science”. *The Annals of Applied Statistics* 1 (1): 17–35. doi:10.1214/07-AOAS136.

— . 2005. “Correlated topic models”. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, 147–154. Vancouver, BC, Canada: MIT Press.

— . 2006. “Dynamic topic models”. In *Proceedings of the 23rd International Conference on Machine Learning*, 113–120. Pittsburgh, PA.

Blei, David M., and Pedro J. Moreno. 2001. *Topic segmentation with an aspect hidden Markov model.* Tech. rep. Cambridge, MA: Cambridge Research Laboratory.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. “Latent Dirichlet Allocation”. *Journal of Machine Learning Research* 3:993–1022. doi:10.1162/jmlr.2003.3.4-5.993.

Bouveyron, C., P. Latouche, and R. Zreik. 2016. “The stochastic topic block model for the clustering of vertices in networks with textual edges”. *Statistics and Computing*: 1–21. doi:10.1007/s11222-016-9713-7.

Boyd-Graber, Jordan, and David Blei. 2009. “Syntactic topic models”. In *Advances in Neural Information Processing Systems*, 21:185–192.

Boyd-Graber, Jordan, David Mimno, and David Newman. 2014. “Care and feeding of topic models: Problems, diagnostics, and improvements”. Chap. 12 in *Handbook of Mixed Membership Models and Its Applications*, ed. by Edoardo M. Airoldi et al., 225–254. Boca Raton, FL: Chapman / Hall. ISBN: 9781466504080.

Cappé, Olivier, Eric Moulines, and Tobias Rydén. 2005. *Inference in Hidden Markov Models.* 48:574–575. New York, NY: Springer Science+Business Media. ISBN: 9780387402642.

Chen, Harr, et al. 2009. “Global models of document structure using latent permutations”. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 371–379. ISBN: 9781932432411.

Deerwester, Scott, et al. 1990. “Indexing by latent semantic analysis”. *Journal of the American Society for Information Science* 41 (6): 391–407.

Dempster, A. P., N. M. Laird, and Donald B. Rubin. 1977. “Maximum likelihood from incomplete data via the EM algorithm”. *Journal of the Royal Statistical Society, Series B (Methodological)* 39 (1): 1–38.

Gelfand, Alan E., and Adrian F. M. Smith. 1990. “Sampling-based approaches to calculating marginal densities”. *Journal of the American Statistical Association* 85 (410): 398–409.

Green, Peter J. 1995. “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination”. *Biometrika* 82 (4): 711–732. doi:10.1093/biomet/82.4.711.

Griffiths, Thomas L., and Mark Steyvers. 2002. “A probabilistic approach to semantic representation”. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, 381–386. Fairfax, VA.

— . 2004. “Finding scientific topics”. *Proceedings of the National Academy of Sciences of the United States of America* 101 (Supplemental 1): 5228–35. doi:10.1073/pnas.0307752101.

Griffiths, Thomas L., Mark Steyvers, and Joshua B. Tenenbaum. 2007. “Topics in semantic representation”. *Psychological Review* 114 (2): 211–244. doi:10.1037/0033-295X.114.2.211.

Gruber, Amit, and Ashok C. Popat. 2007. *Notes Regarding Computations in OpenHTMM.* Tech. rep. Google, Inc.

Gruber, Amit, Michal Rosen-Zvi, and Yair Weiss. 2007. “Hidden topic Markov models”. In *Proceeding of the International Conference on Artificial Intelligence and Statistics*, 163–170.

Hastings, W. K, and Hastings. W. K. 1970. “Monte Carlo sampling methods using Markov chains and their applications”. *Biometrika* 57 (1): 97–109. doi:10.1093/biomet/57.1.97.

Hofmann, Thomas. 1999. “Probabilistic latent semantic indexing”. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 50–57. Berkeley, CA: ACM. doi:10.1145/312624.312649.

Jasra, A., C. C. Holmes, and D. A. Stephens. 2005. "Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling". *Statistical Science* 20 (1): 50–67. doi:`10.1214/088342305000000016`.

Levinson, S. E., L. R. Rabiner, and M. M. Sondhi. 1983. "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition". *The Bell System Technical Journal* 62 (4): 1035–1074. doi:`10.1002/j.1538-7305.1983.tb03114.x`.

MacKay, David J. C., and Linda C. Bauman Peto. 1995. "A hierarchical Dirichlet language model". *Natural Language Engineering* 1 (3): 289–307.

Metropolis, Nicholas, et al. 1953. "Equation of state calculations by fast computing machines". *The Journal of Chemical Physics* 21 (6): 1087–1092. doi:`10.1063/1.1699114`.

Mimno, David, et al. 2011. "Optimizing semantic coherence in topic models". In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 262–272. 2. Edinburgh, UK: Association for Computational Linguistics. ISBN: 9781937284114.

Nigam, Kamal, et al. 2000. "Text classification from labeled and unlabeled documents using EM". *Machine Learning* 39:103–134.

Porteous, Ian, et al. 2008. "Fast collapsed Gibbs sampling for latent Dirichlet allocation". In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 569–577. Las Vegas, NV: ACM. doi:`10.1145/1401890.1401960`.

Rabiner, L.R. 1989. "A tutorial on hidden Markov models and selected applications in speech recognition". *Proceedings of the IEEE* 77 (2): 257–286.

Rosen-Zvi, Michal, et al. 2010. "Learning author-topic models from text corpora". *ACM Transactions on Information Systems* 28 (1): 38. doi:`10.1145/1658377.1658381`.

Rydén, Tobias. 2008. "EM versus Markov chain Monte Carlo for estimation of hidden Markov models: a computational perspective". *Bayesian Analysis* 3, no. 4 (): 659–688. doi:`10.1214/08-BA326`.

Salton, Gerard, and M. J. McGill. 1983. *Introduction to modern information retrieval.* 400. ISBN: 0070544840.

Stephens, M. 2000. "Dealing with label switching in mixture models". *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* 62 (4): 795–809. doi:10.1111/1467-9868.00265.

Viterbi, Andrew J. 1967. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". *IEEE Transactions on Information Theory* 13 (2): 260–269.

Wallach, Hanna M. 2006. "Topic modeling: Beyond bag-of-words". In *Proceedings of the 23rd International Conference on Machine Learning*, 977–984. Pittsburgh, PA: ACM. doi:10.1145/1143844.1143967.

Wan, Li, Leo Zhu, and Rob Fergus. 2012. "A hybrid neural network-latent topic model". *Journal of Machine Learning Research* 22:1287–1294.

Wang, Xuerui, and Andrew McCallum. 2006. "Topics over time: A non-Markov continuous-time model of topical trends". In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 424–433. Philadelphia, PA: ACM. ISBN: 1595933395. doi:10.1145/1150402.1150450.

Yu, Xingchen, and Ernest Fokoue. 2014. "Probit normal correlated topic models". *Open Journal of Statistics* 4:879–888. doi:10.4236/ojs.2014.411083.

Zhang, Aonan, Jun Zhu, and Bo Zhang. 2013. "Sparse relational topic models for document networks". In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 670–685. Prague, Czech Republic: Springer Berlin Heidelberg. doi:10.1007/978-3-642-40988-2_43.

Zhu, Jun, et al. 2014. "Gibbs max-margin topic models with data augmentation". *Journal of Machine Learning Research* 15:1073–1110.