UNIVERSITY OF CALIFORNIA,
IRVINE

Latent Variable Modeling for Networks and Text: Algorithms, Models and Evaluation Techniques

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

James Richard Foulds

Dissertation Committee:
Professor Padhraic Smyth, Chair
Professor Alexander Ihler
Professor Mark Steyvers

2014

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I would like to begin by thanking my advisor, Padhraic Smyth. He has taught me a great deal about probabilistic and statistical machine learning, and also about every other aspect of research. His guidance, support and mentorship have been invaluable throughout the entire process which led to this dissertation. Whenever I am stuck while writing an article, preparing a presentation, or choosing a research direction, I just ask myself "what would Padhraic do?" and the answer often comes to me. One could not ask for a better advisor.

I have been very fortunate to have an excellent committee and a talented group of collaborators and colleagues. I am grateful to my candidacy and final defense committee members Alex Ihler, Max Welling, Carter Butts, Mark Steyvers and Rina Dechter for their time, effort and insightful comments. Thank you to my collaborators Levi Boyles, Chris DuBois, Arthur Asuncion, Nick Navaroli, Max Welling, Alex Ihler and Carter Butts for all your hard work and many useful discussions. My group members and UCI machine learning contemporaries have been great colleagues and friends – thank you to Andrew Frank, America Chambers, Chris Dubois, Corey Shaninger, Jon Hutchins, Chaitanya Chemudugunta, Scott Triglia, Scott Crawford, Kevin Bache, Moshe Lichmann, Nick Navaroli, Tracy Holsclaw, Ralf Krestel, Dilan Görür, Romain Thibaux, Andrew Gelfand, Dave Orendorff, Lars Otten, Todd Johnson, Darren Davis, Eric Nalisnick, Zach Butler, Geng Ji, and others.

On a more personal note, it is difficult to imagine how my life would have been at UCI without my wonderful group of friends. The "Spoons on the Beach" are dear to my heart – Kevin, Kyle, Robin, Olivier, Mary, Joanna, Mingxia, Corey, Yue, Sky, Gabe, Nick, Levi, Nicki, Moshe, Eugenia, Katherine, Kristin, Medhi, Zhen, ZJ, and honorary "Spoons" Caitlin, Amber, Maeraj and Michael: thank you for all the good times, and all the support in the harder times. I love you all. Thank you to my drum team Hansori – may your "one sound" continue to reverberate strongly. Thanks to Kevin, Kyle, Nick and Andy for being amazing roommates and friends, and for all the delicious breakfasts. Special thanks go to my parents Maureen and Les, and my sister Susie, for your love and support. Finally, thank you to Rosie for your patience, understanding and support, especially in the final months of the preparation of this dissertation. To all of you, I am *so* grateful for everything.

This dissertation was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20155, the Office of Naval Research under MURI grant N00014-08-1-1015, and a UC Irvine ICS Dean's Fellowship.

CURRICULUM VITAE

James Richard Foulds

EDUCATION

Doctor of Philosophy in Computer Science — **2014**
University of California, Irvine — *Irvine, California*

Master of Science (First Class Honours) in Computer Science — **2008**
University of Waikato — *Hamilton, New Zealand*

Bachelor of Computing and Mathematical Sciences (First Class Honours) — **2006**
Specialization in Artificial Intelligence. GPA: 8.92/9.0 (A+ average)
University of Waikato — *Hamilton, New Zealand*

REFEREED PUBLICATIONS

J. R. Foulds, P. Smyth. **Annealing Paths for the Evaluation of Topic Models**. Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence, 2014.

J. R. Foulds, P. Smyth. **Modeling scientific impact with topical influence regression**. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2013.

J. R. Foulds, L. Boyles, C. DuBois, P. Smyth and M. Welling. **Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation**. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2013.

C. DuBois, J. R. Foulds, P. Smyth. **Latent set models for two-mode network data**. Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM), 2011.

J. R. Foulds, A. Asuncion, C. DuBois, C. T. Butts, P. Smyth. **A dynamic relational infinite feature model for longitudinal social networks**. Proceedings of the 14th International Conference on AI and Statistics (AI Stats), 2011.

J. R. Foulds, N. Navaroli, P. Smyth, A. Ihler. **Revisiting MAP estimation, message passing and perfect graphs**. Proceedings of the 14th International Conference on AI and Statistics (AI Stats), 2011.

J. R. Foulds and P. Smyth. **Multi-instance mixture models and semi-supervised learning**. SIAM International Conference on Data Mining, 2011.

J. R. Foulds and E. Frank. **Speeding up and boosting diverse density learning**. Proceedings of the 13th International Conference on Discovery Science, pages 102-116. Springer, 2010.

J. R. Foulds and E. Frank. **A review of multi-instance learning assumptions**. Knowledge Engineering Review, 25(1):1-25, 2010.

J. R. Foulds and E. Frank. **Revisiting multiple-instance learning via embedded instance selection**. In Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence, Auckland, New Zealand. Springer, 2008.

J. R. Foulds and L. R. Foulds. **A probabilistic dynamic programming model of rape seed harvesting**. International Journal of Operational Research 2006, Vol. 1, No. 4, 2006.

J. R. Foulds and L. R. Foulds. **Bridge lane direction specification for sustainable traffic management**. Asia-Pacific Journal of Operational Research, Vol. 23, No. 2, 2006.

WORKSHOP PUBLICATIONS

J. R. Foulds and P. Smyth. **Robust Evaluation of Topic Models**. In NIPS Workshop on Topic Models, 2013.

J. R. Foulds and D. Görür. **Diverse personalization with determinantal point process eigenmixtures**. In NIPS Workshop on Personalization, 2013.

J. R. Foulds and P. Smyth. **Modeling scientific impact with topical influence regression**. In NIPS Workshop on Algorithmic and Statistical Approaches for Large Social Network Data Sets, 2012.

For a complete and up-to-date curriculum vitae, please contact the author of this dissertation.

ABSTRACT OF THE DISSERTATION

Latent Variable Modeling for Networks and Text:
Algorithms, Models and Evaluation Techniques

By

James Richard Foulds

Doctor of Philosophy in Computer Science

University of California, Irvine, 2014

Professor Padhraic Smyth, Chair

In the era of the internet, we are connected to an overwhelming abundance of information. As more facets of our lives become digitized, there is a growing need for automatic tools to help us find the content we care about. To tackle the problem of information overload, a standard machine learning approach is to perform dimensionality reduction, transforming complicated high-dimensional data into a manageable, low-dimensional form. Probabilistic latent variable models provide a powerful and elegant framework for performing this transformation in a principled way. This thesis makes several advances for modeling two of the most ubiquitous types of online information: networks and text data.

Our first contribution is to develop a model for social networks as they vary over time. The model recovers latent feature representations of each individual, and tracks these representations as they change dynamically. We also show how to use text information to interpret these latent features.

Continuing the theme of modeling networks and text data, we next build a model of citation networks. The model finds influential scientific articles and the influence relationships between the articles, potentially opening the door for automated exploratory tools for scientists.

The increasing prevalence of web-scale data sets provides both an opportunity and a challenge. With more data we can fit more accurate models, as long as our learning algorithms are up to the task. To meet this challenge, we present an algorithm for learning latent Dirichlet allocation topic models quickly, accurately and at scale. The algorithm leverages stochastic techniques, as well as the collapsed representation of the model. We use it to build a topic model on 4.6 million articles from the open encyclopedia Wikipedia in a matter of hours, and on a corpus of 1740 machine learning articles from the NIPS conference in seconds.

Finally, evaluating the predictive performance of topic models is an important yet computationally difficult task. We develop one algorithm for comparing topic models, and another for measuring the progress of learning algorithms for these models. The latter method achieves better estimates than previous algorithms, in many cases with an order of magnitude less computational effort.

Chapter 1

Introduction

> It's the job that's never started as takes longest to finish.
>
> *J.R.R. Tolkien, the Fellowship of the Ring*

In recent decades the amount of digital information available on the internet has been increasing at an astounding rate. As consumers of information, we are inundated on a daily basis with overwhelming amounts of content in the form of news websites, online encyclopedias, weblogs, social media, photos, music, streaming video and more.

Furthermore, we as a society are becoming increasingly connected to the internet throughout our daily lives. The data traffic from mobile devices alone in 2013 was close to 18 times the total internet traffic in the year 2000 (Cisco Systems, 2014). Cisco projects that the number of mobile-connected devices will exceed the earth's population by the end of this year (2014), and that traffic from wearable devices will increase 36-fold by 2018. With more digital information being available, and greater portions of our lives being connected to it, information becomes increasingly pertinent to the human condition. An important consequence is that as internet users consume content, they also irrevocably consume even

more valuable resources: their time and their attention. This motivates the use of automatic methods to understand, summarize and recommend this content.

Scientists have even stronger needs for such automatic tools than the average digital consumer. The overall number of scientific publications has been estimated to grow exponentially with a growth rate of about 4.7% per year (Price, 1963), with no sign of slowing (Larsen & von Ins, 2010), motivating analysis tools and recommender systems for scientific articles, e.g. (Wang & Blei, 2011; El-Arini & Guestrin, 2011). More directly, the growing field of the *digital humanities* seeks to employ computational tools to advance research in the disciplines of the humanities. Examples include modeling ancient Roman households based on databases of artifacts discovered by archaeologists (Mimno, 2011), and modeling social networks, digital or otherwise (Nowicki & Snijders, 2001; Kemp *et al.* , 2006).

A standard approach to the problem of information overload is to perform dimensionality reduction, transforming complicated high-dimensional data into a smaller, more manageable representation. Ideally, the transformed low-dimensional representations are semantically meaningful and able to be understood directly by humans. This idea dates back at least as far as the principal components analysis (PCA) technique of Pearson (1901) and Hotelling (1933), which seeks to re-represent data in fewer dimensions while minimizing the total squared error on reconstruction.

PCA continues to enjoy widespread use, over a century after Pearson's original article was published. However, its squared error minimization objective corresponds to an implicit Gaussian noise assumption which is not always appropriate for every data set (Hofmann, 1999b; Buntine & Jakulin, 2006). For instance, PCA can be applied to text data, in a technique known as latent semantic analysis (Deerwester *et al.* , 1990). In this scenario a document is represented as a discrete vector of "bag of words" term counts. Typically the document vectors are very sparse, meaning that the large-sample approximation of binomially distributed count data by a Gaussian is not accurate. Buntine & Jakulin (2006) show

that the implicit Gaussian assumption of PCA may consequently result in issues such as greatly over-estimating the probability of rare events. The resulting low-dimensional representations are also not as interpretable as we would like, as our positive integer-valued count vectors are transformed into continuous vectors with potentially negative entries, and as such cannot be interpreted as "typical" documents (Buntine & Jakulin, 2006). Furthermore, many words have multiple senses – a phenomenon known as *polysemy*. LSA does not explicity encode this, which limits its usefulness for word disambiguation and gist extraction (Griffiths *et al.* , 2007).

What is needed, then, is a set of techniques which can obtain meaningful low-dimensional representations of data as in PCA, but which support alternative statistical assumptions. This is achieved by a class of probabilistic models known as *latent variable models* (cf. Bishop (1998)). These models represent a distribution on the observed variables of interest by way of additional posited *latent* (hidden, unobserved) variables. The assumed generative process of the model can be chosen to encode appropriate distributional assumptions. Latent variable models produce (potentially) low-dimensional representations of the data by way of the latent variables and model parameters themselves, which are inferred from the data using statistical techniques.

Returning to our example of bag-of-words text data, the latent variable model called probabilistic LSA (PLSA) (Hofmann, 1999b,a) and its Bayesian extension, LDA (Blei *et al.* , 2003), use multinomial distributional assumptions. These modeling assumptions are more appropriate to sparse, discrete count data than the implicit Gaussian assumption used by LSA, leading to better predictive performance and more interpretable latent representations (Hofmann, 1999b,a; Griffiths *et al.* , 2007).

To use latent variable models in practice, we need to

1. define models applicable to the tasks at hand,
2. develop learning algorithms to fit them, and
3. implement evaluation strategies for validating the models.

This thesis makes advances on each of these fronts. The key contributions of the thesis are two new latent variable models, and two novel algorithmic methods. The proposed techniques focus on network and text data sets, which are perhaps the most prevalent types of information available for internet and digital humanities applications. Furthermore, some of the methods introduced here explore the interplay of *both* of these types of digital information in the case where network and text data are available together. In addition, many of the ideas explored here have the potential to be applied more generally to other types of data.

We first consider the problem of modeling social networks in a longitudinal setting where the network is observed repeatedly over time. A Bayesian nonparametric model is introduced for this setting, allowing the latent variable assignments to vary dynamically and the complexity of the latent representation to be inferred from the data. We also explore the use of text to infer the semantics of the latent features in these models for both email data and for a network of twitter users. Continuing our theme of network and text modeling, the second contribution of the dissertation is a model which leverages text information to recover influence relationships between scientific articles along the citation graph.

The success of latent variable models at finding meaningful latent structure in data motivates their application in increasingly large data sets, where human attention is increasingly diluted across the data. This brings with it challenges of scale. For example, it is not feasible to fit a topic model to the online encyclopedia Wikipedia using traditional single-threaded batch learning algorithms. This thesis introduces a stochastic algorithm for efficiently and

accurately learning topic models, by exploiting the collapsed representation of the model, in which the model parameters are marginalized out and inference is performed on the latent variables alone.

Evaluating topic models is a task even more computationally challenging than learning these models, as computing the likelihood of a single held-out document involves an intractable integral or an intractable sum, and there may be thousands or tens of thousands of held-out documents. The final contribution of this thesis is to introduce new reliable methods for comparing the predictive performance of topic models and for efficiently estimating their predictive quality per iteration of the algorithms used to train them. This facilitates thorough empirical comparisons of the convergence properties of different learning algorithms.

The remainder of this chapter provides an introduction to latent variable modeling and summarizes the contributions of the dissertation.

1.1 Latent Variable Models

Suppose we are given a collection $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N)$ of D-dimensional data observations. In a probabilistic modeling context, we assume that $\mathbf{x}$ is random and we are interested in learning its distribution $Pr(\mathbf{x})$. A latent variable model approaches this by augmenting the observed variables $\mathbf{x}$ with unobserved (a.k.a. "*latent*" or "*hidden*") variables $\mathbf{z}$, allowing us to reason over the joint distribution $Pr(\mathbf{x}, \mathbf{z})$. Note that we can recover $Pr(\mathbf{x})$ from this joint distribution through marginalizing the latent variables, since $Pr(\mathbf{x}) = \sum_{\mathbf{z}} Pr(\mathbf{x}, \mathbf{z})$ due to the law of total probability.[1] The latent variables may correspond to real-world quantities

[1] If the latent variables are continuous, this sum is replaced with an integral. Notationally, we will assume discrete variables whenever possible for simplicity.

which are hidden from us, or to convenient artificial constructs introduced for modeling purposes.[2]

Due to the product rule, the joint distribution $Pr(\mathbf{x}, \mathbf{z})$ can be factorized as $Pr(\mathbf{x}|\mathbf{z})Pr(\mathbf{z})$. We can think of $Pr(\mathbf{z})$ as the prior probability of the latent variables. It is often convenient to assume that the $\mathbf{x}_i$ are conditionally independent given $\mathbf{z}$, i.e. $Pr(\mathbf{x}, \mathbf{z}) = \prod_i Pr(\mathbf{x}_i|\mathbf{z})Pr(\mathbf{z})$. This conditional independence assumption still allows for complex dependencies between the $\mathbf{x}_i$ in the marginal distribution $Pr(\mathbf{x})$ by way of the latent structure. In this form, we can construct our latent variable models via an assumed generative process, where the latent variables are drawn first, and then the observed variables are each drawn independently based on the latent variables.

We motivated latent variable models above through their ability to perform dimensionality reduction. This is typically achieved by associating each D-dimensional observation $\mathbf{x}_i$ with a K-dimensional latent representation $\mathbf{z}_i$, where $K \ll D$, and further assuming that $Pr(\mathbf{x}, \mathbf{z}) = \prod_i Pr(\mathbf{x}_i|\mathbf{z}_i)Pr(\mathbf{z})$ (Tipping & Bishop, 1999). With an appropriately specified model, each $\mathbf{z}_i$ captures important properties of its $\mathbf{x}_i$, while living in a lower dimensional space. In the nomenclature of neural networks, we can think of $\mathbf{z}$ as a *bottleneck* through which the data are squeezed, thus compressing them.[3]

[2] In the words of Box & Draper (1987), "all models are wrong. Some models are useful." Even very unrealistic models may be useful for finding interesting latent structure in the data, and may still be useful for density estimation and prediction.

[3] Alternatively, it is still possible to achieve compression by setting $K > D$ as long as the latent representations are made to be sparse, e.g. Teh *et al.* (2003). We will not consider these *sparse overcomplete* representations further here.

1.1.1 A Simple Example: Mixture Models

A simple example of a latent variable model is the mixture model,

$$Pr(\mathbf{x}_i) = \sum_k Pr(z_i = k) Pr(\mathbf{x}_i | z_i = k) \ . \tag{1.1}$$

Here, the latent variable z_i is a discrete class for each data point $\mathbf{x}_i$. The component distributions $Pr(\mathbf{x}_i|z_i = k)$ specify the data probability for each class, e.g. $Pr(\mathbf{x}_i|z_i = k) = \text{Gaussian}(\mathbf{x}_i; \mu_k, \Sigma_k)$. A useful property of many latent variable models is that they often correspond to an intuitive generative "story" of how the data were assumed to be generated. In the case of the mixture model, Equation 1.1 is equivalent to the following generative process:

- For each data point i
  - Draw a discrete latent variable assignment $z_i \sim \text{discrete}(\pi)$
  - Draw the data point from component z_i, $\mathbf{x} \sim Pr(\mathbf{x}_i|z_i)$,

where $\pi_k = Pr(z_i = k)$. Drawing z_i from discrete(π) can be thought of as rolling a biased K-sided die to select the latent class assignment. Even when the component distributions are unimodal, the latent variables in the mixture model result in a flexible multi-modal distribution by adding probability mass around the mode of each component. Given the component distributions and the probability of each component, we can perform inference to estimate the assignments via Bayes rule.

In this way the latent variable framework provides us with a (probabilistic) clustering, i.e. a partition of the data into related sets of data points, thus extracting hidden structure present in the data in a model-based way. In terms of dimensionality reduction, we have mapped our possibly high-dimensional data points x_i onto a discrete cluster label z_i. We

can also interpret this as mapping $\mathbf{x}_i$ onto a K-dimensional binary vector with zeros in every co-ordinate except the z_ith entry, which contains a one, thus projecting the D-dimensional $\mathbf{x}$'s onto a K-dimensional space.

1.2 Latent Variable Models as Matrix Factorization

Principal component analysis finds a latent representation of the data matrix by factorizing it, via the singular value decomposition, in order to reduce its dimensionality. It should perhaps be unsurprising, then, that many latent variable models can also be interpreted as performing matrix factorization. Buntine & Jakulin (2006) and Griffiths *et al.* (2007) illustrate this interpretation for an important class of latent variable models known as topic models, and many social network models are also explicitly described this way, e.g. Miller *et al.* (2009).

We will use the discrete component analysis (DCA) framework of Buntine & Jakulin (2006) to encode many of the major latent variable models in the literature, including all models encountered in this thesis, as matrix factorization models. To achieve this, here we generalize Buntine and Jakulin's framework slightly by relaxing their assumption that the data are discrete, and by including a variant of it for one-mode network models. This will allow us to give a very concise review of the literature, as well as showing the relationships between the different models. In the following, we will also slightly modify DCA in order to make use of the terminology and notation of *generalized linear models* (GLMs) (Mccullagh & Nelder, 1989), which are closely related to the present framework.

1.2.1 A Matrix Factorization Framework

Let us represent our data set in an $N \times D$ matrix $\mathbf{x}$, where N is the number of data points and D is their dimensionality. The entries of $\mathbf{x}$ are assumed to be real-valued, or belonging to a subset of the real numbers such as the integers or binary numbers. We model the expected value μ of the data matrix as a function of a matrix product of latent variable matrices,

$$\mu \triangleq E[\mathbf{x}|\mathbf{R}, \mathbf{C}] = g^{-1}(\eta) \tag{1.2}$$

$$\eta = \mathbf{R}\mathbf{C}^\intercal \ , \tag{1.3}$$

where η is an auxiliary $N \times D$ matrix, and g, which is assumed monotonic, differentiable and invertible, is known in the GLM literature as a *link function*. Its inverse g^{-1}, which maps the linear predictor η to the mean μ, is called a *mean function* or an *inverse link function*. The matrix $\mathbf{R}$ is $N \times K$ and consists of low-dimensional latent variable representations of each data point (i.e. each row of $\mathbf{x}$), and $\mathbf{C}$ is a $D \times K$ matrix consisting of low-dimensional representations of each feature (i.e. each column of $\mathbf{x}$). In a manner reminiscent of GLMs, the final distribution is specified by

$$Pr(\mathbf{x}|\mathbf{R}, \mathbf{C}, \theta^{(f)}) = f_{\theta^{(f)}}(\eta) \ , \tag{1.4}$$

where f is a distribution, typically from the exponential family, with $\theta^{(f)}$ optionally specifying additional parameters such as variances. For example, if $\mathbf{x}$ contains count data, f may be a Poisson distribution on each of the entries, $x_{ij} \sim \text{Poisson}(\eta_{ij})$. After including a prior distribution $Pr(\mathbf{R}, \mathbf{C}, \theta^{(f)})$, we have now specified a family of latent variable models $Pr(\mathbf{x}, \mathbf{z})$, where the latent variables $\mathbf{z} = \{\mathbf{R}, \mathbf{C}, \theta^{(f)}\}$ correspond to K-dimensional compressed representations of the rows and columns of $\mathbf{x}$, as well as any additional parameters for the likelihood. Alternatively, instead of interpreting the rows of $\mathbf{C}$ as latent representations for the features in $\mathbf{x}$, we can interpret the *columns* of $\mathbf{C}$ as D-dimensional representations of each

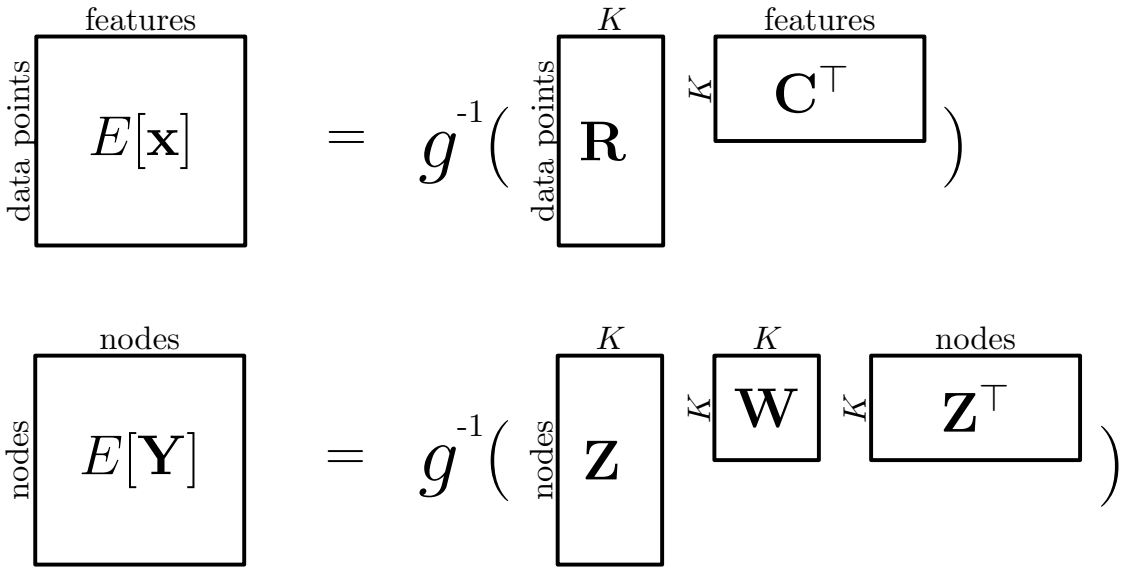


Figure 1.1: A matrix factorization framework for latent variable models.
Top: Rectangular data matrices and two-mode networks. *Bottom*: Single-mode networks.

of the K latent dimensions in $\mathbf{R}$. See Figure 1.1 (*top*) for an illustration of this modeling framework.

Consider, for example, the Gaussian mixture model. If data point $\mathbf{x}_i$ belongs to class k, we can set $\mathbf{R}_i$ (the ith row of matrix $\mathbf{R}$) to be a binary vector consisting of a single one at the kth entry, and zeros elsewhere. Let each column c of $\mathbf{C}$ be the mean $\mathbf{m}_c$ of the kth component Gaussian. We then may compute $\eta = \mathbf{R}\mathbf{C}^\intercal$. The matrix product "selects" the component mean $\mathbf{m}_k$, and re-represents $\mathbf{x}_i$ with $\eta_i = \mathbf{m}_k^\intercal$. So we can write $Pr(\mathbf{x}|\mathbf{z})$ for the Gaussian mixture model as $Pr(\mathbf{x}|\mathbf{R}, \mathbf{C}, \Sigma) = f_\Sigma(\eta)$, where f consists of a multivariate Gaussian distribution for each row i with mean η_i and covariance $\mathbf{\Sigma}_k$, with k being the class assignment for data point $\mathbf{x}_i$.

This framework is very general, and it includes the vast majority of latent variable models in the literature (after generalizing it to network models, which we do below). Essentially the only constraining assumption is that the expectation of $\mathbf{x}$ under the model can be written as $g^{-1}(\mathbf{R}\mathbf{C}^\intercal)$ for some latent variables $\mathbf{R}$ and $\mathbf{C}$ and mean function g^{-1}. This is exceedingly common in practice. Although $\eta = \mathbf{R}\mathbf{C}^\intercal$ is linear, the mean function g^{-1} may introduce

non-linearities. Furthermore, both $\mathbf{R}$ and $\mathbf{C}$ are unobserved *free parameters*. This is in contrast to GLMs (Mccullagh & Nelder, 1989), which follow a similar pattern except that η depends on a *fixed*, *observed* feature vector $\mathbf{x}_i$, and the goal is to perform regression, predicting a response variable y_i for each $\mathbf{x}_i$. GLMs have a form somewhat parallel to the above framework,

$$\eta_i = \mathbf{x}_i^\intercal \beta \tag{1.5}$$

$$\mu_i \triangleq E[y_i|\mathbf{x}_i, \beta, \theta^{(f)}] = g^{-1}(\eta_i) \tag{1.6}$$

$$Pr(y_i|\mathbf{x}_i, \beta, \theta^{(f)}) = f_{\theta^{(f)}}(\mu_i) \ , \tag{1.7}$$

where it is assumed that f is in the exponential family. The linearity assumption in our latent variable framework is in some sense less restrictive than for GLMs, as η is a linear function of adjustable free parameters, instead of being a linear function of a fixed observation vector. We can think of the framework as an unsupervised version of generalized linear models (Buntine & Jakulin, 2006).

1.2.2 Terminology: Latent Variables vs Parameters

Before extending the framework to network models, we will pause to discuss the terminology used in latent variable modeling, in the context of the framework. Specifically, a frequentist statistician makes a clear distinction between *variables*, which are random, and *parameters*, which are assumed fixed but unobserved. Here, we assume a Bayesian stance, where all elements of the models are considered to be random. From a Bayesian perspective, there is no mathematical difference between parameters and latent variables. In our case, for example, all elements of $\mathbf{z} = \{\mathbf{R}, \mathbf{C}, \theta^{(f)}\}$, are latent, and are variable, and so therefore they are latent variables. In this thesis, we will also use the word parameter to describe some of

the latent variables when this is convenient. Following Murphy (2012), we will refer to the variables which are not associated with specific data points, $\mathbf{C}$ and $\theta^{(f)}$, as parameters.

In certain cases such as for topic models, the convention is to also refer to continuous variables within $\mathbf{R}$ as parameters, and to use the term "latent variable" only to refer to discrete hidden variables. We follow this convention when appropriate.

1.2.3 Single-Mode Network Data

In the above we have assumed that our data matrix $\mathbf{x}$ is a rectangular $N \times D$ matrix. This does not hold for non-bipartite network data, where the observation $\mathbf{Y}$ is an $N \times N$ (binary, or weighted) adjacency matrix of a graph, representing for example connectivity in a social network.[4] For such data the row and column entities are the same, so we do not need separate row and column latent variables $\mathbf{R}$ and $\mathbf{C}$. Instead, we can represent the entities with one $N \times K$ matrix of latent representations, $\mathbf{Z}$. When building network models with these latent variables $\mathbf{Z}$, a matrix factorization modeling framework is once again a convenient formalizm (Hoff, 2007). The distribution over the adjacency matrix is once again parameterized in GLM-style as

$$\mu \triangleq E[\mathbf{Y}] = g^{-1}(\eta) \tag{1.8}$$

$$\eta = \mathbf{Z}\mathbf{W}\mathbf{Z}^{\intercal} \ . \tag{1.9}$$

Here, $\mathbf{W}$ is a $K \times K$ matrix which encodes the relationships between the latent variables used to represent the entities. See Figure 1.1 (*bottom*) for an illustrative diagram of the modeling framework. In practice, it is useful to extend the framework slightly to allow additional

[4] Bipartite networks, also known as two-mode networks, where there are two types of entities and there are no connections within each type, can be written as rectangular matrices. Each row of the matrix is associated with entities of type one, and each column is associated with entities of type two. The rectangular matrix factorization framework for latent variable models applies directly, see e.g. DuBois *et al.* (2011).

linear terms specifying actor-specific tendencies to form and receive ties, as well as intercept terms (Krivitsky *et al.* , 2009),

$$Pr(y_{ij}|\mathbf{Z}, \mathbf{W}, \rho, \xi, \epsilon, \theta^{(f)}) = f_{\theta^{(f)}}(g^{-1}(\mathbf{z}_i \mathbf{W} \mathbf{z}_j^\intercal + \rho_i + \xi_j + \epsilon)) \ . \tag{1.10}$$

These linear terms could in principle be included for a rectangular $\mathbf{x}$ as well. Since networks are frequently represented as binary adjacency matrices, the likelihood portion of the model is typically specified by conditionally independent Bernoulli distributions for each y_{ij}. The framework allows us to concisely describe many of the models in the literature. A survey of latent variable models for rectangular data matrices is given in Figure 1.2, and for single-mode networks in 1.3, leveraging the framework to show the different choices along which the models in the literature vary. We detail some of the possible modeling options in the next section.

1.3 Modeling Choices

We can specify many different latent variable models by using different choices for the row and column latent representations, the link function and the prior on the latent variables.

1.3.1 Latent Representation

By constraining the latent representations, different semantics can be imposed. In the following we focus our discussion on the row representations $\mathbf{R}_i$ of the data points. Some options are:

Figure 1.2: Matrix factorization representations of latent variable models

Model	**Rows**	**Columns**	**Likelihood**	**Infinite**
Factor analysis cf. Spearman (1904)	continuous	continuous	row-wise diagonal multivariate Gaussian	no
Infinite sparse factor analysis Knowles & Ghahramani (2007)	continuous	continuous, sparse	row-wise isotropic multivariate Gaussian	yes
Probabilistic PCA Tipping & Bishop (1999)	continuous	continuous	row-wise isotropic multivariate Gaussian	no
Gaussian mixture model, cf. Orchard & Woodbury (1972)	latent class	continuous	row-wise multivariate Gaussian	no
Infinite Gaussian mixture model, Rasmussen (1999)	latent class	continuous	row-wise multivariate Gaussian	yes
Collaborative filtering cf. Bell & Koren (2007)	continuous	continuous	element-wise Gaussian	no
Binary matrix factorization Meeds *et al.* (2007)	binary latent feature	binary latent feature	element-wise Gaussian	yes
Latent set model DuBois *et al.* (2011)	binary latent feature	sparse feature probabilities	noisy-or	no
Latent Dirichlet allocation Blei *et al.* (2003)	mixed membership	mixed membership	row-wise multinomial	no
HDP topic model Teh *et al.* (2006)	mixed membership	mixed membership	row-wise multinomial	yes
Gamma Poisson topic model Canny (2004)	continuous	continuous	Poisson	no
Focused topic model Williamson *et al.* (2010)	sparse mixed membership	mixed membership	row-wise multinomial	yes
SparseTM topic model Wang & Blei (2009)	mixed membership	sparse mixed membership	row-wise multinomial	yes
Restricted Boltzmann machines, Hinton (2002) Smolensky (1986)	latent feature	continuous	Bernoulli logit	No

Figure 1.3: Matrix factorization representations of network models

Model	**Entity**	**Feature-feature interaction**	**Likelihood / Link**	**Infinite**
Latent eigenmodel Hoff (2007)	continuous	continuous, diagonal	Bernoulli probit	no
Latent space model[a] Hoff *et al.* (2002)	continuous	identity	Bernoulli logit	no
Stochastic block model Nowicki & Snijders (2001)	latent class	probability	Bernoulli	no
Infinite relational model Kemp *et al.* (2006)	latent class	probability	Bernoulli	yes
Mixed membership stochastic block model Airoldi *et al.* (2008)	mixed membership	probability	Bernoulli	no
Latent feature relational model Miller *et al.* (2009)	latent feature	continuous	Bernoulli logit	yes

[a]The latent space model computes probabilities of ties based on the distance between the actors in the latent space. The latent eigenmodel weakly generalizes the latent space model, in the sense that it can closely approximate it (Hoff, 2007). It can also be mapped to the matrix factorization framework using the identity $\|q - p\| = \sqrt{-2q \cdot p + \|q\|^2 + \|p\|^2}$.

Latent Class

In a latent class model such as a mixture model, we set $\mathbf{R}_i$ to be a binary vector which sums to one, i.e. there is exactly one entry with the value one. This encodes the property that each data point belongs to a single class. E.g., $\mathbf{R}_i = (0, 0, 1)$ assigns x_i to class three.

Mixed Membership

If we relax the constraint that R_i is binary while continuing to require that $\sum_k R_{ik} = 1$, the latent representation allows each data point x_i to have partial or "*mixed*" membership of the K classes (Hofmann, 1999a,b; Pritchard *et al.* , 2000; Blei *et al.* , 2003; Erosheva *et al.* , 2004). E.g., $\mathbf{R}_i = (0.1, 0.8, 0.1)$ assigns 80% of its membership weight to class two.

We can interpret R_{ik} as the probability that an additional latent class assignment $z_i = k$, i.e. x_i belongs to class k. Since $\eta_{ij} = \mathbf{R}_i \mathbf{C}_j^\intercal = \sum_k R_{ik} C_{jk} = \sum_k Pr(z_i = k) C_{jk}$, we can understand this as averaging over or marginalizing out the class assignment. This formulation is useful for models where each data point has multiple opportunities to select a class, according to its mixed membership vector. This should be contrasted with standard mixture models, where each entity is assigned just one class. Mixed membership models arise naturally in the study of population genetics, where the genes of an individual are determined by multiple ancestral populations (Pritchard *et al.* , 2000). In the population genetics literature, these models are called *admixture* models.

In machine learning, the social network model known as the mixed membership stochastic blockmodel (Airoldi *et al.* , 2008) posits that each actor in the network selects a different class when determining the presence or absence of an edge connecting them to each other actor. In the latent Dirichlet allocation topic model for text corpora (Blei *et al.* , 2003), each word in a document is assigned its own latent class according to that document's mixed membership distribution, instead of assigning the document to a single latent class. Note that here the latent variables have a *hierarchical* structure, in which the latent class variables z_i are generated based on other latent variables, namely the mixed membership vectors. Hierarchical model-building is a common pattern in latent variable modeling, as it allows for more sophisticated latent structure.

Binary Latent Features

The mixed membership latent representation requires that the membership values sum to one across the classes. This implies a "conservation of mass" relationship between the latent classes, where a limited amount of probability mass is spread across all of the classes. Thus, in a two-class model where class one corresponds to "[the person associated with data point] $\mathbf{x}_i$ likes basketball" and class two corresponds to "$\mathbf{x}_i$ likes tennis," then if we increase the

extent to which $\mathbf{x}_i$ likes basketball then we must decrease the extent to which $\mathbf{x}_i$ likes tennis by the same amount. This is undesirable in some applications – e.g., we should not expect an interest in basketball to require a lack of interest in tennis.

If we wish to avoid this conservation of mass constraint but still allow multiple classes to be chosen for a given data point (e.g. $\mathbf{x}_i$ likes both basketball and tennis), an alternative choice is to use binary vectors for the latent representation, without any constraint on the sum (Griffiths & Ghahramani, 2006). In this case, $\mathbf{R}_i = (1, 1)$ corresponds to $\mathbf{x}_i$ having an interest in both basketball and tennis.

Feature Probabilities

An alternative to the binary latent feature representation is to relax the constraint that features are either "on" with value one, or "off" with value zero, and instead assign them a "probability of being on," i.e. a value between zero and one. This representation is useful when we want feature semantics along the lines of "in each interaction, make use of feature k with probability π_k" (DuBois *et al.* , 2011).

Continuous Latent Space

The above kinds of models constrain the representation, which helps to control the semantics of the latent variables in potentially useful ways such as forcing the model to output a clustering of the data. We need not restrict ourselves to binary or mixed membership vectors, however. Unconstrained continuous latent representations can encode similarities between entities within a latent embedding of social actors (Hoff *et al.* , 2002), or encode levels of (dis)interest or affinity with latent categories in recommender systems for products such as movies (Bell & Koren, 2007). If sparsity in the latent representation is desired, this can be

achieved by taking the Hadamard (elementwise) product of a continuous vector and a binary vector (Griffiths & Ghahramani, 2006).

1.3.2 Link Function and Likelihood

The likelihood f and the inverse link function g^{-1} relate the latent variables to the observed variables. Some possible options are

- **Gaussian**
  - Each entry of the matrix has a univariate Gaussian

    $x_{ij} \sim \text{Gaussian}(\eta_{ij}, \sigma_{z_i})$
  - Each row of the matrix has a multivariate Gaussian

    $\mathbf{x}_i \sim \text{Gaussian}(\eta_i, \Sigma_{z_i})$
  - A multivariate Gaussian across the entire matrix

    $\mathbf{x}(:) \sim \text{Gaussian}(\eta, \Sigma)$

- **Poisson**
  - Constraining the parameters to be positive

    $x_{ij} \sim \text{Poisson}(\eta_{ij})$
  - Transforming the parameters to be positive

    $x_{ij} \sim \text{Poisson}(\exp(\eta_{ij}))$

- **Bernoulli**
  - Constraining $0 \leq \eta_{ij} \leq 1$

    $x_{ij} \sim \text{Bernoulli}(\eta_{ij})$
  - Logit link function

    $x_{ij} \sim \text{Bernoulli}(\sigma(\eta_{ij}))$
  - Probit link function

    $x_{ij} \sim \text{Bernoulli}(\Phi(\eta_{ij}))$
- **multinomial**
  - $\mathbf{x}_i \sim \text{multinomial}(\eta_i, N_i)$ (each row has a multinomial distribution)
  - $\mathbf{x}(:) \sim \text{multinomial}(\eta, N)$ (a multinomial across the entire matrix)

1.3.3 Priors

It is possible to select any prior on the latent variables whose support is the space of the chosen latent representation. However, priors which are conjugate, i.e. the posterior is in the same family as the prior, are often selected for computational reasons. Conjugate priors are more likely to result in tractable update equations for learning and inference algorithms, and in some cases allow variables to be marginalized out. Note that conjugacy may be with respect to the inverse link function, or in a hierarchical model where there are multiple layers of latent variables, to the next layer. If we have Bernoulli distributed variables we can place beta priors on their parameters (Griffiths & Ghahramani, 2006); with Poisson variables we can use gamma priors (Canny, 2004; Titsias, 2008); with multinomial variables we may use Dirichlet priors (Blei *et al.* , 2003), and so on.

Bayesian Nonparametric Priors

The ideal dimensionality K of the latent variables is often unknown in practice. One strategy for resolving this is to use a *Bayesian nonparametric* model, where the prior allows for an unbounded dimensionality on the latent variables. For example, consider the mixture model. We must specify a prior distribution $Pr(\mathbf{z})$ on the latent class assignments. Normally this consists of a discrete distribution, which can be represented as a K-dimensional vector which sums to one. In a Bayesian framework, we can place a further Dirichlet prior over this discrete distribution.

Alternatively, the *Dirichlet process* (DP) (Ferguson, 1973) is a prior with support over *all* discrete distributions of any dimensionality. Surprisingly, this prior distribution also results in tractable posterior distributions. This makes it feasible to define mixture models with an unbounded number of classes (Neal, 1992, 2000; Rasmussen, 1999).

In a mixture modeling context, drawing from the discrete distribution arising from Dirichlet process can be understood using an equivalent process known as the Chinese restaurant process (CRP) (Aldous, 1985). In the CRP, there are an infinite number of "tables" (i.e. clusters), and each "customer" of the restaurant (i.e. data point) sits at a table based on the popularity of the dish at that table, i.e. the number of "customers" already at the table. Continuing with the culinary metaphors, for binary latent feature models and for sparse continuous representations, we can instead use the Indian buffet process (IBP), which is a prior on binary matrices with an unbounded number of columns (Griffiths & Ghahramani, 2005, 2006). There are deep connections between the Dirichlet process and the Indian buffet process. The DP can be understood using a "stick-breaking" construction, where a "stick" of length one is broken repeatedly, and each cluster is assigned probability mass according to the discarded portion of the stick (Sethuraman, 1994). The IBP can also be derived using

a similar stick-breaking process, where the other half of the stick from the CRP gives the probability of each feature being active (Teh *et al.* , 2007b).

Frequentist Approaches

Another strategy is to use a frequentist approach to modeling. In this case, some or all of the latent values are interpreted as fixed but unknown *parameters*, instead of random variables. Then, no prior distribution is specified on the parameters, and learning proceeds using a frequentist method such as maximum likelihood learning (e.g. Hofmann (1999b)). An intermediate approach between frequentist learning and fully Bayesian models is to add penalty terms to the modeling objective function such as the log-likelihood. These terms can be used to encourage certain properties such as sparsity. In some cases, they also correspond to a Bayesian prior. For example, the lasso (Tibshirani, 1996), may be used, which leads to sparse representations. The lasso is equivalent to maximizing the posterior probability of the parameters, with a Laplace prior centered at zero.

1.3.4 Sequential Data

Often we do not observe just a single data matrix $\mathbf{x}$, but multiple snapshots of it over time, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(T)}$. A straightforward way to model such sequential data in a latent variable framework is to allow the latent variables to change over time, $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(T)}$, and model the data at each timestep as being drawn in the usual way based on the latent variables at that timestep. The simplest approach is to use a hidden Markov model, by assuming that the latent variables are generated according to a Markov chain, i.e. $Pr(\mathbf{z}^{(i)}|\mathbf{z}^{(i-1)}, \mathbf{z}^{(i-2)}, \ldots, \mathbf{z}^{(1)}) = Pr(\mathbf{z}^{(i)}|\mathbf{z}^{(i-1)})$. If each latent variable $z_k^{(.)}$ is assumed to have its own independent Markov chain, i.e. the latent variables and their dynamics are independent from each other in the prior, this is called a factorial hidden Markov model

(Ghahramani & Jordan, 1997). Nonparametric Bayesian latent variable models for sequential data are also possible, including the infinite HMM (Beal *et al.* , 2002) and the infinite factorial HMM (Van Gael *et al.* , 2009). It is also possible to use higher-order Markov dependencies (at greater computational expense), or alternatively to use models with continuous time instead of discrete timesteps (Fan & Shelton, 2009).

1.4 Learning and Inference

We have seen how we can design latent variable models which posit interesting latent structure in the data. However, all of this modeling is for naught unless we are able to *recover* this latent structure. To achieve this, we need to develop algorithms to infer latent variables and parameters from data.[5]

In a Bayesian context, all of our latent variables and parameters $\mathbf{z}$ are random variables, and we can write down their posterior probability using Bayes rule:

$$Pr(\mathbf{z}|\mathbf{x}) = \frac{Pr(\mathbf{x}|\mathbf{z})Pr(\mathbf{z})}{Pr(\mathbf{x})} \ . \tag{1.11}$$

Ideally we would like to infer the full posterior $Pr(\mathbf{z}|\mathbf{x})$. One strategy for this is to perform Markov chain Monte Carlo (MCMC) techniques, which simulate from the distribution by sampling from a Markov chain invariant to it (Metropolis *et al.* , 2004; Hastings, 1970; Geman & Geman, 1984; Gelfand & Smith, 1990).[6] Given enough computation time, these methods will obtain draws from the posterior as long as the Markov chain is ergodic (i.e. it

[5] In the graphical model literature, some authors make a distinction between *inference*, which recovers latent variables, and *learning*, which recovers parameters. In this thesis we are mainly operating in a Bayesian framework where there is not always a clear distinction between latent variables and parameters. We will therefore switch between learning and inference terminology depending on the context, but in both cases we refer to the recovery of unknown values of interest in our models.

[6] A Markov chain is *invariant* to a distribution if, starting from a draw from that distribution, subsequent iterations of the Markov chain lead to samples which are still draws from that distribution.

will eventually be able to reach any point in the space, without periodic behavior, and with the ability to return to any previous state). The simplest MCMC algorithm, which we will make repeated use of in this thesis, is the *Gibbs sampler*, which samples from the posterior by repeatedly iterating the update

$$\mathbf{z}_i \sim Pr(\mathbf{z}_i|\mathbf{z}_{\neg i}, \mathbf{x}) \tag{1.12}$$

for each $\mathbf{z}_i$, where $\mathbf{z}_{\neg i}$ consists of the variables in $\mathbf{z}$ excluding $\mathbf{z}_i$. This strategy can be very effective in some cases, and MCMC is one of very few algorithms which is guaranteed to eventually correctly estimate the true posterior in the long run. However, in some cases MCMC will mix poorly and take a very long time to correctly draw from the posterior. Such failures can also be very hard to diagnose. Furthermore, given a computational budget it may be beneficial to use optimization techniques instead of simulation. Optimization algorithms tend to travel more directly "uphill" in their objective functions, exhibiting less "random walk" behaviour than their simulation counterparts. This means that they may reach a reasonable solution more quickly than MCMC.

When using optimization algorithms, an objective function must be chosen. It is generally not feasible to optimize the full posterior $Pr(\mathbf{z}|\mathbf{x})$ directly. A more practical optimization strategy known as variational Bayesian (VB) inference instead reasons with a more tractable distribution $q(\mathbf{z})$ such as a fully factorized distribution. The VB method proceeds by minimizing the KL-divergence between $q(\mathbf{z})$ and $Pr(\mathbf{z}|\mathbf{x})$ (Jaakkola & Jordan, 1997; Jordan *et al.* , 1999). A disadvantage of this approach is that if $Pr(\mathbf{z}|\mathbf{x})$ is not a member of the family q (which is typically the case), even the optimal variational distribution $\hat{q}(\mathbf{z})$ is only an approximation, and thus the variational Bayes approach introduces a bias in the estimation.

An alternative optimization strategy is to find a single point estimate with maximum posterior probability. This strategy is known as maximum a posteriori (MAP) estimation. If a uniform prior is used then this becomes equivalent to maximizing the likelihood function, a strategy known as maximum likelihood estimation. Normally the estimation is only performed over parameters, marginalizing out other hidden variables. This approximates the entire posterior distribution as a single delta function, which is a less rich representation than that of variational Bayes. Unlike VB, however, MAP and maximum likelihood estimators are asymptotically consistent. Let $\mathbf{z}^{(p)}$ be the parameters (i.e. variables not associated with individual data points) and $\mathbf{z}^{(l)}$ be other latent variables. Then the optimization problem is to solve

$$\arg\max_{\mathbf{z}^{(p)}} Pr(\mathbf{z}^{(p)}|\mathbf{x}) = \arg\max_{\mathbf{z}^{(p)}} \frac{Pr(\mathbf{x}|\mathbf{z}^{(p)})Pr(\mathbf{z}^{(p)})}{Pr(\mathbf{x})} \tag{1.13}$$

$$= \arg\max_{\mathbf{z}^{(p)}} \log Pr(\mathbf{x}|\mathbf{z}^{(p)}) + \log Pr(\mathbf{z}^{(p)}) \tag{1.14}$$

$$= \arg\max_{\mathbf{z}^{(p)}} \log \sum_{\mathbf{z}^{(l)}} Pr(\mathbf{x}, \mathbf{z}^{(l)}|\mathbf{z}^{(p)}) + \log Pr(\mathbf{z}^{(p)}) \ . \tag{1.15}$$

If $\mathbf{z}^{(l)}$ is empty then it is typically straightforward to take the derivative of $Pr(\mathbf{x}|\mathbf{z}^{(p)})$ and gradient ascent algorithms may be used. If not, an alternative is to perform the expectation-maximization (EM) algorithm (Dempster *et al.* , 1977). EM operates by performing an "E-step" which involves computing the expectation of the complete data log-likelihood (the likelihood with the latent variables $\mathbf{z}^{(l)}$ treated as observed) given the current parameters $\mathbf{z}^{(p)}$, which is a lower bound on the log-likelihood. This lower bound is then optimized (the "M-step"). The procedure is repeated until convergence. The algorithm monotonically improves the objective function, and is guaranteed to find a local maximum. It can also be interpreted as performing co-oordinate descent in an augmented state space, and each iteration can be viewed as solving a small variational inference problem (Neal & Hinton, 1998).

If an optimization strategy is used, an alternative is to use stochastic optimization algorithms. These algorithms look at a subset of the data at a time and estimate what the optimization update would be if the full dataset was processed, then perform that update. This allows updates to occur much sooner, which can bring significant computational advantages, particularly if the data set is very large. A disadvantage is that the convergence rate, in terms of the amount of data examined, is much lower for stochastic algorithms than full batch algorithms. This may result in worse solutions for a reasonable amount of computation, in the cases where the batch algorithms can be applied for many iterations. Stochastic versions of gradient descent (cf. Bottou (1998)), expectation maximization (Cappé & Moulines, 2009) and variational inference (Hoffman *et al.* , 2013) are available.

1.5 Latent Dirichlet Allocation Topic Models

Now that we have seen a high-level overview of latent variable modeling, let us consider an important example: the latent Dirichlet allocation (LDA) topic model (Blei *et al.* , 2003). Given a corpus of text documents, the LDA model can be used to find semantic themes ("*topics*") which are meaningful to humans, in a completely unsupervised way. We will make use of LDA in every chapter of this thesis.

1.5.1 A Simple Naive Bayes Text Model

When modeling text data, we are typically given a corpus of documents such as news articles, weblogs or scientific papers. We can encode each document d with a sequence of words $w^{(d)} = w_1^{(d)}, w_2^{(d)}, \ldots, w_{N^{(d)}}^{(d)}$. A common pre-processing step is to represent each document as a sparse D-dimensional *bag of words* vector $\mathbf{x}^{(d)}$, where $\mathbf{x}_j^{(d)}$ is the number of occurrences of the word j in document d. The bag of words representation sacrifices the information available

in the ordering of the words, but gains a simple, fixed dimensional vector representation for all documents in the corpus, allowing us to apply standard machine learning techniques.

A simple latent variable approach to modeling text data is to use a mixture model, which assigns each document to a latent cluster, as in Equation 1.1. A Gaussian mixture model could in principle be applied to the bag of words vectors $\mathbf{x}^{(d)}$, but this approach does not work well in practice because the Gaussian assumption is not accurate for sparse, discrete count data (Buntine & Jakulin, 2006). Perhaps the most straightforward alternative is to make the *naive Bayes* assumption, namely that each word is conditionally independent given the topic of the document, and use *discrete* distributions for drawing each of the words. This naive Bayes text model (cf. Carpenter (2010)) corresponds to the assumed generative process in Algorithm 1.

Algorithm 1 Generative process for the naive Bayes text model

- For each document d
  - $z^{(d)} \sim \text{discrete}(\pi)$ *//Sample a topic*
  - For each word i in document d
    - $w_i^{(d)} \sim \text{discrete}(\Phi^{(z^{(d)})})$ *//Sample a word*

This model makes use of the discrete distribution, which is represented by a vector which sums to one, and can be viewed as rolling a weighted die and recording the outcome. For example, discrete(π) rolls a K-sided die weighted by π which chooses the cluster assignment $z^{(d)}$ for each document d. In a text modeling context, we call such a cluster assignment a *topic*, as it is intended that this coincides with the semantic theme of the document. Similarly, a topic k is represented by a D-dimensional discrete distribution over words $\Phi^{(k)}$, which can be understood as a D-sided weighted die.

1.5.2 Latent Dirichlet Allocation

The simple naive Bayes model we considered above uses a latent class representation for each document, which assumes that every document is associated with exactly one topic. Instead, the latent Dirichlet allocation (LDA) model of Blei *et al.* (2003) relaxes this assumption, and represents each document d with a K-dimensional *mixed membership* vector $\theta^{(d)}$ which sums to one. This means that a document can be about multiple topics, to varying degrees. For example, this thesis is about latent variable models, networks, and text data. We could represent it by

$$\theta^{(\text{thesis})} = [0.4, 0.25, 0.35]^\intercal \ , \tag{1.16}$$

where the entries correspond to topics on latent variables, networks, and text, respectively. Since each $\theta^{(d)}$ sums to one, we can once again think of it as a weighted K-sided die. Rolling this die chooses a topic. In the LDA model, we roll the die $\theta^{(d)}$ for every word i in every document d, thereby selecting a topic assignment $z_i^{(d)}$ for each word. We then draw each word $w_i^{(d)}$ from the chosen topic, as in the naive Bayes mixture model. This corresponds to the words in each document being drawn from a mixture model with its own unique prior over components (topics), $\theta^{(d)}$. After adding Dirichlet priors for the topics $\Phi^{(k)}$ and the document-level distributions over topics $\theta^{(d)}$, we have the full generative process for LDA, given in Algorithm 2, with a directed graphical model diagram provided in Figure 1.4.

While the naive Bayes text model used a latent class representation for documents, LDA uses a mixed membership representation for documents, and a latent class representation for words. Thus, we can think of LDA as *clustering the words*, while performing a *soft clustering on the documents*. LDA topic models are sometimes called *admixture* topic models, with reference to genetic admixture models in population genetics, which model the process of mixing genes from multiple population groups through interbreeding (Pritchard *et al.* ,

Algorithm 2 Generative process for LDA

- For each topic k
  - $\Phi^{(k)} \sim \text{Dirichlet}(\beta)$ *//Sample a topic*
- For each document d
  - $\theta^{(d)} \sim \text{Dirichlet}(\alpha)$ *//Sample a distribution over topics*
  - For each word i in document d
    - $z_i^{(d)} \sim \text{discrete}(\theta^{(d)})$ *//Sample a topic*
    - $w_i^{(d)} \sim \text{discrete}(\Phi^{(z_i^{(d)})})$ *//Sample a word*

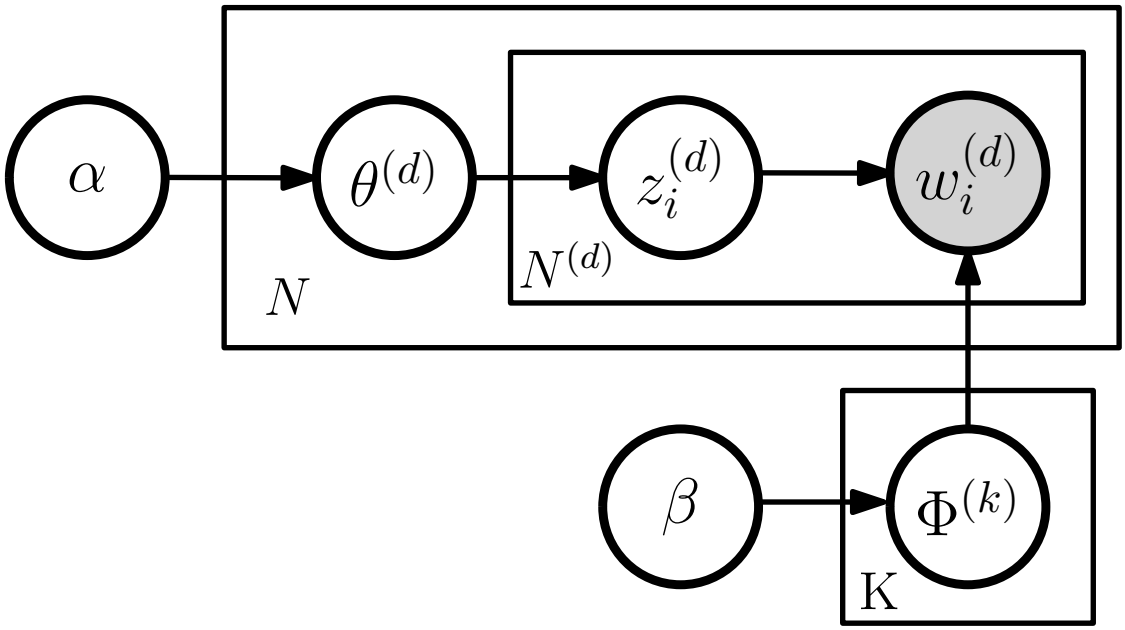


Figure 1.4: Directed graphical model diagram for LDA

2000). While an individual may acquire genes from several different ancestral populations, in LDA a document may acquire words from several different topics.

1.5.3 The Collapsed Representation of LDA, Priors and Polya Urn Models

The LDA model uses Dirichlet priors for both the topics and the distributions over topics. These distributions are *conjugate* to the discrete distributions, and to the multinomial dis-

tributions which arise from repeatedly drawing from these discrete distributions, once for each word. This allows us to marginalize ("*collapse*") out the topics and distributions over topics, and reason only about the latent topic assignments $\mathbf{z}$. This leads to much faster mixing when performing Gibbs sampling to infer the model from data, and has other advantages such as fewer, simpler update equations. This *collapsed Gibbs sampling* algorithm, due to Griffiths & Steyvers (2004), consists of iterating the following sampling update for each word in each document:

$$Pr(z_i^{(d)} = k | z^{-(d,i)}, \ldots) \propto (n_k^{(d)-(d,i)} + \alpha_k) \frac{n_k^{(w_i^{(d)})-(d,i)} + \beta_{w_i^{(d)}}}{n_k^{-(d,i)} + \sum_w \beta_w} \ . \tag{1.17}$$

where $n_k^{(d)}$ is the number of words in document d assigned to topic k, $n_k^{(w_i^{(d)})}$ is the number of times word w is assigned to topic k, n_k is the number of times in the corpus that words are assigned to topic k, and $-(d,i)$ excludes the current topic assignment for word i of document d in the count.

The collapsed representation can help us to interpret the Dirichlet hyperparameters α and β. We can see in Equation 1.17 that these values get added to the counts of the topics when performing collapsed Gibbs sampling updates. Thus, we can think of them as extra "words" previously assigned by the priors.

To further hone our intuition regarding the hyperparameters, after collapsing we can interpret the compound distribution of the Dirichlet and the multinomial as an urn model. Let us consider this distribution by itself before proceeding to the full LDA model. Suppose we have the following model with a K-dimensional Dirichlet prior α, where every entry of α is a non-negative integer:

$$\theta \sim \text{Dirichlet}(\alpha) \tag{1.18}$$

$$\mathbf{x} \sim \text{multinomial}(\theta, N) \ . \tag{1.19}$$

Marginalizing out θ, it can be shown that this corresponds to an urn model, which draws the count vector $\mathbf{x}$ via

- Begin with an empty urn
- For each k, $1 \leq k \leq K$
    - add α_k balls of color k to the urn
- For each i, $1 \leq i \leq N$
    - Reach into the urn and draw a ball uniformly at random
    - Observe its color, k. Count it, i.e. add one to x_k
    - Place the ball back in the urn, along with a *new ball of the same color*.

This scheme is known as a *Polya urn scheme*, a *Polya* distribution or a *Dirichlet-multinomial* distribution (cf. Minka (2000)). We can also allow a real-valued α vector by including "partial" balls in the urn, which are selected with probability proportional to their α value. This is made clear by examining how real-valued Dirichlet parameters are used in Equation 1.17. Focusing on the first term, $(n_k^{(d)-(d,i)} + \alpha_k)$, which arises from drawing the last ball from such an urn model, we see that color (topic) k may be selected by choosing one of the $n_k^{(d)-(d,i)}$ previous balls, or choosing one of the α_k balls. If $\alpha_k < 1$, the chance to pick this "partial ball" becomes correspondingly smaller.

An important property of this model is that when a color is drawn, by adding a new ball of the same color it becomes more likely that the color will be drawn again. This is known as a "*rich get richer*" property, or the *Matthew effect* (Merton, 1968).[7] In the context of LDA, if we marginalize the $\theta^{(d)}$'s, we can write LDA as such an urn scheme, given in Algorithm 3.

[7]The "Matthew effect" refers to a Biblical quote: "For unto every one that hath shall be given, and he shall have abundance: but from him that hath not shall be taken even that which he hath" — Matthew 25:29.

Algorithm 3 Polya urn interpretation of the generative process for LDA documents

- For each topic k
  - $\Phi^{(k)} \sim \text{Dirichlet}(\beta)$ *//Sample a topic*
- For each document d
  - Document d has an empty urn
  - For each topic k
    - add α_k balls of color k to the urn for document d
  - For each word i in document d
    - Reach into the urn and draw a ball uniformly at random
    - Observe its color, k. Assign $z_i^{(d)} = k$
    - Sample a word by rolling a D-sided die, $w_i^{(d)} \sim \text{discrete}(\Phi^{(z_i^{(d)})})$
    - Place the ball back in the urn, along with a *new ball of the same color.*

When selecting the hyperparameters in LDA, we typically set them to be values much less than one, corresponding to only a small fraction of a ball of each "color" (topic) initially. Since an entire ball is added in each draw, it is very likely that previously selected colors will continue to be selected in the process. This leads to count vectors which are very sparse in the prior, which effectively acts as a sparsity-inducing regularizer which often improves interpretability and generalization performance. It can also be beneficial to learn the hyperparameters, typically using an asymmetric α parameter and a symmetric β parameter (Wallach *et al.* , 2009a; Asuncion *et al.* , 2009). Finally, conditional LDA models can be constructed by performing regression to learn a unique α for each document, in a model called Dirichlet multinomial regression (DMR) (Mimno & McCallum, 2008). The DMR model parameterizes the α's for each document d as

$$\alpha_k^{(d)} = \exp(\mathbf{x}^{(d)\intercal} \lambda^{(k)}) \ , \tag{1.20}$$

where $\mathbf{x}^{(d)}$ is a feature vector for document d, and $\lambda^{(k)}$ is a parameter vector for topic k. In Chapter 3 of this thesis we consider a variant of DMR which has an intuitive interpretation in terms of the Polya urn scheme. In this variant of the model, we perform *linear* regression

on the α's, and constrain λ to have positive entries. The α vector for each document is parameterized as

$$\alpha_k^{(d)} = \mathbf{x}^{(d)\intercal}\lambda^{(k)} + \alpha \ , \tag{1.21}$$

where $\mathbf{x}^{(d)}$ is binary. Thus, if feature j is present (i.e. $\mathbf{x}_j^{(d)} = 1$), then λ_{kj} is added to entry k of the Dirichlet prior for the document. This can be understood as placing $\lambda_{kj} + \alpha$ balls of color k into the urn for document d before beginning the Polya urn scheme to draw the document.

As a historical note, the use of Dirichlet priors in LDA was refined over several papers. Inspired by latent semantic analysis (LSA) (Deerwester *et al.* , 1990), Hofmann (1999a,b) proposed *probabilistic LSA* (PLSA), a model which is essentially equivalent to LDA but without the Dirichlet priors.[8] Blei *et al.* (2003) introduced the Dirichlet prior on the distributions over topics $\theta^{(d)}$. The addition of the Dirichlet prior on the topics themselves $\Phi^{(k)}$, which is now standard in LDA models, is due to Griffiths & Steyvers (2004).

1.5.4 LDA as Matrix Factorization

We have seen that many latent variable models can be understood as performing matrix factorization. LDA is a probabilistic version of the matrix factorization algorithm LSA, and as such can be understood as a probabilistic matrix factorization method (Hofmann, 1999a,b; Buntine & Jakulin, 2006; Griffiths *et al.* , 2007). Let $\mathbf{\Theta}$ be the $N \times K$ matrix with the $\theta^{(d)}$'s on the rows, and $\mathbf{\Phi}$ be the $D \times K$ matrix with the $\Phi^{(k)}$'s on the columns. Then conditioning on $\theta^{(d)}$ and $\mathbf{\Phi}$, the probability of any word i in document d being word j is

[8] PLSA is an instance of a slightly earlier model, the *aspect model* (Hofmann *et al.* , 1998).

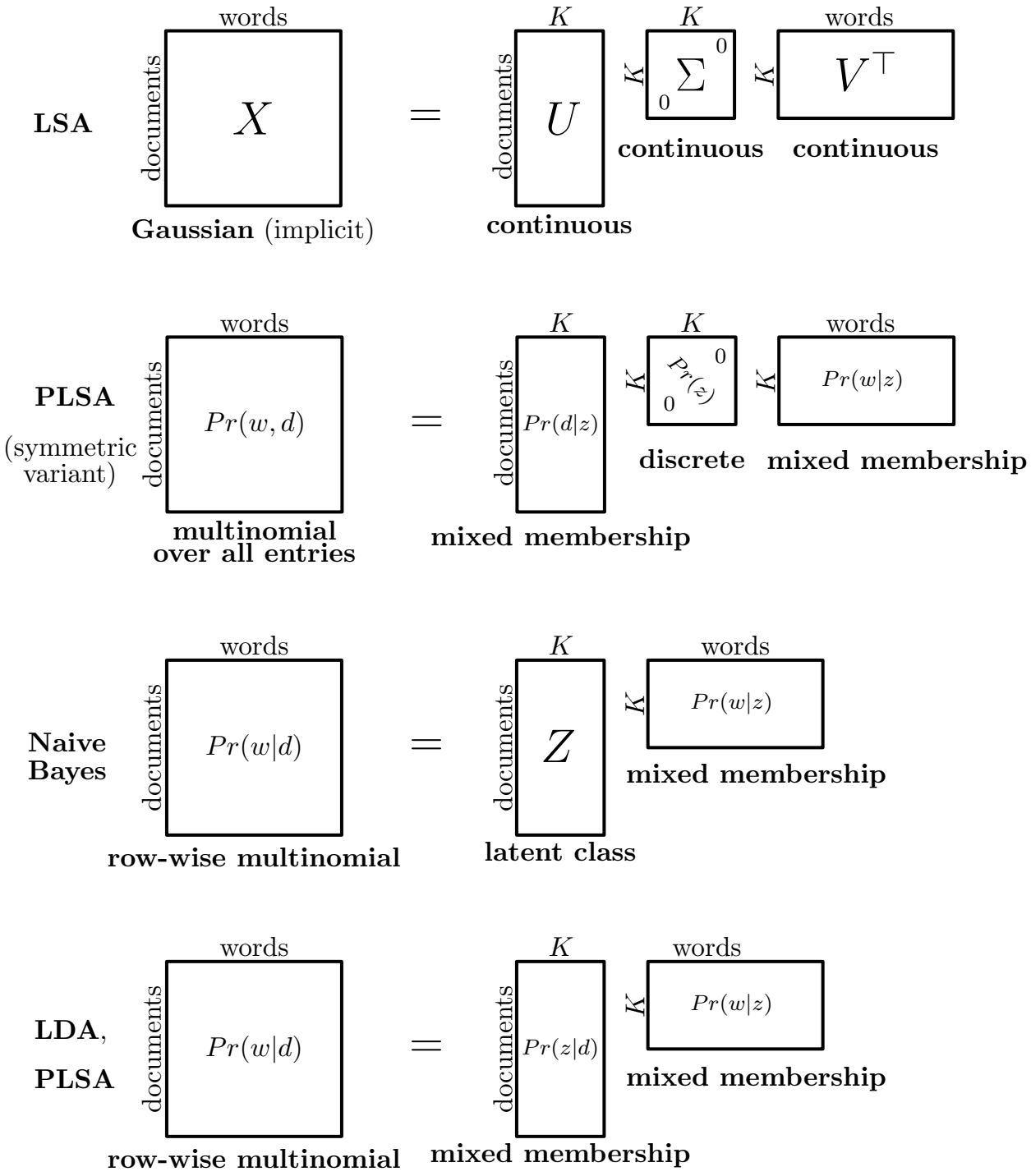


Figure 1.5: A comparison of the matrix factorization representations of LSA, PLSA and LDA topic models. Figure inspired by a diagram in Griffiths *et al.* (2007).

$$Pr(w_i^{(d)} = j|\theta^{(d)}, \mathbf{\Phi}) = \sum_k Pr(w_i^{(d)} = j, z_i^{(d)} = k|\theta^{(d)}, \mathbf{\Phi}) \tag{1.22}$$

$$= \sum_k Pr(z_i^{(d)} = k|\theta^{(d)}) Pr(w_i^{(d)} = j|z_i^{(d)} = k, \mathbf{\Phi}) \tag{1.23}$$

$$= \mathbf{\Theta}_d \mathbf{\Phi}_j^\intercal, \tag{1.24}$$

where $\mathbf{A}_i$ is the ith row of matrix $\mathbf{A}$. Let $\mathbf{x}$ be the $N \times D$ matrix of word counts for each document in the corpus. Then, in the framework of Section 1.2,

$$Pr(\mathbf{x}|\mathbf{\Theta}, \mathbf{\Phi}) = f(\eta) \tag{1.25}$$

$$\eta = \mathbf{\Theta}\mathbf{\Phi}^\intercal \ , \tag{1.26}$$

where f is a multinomial distribution on each row d of η which generates the appropriate number of words N_d for that document d. The matrix factorization interpretations of LDA and related topic models are given in Figure 1.5.

Having covered the essential background material, the remainder of this chapter describes the contributions made in this thesis, and an outline of its structure.

1.6 Contributions

This dissertation makes the following contributions:

- We develop a latent variable model for social networks observed over time, which we refer to as the *Dynamic Relational Infinite Feature model (DRIFT)*. The model posits a binary latent vector of features for each actor. The features are allowed to change

over time, including the introduction and deletion of new features, in a nonparametric Bayesian way.

- We show how to infer the semantics of such latent binary social network features in a model-based way, by incorporating text into the models using a topic modeling framework.

- Continuing the theme of *networks and text*, we introduce and evaluate a model known as *Topical Influence Regression* (TIR) which discovers latent influence relationships between scientific articles, leveraging both the citation network and the text of the articles. Although we focus on scientific articles, the model is a general framework for exploring document corpora where dependencies between the topics of the documents follow a Bayesian network.

- We design a fast, accurate, scalable and easy to implement algorithm for learning topic models. The algorithm, called *Stochastic Collapsed Variational Bayesian Inference, order Zero (SCVB0)*, is a stochastic algorithm which exploits the collapsed representation of LDA topic models. We evaluate the algorithm, showing the superior performance of the model over baselines in both the large and small scale settings.

- We also prove the convergence of SCVB0. The proof involves a detailed re-interpretation of the algorithm as an online expectation maximization algorithm for MAP estimation in topic models, where the MAP estimation is performed with adjusted hyperparameters.

- The evaluation of topic models, often performed by computing the likelihood for held-out documents and comparing this to baseline methods, is a difficult computational challenge which must be addressed for every new topic model variant or learning algorithm explored. We introduce *ratio-AIS*, an algorithm for comparing the performance of two topic models. The algorithm has lower empirical variance in its estimates of the

relative performance of a pair of models than previous approaches. A downside is the potential for a directional bias in the comparison when given insufficient computation. However, this can frequently be detected in practice by comparing the results of two runs of the algorithm performed in different "directions" of comparison. For most other methods, detection of convergence failures is very difficult to do in practice.

- We leverage ratio-AIS to provide an algorithm for efficiently evaluating the progress of topic model learning algorithms, at each iteration during training. The algorithm, which we call *iteration-AIS*, is shown empirically to find better per-iteration curves, in some cases with an order of magnitude less computational effort than previous methods.

1.7 Thesis Outline

The remainder of the dissertation proceeds as follows.

- Chapter 2 introduces models for social networks, exploring latent feature representations as they vary over time, and the automatic recovery of their semantics using a topic modeling framework.
- Chapter 3 proposes a latent variable model for inferring influence relationships between scientific articles.
- Chapter 4 describes and evaluates an efficient algorithm for learning topic models on corpora with millions of documents.
- Chapter 5 develops algorithms for evaluating topic models by computing the likelihood of held-out documents under the models. Specifically, methods are introduced for

comparing the performance of two models, and for evaluating the progress of topic model learning algorithms on a per-iteration basis.

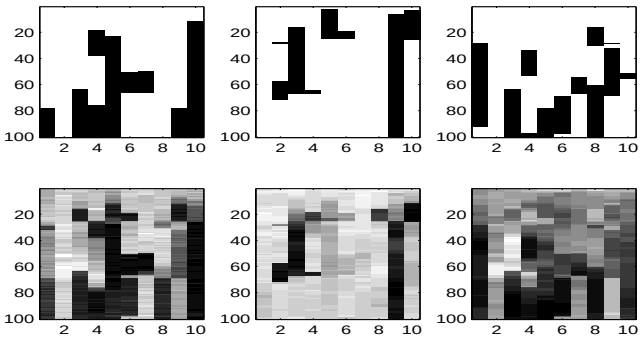
- We conclude in Chapter 6, and also indicate potential directions for future work.
- More detailed proofs and derivations are given in the appendices.

Chapter 2

A Dynamic Latent Feature Model for Social Networks

If you can look into the seeds of time,
and say which grain will grow and which will not ...
William Shakespeare, Macbeth

"Looking into the seeds of time," in a social network model. See Figure 2.2.



In Chapter 1, we overviewed how a latent variable model can be constructed by first selecting a latent representation, and then choosing priors and a link function to connect the latent variables and parameters to the observed data. We discussed Bayesian nonparametric priors which allow the latent representation to have potentially infinite dimension, and described

how sequential data can be modeled using hidden Markov dynamics. Finally, we reviewed statistical techniques that can be used to learn our new model from data.

This chapter makes all of these concepts more concrete with a study on the modeling of social networks using latent variable techniques. The starting point of our study is the latent feature relational model (LFRM) of Miller *et al.* (2009), which represents actors in a social network with latent vectors of binary features. Miller et al. showed how to learn the dimension of these vectors based on the data by using a nonparametric Bayesian prior distribution known as the Indian buffet process (Griffiths & Ghahramani, 2006).

The first research contribution of the chapter is to extend this model to facilitate the analysis of longitudinal social network data, where the network in question is measured repeatedly over time.[1] The model is evaluated both qualitatively and quantitatively, demonstrating its utility for this task. In the second half of the chapter, we also explore an extension of the LFRM designed to recover the semantics of the latent features by using topic models.

The chapter is organized as follows. We set the scene in Section 2.1 with an introduction to the modeling of social networks using statistical techniques. Section 2.2 describes the latent feature relational model, and Section 2.3 reviews a nonparametric prior distribution, called the Indian buffet process, which can be used to infer the latent dimensionality of the model.

The remainder of the chapter details original research contributions. Sections 2.4 and 2.5 discuss the proposed longitudinal extension and an inference algorithm for this model, respectively. In Section 2.6 we evaluate the model (relative to baselines) on prediction tasks for both simulated and real-world network data sets.

The second part of the chapter, in Section 2.7, shows an extension of the LFRM, which uses topic models to recover the semantics of the latent features. Section 2.7.1 details the proposed model, and a demonstration of the model for two exploratory data analysis tasks

[1] This portion of the chapter corresponds to collaborative work, published in Foulds *et al.* (2011).

is given in Section 2.7.2. Finally, section 2.8 concludes and summarizes the contributions of the chapter.

2.1 Social Network Analysis

Social network analysis is the study of the structure of social relationships between *social actors* such as humans, animals, countries and organizations. The quantitative modeling of social networks has a long history, dating back at least as far as the 1930s (e.g. Moreno (1934)). In recent years there has been a resurgence in interest in the analysis of social networks due to the advent of the internet and the meteoric rise of "web 2.0" digital social media technologies on websites such as Facebook, Twitter and Tumblr. Thanks in part to these digital social networks, an increasingly large percentage of human social interactions are occurring online, and hence can be more easily recorded and studied. This opens the way for sociologists to study human interaction behavior at a larger scale than has previously been possible, e.g. Gopalan *et al.* (2012); Sutton *et al.* (2013); Myers & Leskovec (2014). Many of the web 2.0 technology corporations which developed these digital social networks also rely on revenue generated from targeted advertising, meaning that understanding human interaction through social network data has in some cases become a financial necessity.

Social networks are *relational* data, meaning that they contain information that goes beyond the attributes of each of the data points considered in isolation. In this case, the data represent the *relationships* between social actors. In the field of social network analysis, a network on N actors is typically represented by an $N \times N$ binary matrix $\mathbf{Y}$, sometimes referred to as a *sociomatrix*. In this matrix, relations between actors i and j are represented by binary variables y_{ij}, which take the value 1 if a relationship exists and 0 otherwise. The sociomatrix can be interpreted as the adjacency matrix of a graph, with each node

being associated with an actor. For example, the nodes may represent individuals, with y_{ij} indicating the presence or absence of a friendship link between individual i and individual j.

This chapter concerns the statistical modeling of social network data. A useful feature of the *statistical* approach (as opposed to, say, purely graph theoretic analysis, qualitative study or task-oriented ad-hoc solutions) is that it provides a framework which can readily go beyond the simple assumption that the matrix $\mathbf{Y}$ is the whole story. For example, statistical models can be adapted and extended for handling both *missing* edge information, and for incorporating *additional* information such as weighted edges, time-varying edges, the effects of covariates for actors and edges, and additional data associated with the network such as text, audio and images (Wasserman, 1994).

To parameterize statistical network models, exponential-family random graph models (*ERGMs*) (Wasserman & Pattison, 1996) are a canonical approach, in the sense that any distribution over graphs can be represented as one of these models. ERGMs posit that the probability distribution over graphs can be written in exponential family form,

$$Pr(\mathbf{Y}) = \frac{\exp(\theta^{\intercal} t(\mathbf{Y}))}{Z} \ , \tag{2.1}$$

where θ is a parameter vector, $t(\mathbf{Y})$ maps $\mathbf{Y}$ to a vector of sufficient statistics, and $Z = \sum_{\mathbf{Y}'} \exp(\theta^{\intercal} t(\mathbf{Y}'))$ is a normalization constant, also known as the *partition function*. One specifies a family of ERGM models by choosing the sufficient statistics captured by t, such as the number of edges, triangles, star configurations and so on. Although ERGMs are a flexible and natural framework they can be difficult to work with, both from a computational and statistical estimation viewpoint. Even very simple ERGM models can suffer from degeneracy problems, where almost all of the probability mass belongs to a small number of graphs such as the complete graph or the empty graph (Handcock *et al.* , 2003). This leads to further problems with fitting the models, where MCMC algorithms become trapped at these

degenerate graphs. The intractability of the partition function also makes learning and inference very difficult with these models.

Latent variable modeling provides an alternative to ERGMs which largely avoids these difficulties. As discussed in Chapter 1, this approach uses hidden vectors $\mathbf{z}_i$ as "coordinates" to represent the characteristics of each network actor i. The edge indicator variables y_{ij} are modeled as being conditionally independent given the latent variables and parameters of the model.

Like ERGMs, latent variable methods can be justified by their generality. By symmetry, models of social networks should generally treat the nodes as being *exchangeable*, i.e. the ordering of the nodes should not affect the probability of the network. In this case, the matrix variable $\mathbf{Y}$ is said to be row-column exchangeable (Hoover, 1982). A variant of de Finetti's theorem, due independently to Hoover (1982) and Aldous (1985) states that if a model of an array random variable satisfies this exchangeability property, then that model can be represented as a latent variable model, in which the entries are conditionally independent given the latent variables. Hoff (2007) describes this result in the context of social network modeling.[2]

As per the matrix factorization framework of the previous chapter, edge probabilities in these models can often be cast in the following form:

$$Pr(y_{ij} = 1|\mathbf{Z}, \mathbf{W}, \rho, \xi, \epsilon) = g^{-1}(\mathbf{z}_i \mathbf{W} \mathbf{z}_j^\intercal + \rho_i + \xi_j + \epsilon) \ , \tag{2.2}$$

where g is a link function (e.g. g^{-1} is the logistic function), $\mathbf{W}$ is a $K \times K$ parameter matrix specifying how the latent variables interact, ρ and ξ are effects terms reflecting tendencies

[2]Hoff (2007) states that "any statistical model for a sociomatrix in which the nodes are exchangeable can be written as a latent variable model." This appears to neglect a caveat, in that the theorem is proved for arrays of infinite size, just as de Finetti's theorem holds for sequences of infinite length. In the real world, our matrices are finite. The theorem still applies if the matrix random variable can be embedded in a row-column exchangeable array-valued random variable of infinite size, and is likely to be approximately true for sufficiently large arrays otherwise.

of sending and receiving ties and ϵ is a parameter controlling network density. For example, the blockmodel (Fienberg & Wasserman, 1981; Nowicki & Snijders, 2001), which represents each actor with a latent class, is a classic latent variable modeling approach. We now turn to a more recent example, namely the latent feature relational model of Miller *et al.* (2009).

2.2 The Latent Feature Relational Model

According to sociological theories espoused by Simmel (1955), Feld (1981) and others, interactions between human actors are often mediated by shared *foci*. For example, the propensity of individuals to interact may be characterized by their job type (e.g., dentist, graduate student, professor), their leisure interests (e.g., mountain biking, salsa dancing), club memberships, location, social cliques, and so on. If such foci are known to us, we may include them in our models of the network as vectors of observed binary variables.

In real world applications, however, we are very unlikely to observe every possible property which may affect the probability of a link between actors. The latent feature relational model (LFRM) of Miller *et al.* (2009) assumes that at least some of them are unobserved. In this model, the presence and absence of each of these foci for a given actor is represented using a binary feature. In other words, each actor i is represented by a K-dimensional binary row vector $\mathbf{z}_i$, where each feature k corresponds to a property such as being interested in mountain biking. For the most part these binary features are assumed to be latent, although the $\mathbf{z}_i$'s may include observed features when they are available. Latent features can be understood as clusters or latent class memberships that are allowed to overlap, in contrast to the mutually exclusive classes of traditional blockmodels (Fienberg & Wasserman, 1981) from the social network literature.

In the LFRM model, the probability of an edge between two individuals is determined by the interactions of the features that are switched "on" for each of the individuals. For example, graduate students that salsa dance might have a much higher probability of having a link to professors that mountain bike, rather than to dentists that salsa dance. The relationship between feature k and feature k' is encoded by the entry $w_{kk'}$ of the $K \times K$ real-valued feature-feature interaction matrix $\mathbf{W}$. The inverse link function is chosen to be the "sigmoid" logistic function $\sigma(x) = \frac{1}{1+\exp(-x)}$, which squashes real-valued numbers into probability values between zero and one:

$$Pr(y_{ij} = 1 | \mathbf{Z}, \mathbf{W}, \rho, \xi, \epsilon) = \sigma(\mathbf{z}_i \mathbf{W} \mathbf{z}_j^\intercal + \rho_i + \xi_j + \epsilon) \ . \tag{2.3}$$

We can readily see that

$$\mathbf{z}_i \mathbf{W} \mathbf{z}_j^\intercal = \sum_k \mathbf{z}_{ik} (\mathbf{W} \mathbf{z}_j^\intercal)_k = \sum_{k:\mathbf{z}_{ik}=1} (\mathbf{W} \mathbf{z}_j^\intercal)_k = \sum_{k:\mathbf{z}_{ik}=1} \sum_{k':\mathbf{z}_{jk}=1} w_{kk'} \ . \tag{2.4}$$

Furthermore, the logistic function $\sigma(x)$ is the inverse of the logit function. It can be interpreted as converting its input $x = \log \frac{p}{1-p}$, the logarithm of the odds of p, into the probability p. Thus, we can interpret the model as performing the following process, where each for loop corresponds to one of the sums in Equation 2.4:

- For each feature k present for actor i
  - For each feature k' present for actor j
    - Add $w_{kk'}$ to the log-odds of the probability that $y_{ij} = 1$
- Add ρ_i, ξ_j and ϵ to the log-odds of the probability that $y_{ij} = 1$.

To finish specifying the model in a Bayesian context, we need to select priors for the parameters, consisting of the feature interaction matrix $\mathbf{W}$ and the effects and intercept terms ρ,

ξ, ϵ, as well as the latent feature matrix $\mathbf{Z}$. The parameters can simply be given elementwise univariate Gaussian priors,

$$w_{kk'} \sim \text{Gaussian}(0, \sigma_w) \tag{2.5}$$

$$\rho_i \sim \text{Gaussian}(0, \sigma_\rho) \tag{2.6}$$

$$\xi_j \sim \text{Gaussian}(0, \sigma_\xi) \tag{2.7}$$

$$\epsilon \sim \text{Gaussian}(0, \sigma_\epsilon) \tag{2.8}$$

although other choices are possible. For example, $\mathbf{W}$ can instead be constrained to be diagonal, meaning that actors i and j must *both* have feature k if it is to affect the probability of a link between them. Another option is to ensure that the $\mathbf{W}$'s are positive, by using an alternative prior such as an exponential distribution. This changes the semantics so that features can only increase the probability of a link – an assumption which is realistic in many scenarios, and which may make the features more interpretable by preventing the model from creating many small compensatory features to correct for the "mistakes" of other features.

We now consider the prior on the latent features. In the model, each column k of $\mathbf{Z}$ (i.e. each feature) is given its own probability of occurrence a_k, and the latent features are generated via

$$a_k \sim \text{beta}(\frac{\alpha}{K}, 1) \tag{2.9}$$

$$z_{ik} \sim \text{Bernoulli}(a_k)\ . \tag{2.10}$$

Here, in the prior for a_k the hyperparameter α is divided by K to bound the expected number of ones in the matrix as K is increased. This will become important for the nonparametric version of the model, which we discuss in the next section.

2.3 Nonparametric Modeling with the Indian Buffet Process

Miller *et al.* (2009) showed how to learn the dimensionality K of the LFRM's latent features automatically from the data in a Bayesian way, using a nonparametric Bayesian prior known as the Indian buffet process (IBP), due to (Griffiths & Ghahramani, 2006).[3] This section introduces the IBP prior, which then allows us to define the complete, nonparametric version of the LFRM model. It also foreshadows the derivation we will make as part of the new research described in subsequent sections of this chapter. Much of that more complicated derivation proceeds in parallel to this one, so this section may be useful as a gentle precursor to that work.

The IBP is a probability distribution on (equivalence classes of) sparse binary matrices $\mathbf{Z}$ with a finite number of rows but an unbounded number of columns. It can be derived by taking the limit of the distribution $Pr(\mathbf{Z})$ given by Equations 2.9 and 2.10 as the dimensionality K goes to infinity.

We outline the derivation of the IBP from Griffiths & Ghahramani (2006) here. A longer version is available in Griffiths & Ghahramani (2005). From Equation 2.10,

$$Pr(\mathbf{Z}|a) = \prod_k \prod_i Pr(z_{ik}|a_k) = \prod_k a_k^{m_k}(1-a_k)^{N-m_k} \ , \tag{2.11}$$

where m_k is the number of ones in column k. The first step in the derivation is to marginalize out the feature probabilities a_k to obtain $Pr(\mathbf{Z})$. This is done by exploiting the conjugacy relationship between the beta and Bernoulli distributions.

[3] Meeds *et al.* (2007) earlier described a similar IBP-based model for rectangular binary matrices.

Recall the probability density function for a beta distribution with parameters r and s:

$$Pr(p|r,s) = \frac{1}{B(r,s)} p^{r-1}(1-p)^{s-1} \tag{2.12}$$

$$B(r,s) = \int_0^1 p^{r-1}(1-p)^{s-1}dp = \frac{\Gamma(r)\Gamma(s)}{\Gamma(r+s)} \ , \tag{2.13}$$

where $B(r,s)$ is known as the beta function, and $\Gamma(n)$ is the gamma function, which is a generalization of the factorial function to real values. For integer n, $\Gamma(n) = (n-1)!$. An important property is that $\Gamma(n+1) = n\Gamma(n)$. In Equation 2.9, $r = \frac{\alpha}{K}$ and $s = 1$ for each a_k, in which case we have

$$B(\frac{\alpha}{K}, 1) = \frac{\Gamma(\frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K}+1)} = \frac{K}{\alpha} \ . \tag{2.14}$$

Making use of Equations 2.11 – 2.14, we can now obtain $Pr(\mathbf{Z})$ as

$$\begin{aligned}
Pr(Z) &= \int Pr(\mathbf{Z}|a)Pr(a)da \\
&= \prod_{k=1}^{K} \int \prod_i \Big(Pr(z_{ik}|a_k) \Big) Pr(a_k)da_k \\
&= \prod_{k=1}^{K} \int \Big(a_k^{m_k}(1-a_k)^{N-m_k} \Big) \frac{1}{B(\frac{\alpha}{K},1)} {a_k}^{\frac{\alpha}{K}-1}(1-a_k)^{1-1}da_k \\
&= \prod_{k=1}^{K} \frac{B(m_k + \frac{\alpha}{K}, N - m_k + 1)}{B(\frac{\alpha}{K},1)} \\
&= \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N-m_k+1)}{\Gamma(\frac{\alpha}{K}+N+1)} \ .
\end{aligned} \tag{2.15}$$

We would like to create our infinite-dimensional model by taking the limit of this distribution as K goes to infinity. However, this will cause the probability of any one matrix to go to zero. To avoid this, we can instead reason over equivalence classes of matrices. Griffiths and Ghahramani define such equivalence classes by way of the many-to-one function $lof(\mathbf{Z})$. The $lof(\mathbf{Z})$ function maps $\mathbf{Z}$ to its *left-order form*, in which the columns of $\mathbf{Z}$ are sorted left-

to-right in descending order of the binary numbers which they encode. The binary numbers are computed by treating each column as a sequence of bits, with the first row having the highest significance. Consider the equivalence relation $\mathbf{Z} \sim \mathbf{Z}'$ IFF $lof(\mathbf{Z}) = lof(\mathbf{Z}')$. The lof-equivalence class $[\mathbf{Z}]$ of $\mathbf{Z}$ is defined to be the set of matrices with the same left-order form as $\mathbf{Z}$, $\{\mathbf{Z}'|\mathbf{Z}' \sim \mathbf{Z}\}$.

Changing the order of the columns does not affect $Pr(\mathbf{Z})$ in Equation 2.15, so $Pr(\mathbf{Z})$ is column exchangeable and every element of $[\mathbf{Z}]$ has the same probability. So the probability of drawing an element of a particular lof-equivalence class $[\mathbf{Z}]$ is

$$\begin{aligned} Pr([\mathbf{Z}]) &= \sum_{\mathbf{Z}' \in [\mathbf{Z}]} Pr(\mathbf{Z}') \\ &= |[\mathbf{Z}]| Pr(\mathbf{Z}) \\ &= \binom{K}{K_0, K_1, \ldots, K_{2^N-1}} \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(\frac{\alpha}{K} + N + 1)}, \end{aligned} \tag{2.16}$$

where K_h is the number of columns of $\mathbf{Z}$ which encode the number h in binary, and $\binom{K}{K_0, K_1, \ldots, K_{2^N-1}} = \frac{K!}{\prod_{h=0}^{2^N-1} K_h!}$ is the multinomial coefficient of their counts. Griffiths and Ghahramani take the limit of this equation as K approaches infinity. The details are described in an appendix of Griffiths & Ghahramani (2005). The result of this limit defines the Indian buffet process,

$$\lim_{K \to \infty} Pr([\mathbf{Z}]) = \frac{\alpha^{K^+}}{\prod_{h=1}^{2^N-1} K_h!} \exp(-\alpha H_N) \prod_{k=1}^{K^+} \frac{(N - m_k)!(m_k - 1)!}{N!} , \tag{2.17}$$

where K^+ is the number of "active" features, i.e. those which contain a one, and $H_N = \sum_{j=1}^{N} \frac{1}{j}$ is the Nth harmonic number.

The IBP is named after a metaphorical process that also gives rise to this probability distribution, where N customers enter an Indian buffet restaurant and sample some subset of an infinitely long sequence of dishes. The first customer samples the first Poisson(α) dishes.

The ith customer then samples the previously sampled dishes proportionately to their popularity, with probability $\frac{m_k}{i}$, and samples Poisson(α/i) new dishes. The matrix $\mathbf{Z}$ of dishes sampled by customers is a draw from the IBP distribution.

A typical application of the IBP is to use it as a prior on a matrix that specifies the presence or absence of latent features which explain some observed data. The motivation of such an infinite latent feature model in this context is that the number of features can be automatically adjusted during inference, and hence does not need to be specified ahead of time. In the case of interest in this chapter, Miller *et al.* (2009) complete the full nonparametric Bayesian version of their LFRM model by using the IBP, as given in Equation 2.17, as the prior over $\mathbf{Z}$, instead of its finite version in Equations 2.9 and 2.10.

2.4 The Dynamic Relational Infinite Feature Model

Social network data are becoming increasingly digitized. Measurement of digital social networks can be automated, and so it is becoming much less costly to obtain repeated measurements of networks. The result is the increasing availability of social network data sets with a temporal component, such as email, online social networks, instant messaging, and so on. Consequently, there is considerable motivation to develop latent representations for network data over time. A primary contribution of this chapter is to develop a nonparametric Bayesian generative model for such time-varying social network data, by creating a dynamic extension of the LFRM model. We shall to refer to the model as the *Dynamic Relational Infinite Feature model* (DRIFT).

The DRIFT model leverages ideas from the infinite factorial HMM (Van Gael *et al.* , 2009), an approach that modifies the IBP into a factorial HMM with an unbounded number of hidden chains. Modeling temporal changes in latent variables for actors in a network has

previously been proposed by Sarkar & Moore (2005), Sarkar *et al.* (2007) and Fu *et al.* (2009). Sarkar & Moore (2005) and Sarkar *et al.* (2007) use continuous representations of the actors which evolve via Gaussian linear motion models, while we use binary latent feature representations which evolve by Markov switching. It is not always as straightforward to interpret unconstrained continuous representations like the ones used in those models, while the latent feature representation we use can be understood as performing community detection or clustering with overlapping clusters.

Fu *et al.* (2009) use a mixed membership model which does provide a community detection interpretation. This model considers the dynamics of the priors on the latent representations. In contrast, our approach explicitly models the dynamics of the actors' latent representations, which makes it more suitable for forecasting. Other statistical models for dynamic network data have been also proposed but typically deal only with the observed graphs $\mathbf{Y}^{(t)}$ (e.g. Snijders (2006); Butts (2008)) and do not use latent representations. Fan & Shelton (2009) model the relationships between (partially) *observed* features and social networks, in continuous time. Although they do not use a latent variable approach to model the network, they consider the case where the social *graph* itself is latent.

Subsequently to its publication in Foulds *et al.* (2011), an extension of the finite version of the DRIFT model was proposed by Heaukulani & Ghahramani (2013), in which the features of an actor's neighbors may propagate to the actor in each timestep. Another extension was proposed by Kim & Leskovec (2013), who explicitly model the birth and death of groups ("features," in our terminology), as well as the effects of non-membership on interaction. To the best of our knowledge, the proposed DRIFT model was the first nonparametric Bayesian latent variable model for social networks over time. We now detail the assumed generative process of the DRIFT model.

2.4.1 Generative Model

With time-varying ("*longitudinal*", or "*panel*") network data, we have a sequence of observed networks $\mathbf{Y}^{(t)}$ indexed by time $t = 1, \ldots, T$, rather than a single observed network $\mathbf{Y}$. In this chapter, we extend the LFRM of Miller *et al.* (2009) to model such data via a hidden Markov process. By introducing temporal dependence at the feature level, an individual's features $\mathbf{z}_i^{(t)}$ may change over time t as that individual's interests, group memberships, and behavior evolve. In turn the relational patterns in the networks $\mathbf{Y}^{(t)}$ will change over time as a function of the $\mathbf{z}_i^{(t)}$'s.

Thus, if Alice moves from Los Angeles to New York and abandons her hobby of playing tennis, she may be less likely to correspond with Bob, who lives in nearby Orange county and is an avid tennis player. However if Bob takes up mountain biking, and Alice is a road cyclist, they may resume a frequent pattern of correspondence due to their related interests.

Similarly to our treatment of the LFRM model, we start by defining the finite version of the model with K latent features. The final model is defined to be the limit of this model as K approaches infinity. First, we consider the "likelihood" portion of the model, which corresponds closely to the LFRM except that it is defined for multiple timesteps.[4]

The Likelihood

Let there be N actors, and T discrete time steps. At time t, we observe $\mathbf{Y}^{(t)}$, an $N \times N$ binary sociomatrix representing relationships between the actors at that time.[5] At each time step t there is an $N \times K$ binary matrix of latent features $\mathbf{Z}^{(t)}$, where $z_{ik}^{(t)} = 1$ if actor i has

[4] A frequentist statistician reserves the term *likelihood* for the probability of the data given the parameters, marginalizing out the "nuisance" latent variables. We use the term more loosely, to refer to the portion of the model which connects to the data.

[5] We will typically assume that $\mathbf{Y}^{(t)}$ is constrained to be symmetric, corresponding to a symmetric relation such as research collaboration, although it is straighforward to relax this assumption.

feature k at that time step. As in the LFRM, the $K \times K$ matrix $\mathbf{W}$ is a real-valued matrix of weights, determining the way in which pairs of features affect the network. The edges between actors at time t are assumed to be conditionally independent given $\mathbf{Z}^{(t)}$ and $\mathbf{W}$. The probability of each edge is as in the LFRM:

$$Pr(y_{ij}^{(t)} = 1 | \mathbf{Z}, \mathbf{W}, \rho, \xi, \epsilon) = \sigma(\mathbf{z}_i^{(t)} \mathbf{W} \mathbf{z}_j^{(t)\intercal} + \rho_i + \xi_j + \epsilon) \ , \tag{2.18}$$

where $\mathbf{z}_i^{(t)}$ is the ith row of $\mathbf{Z}^{(t)}$, and all other terms are defined as in Equation 2.3.

A Prior on the Latent Variables and their Dynamics

In the prior distribution on the latent variables there are assumed to be null states $z_{ik}^{(0)} = 0$ before the process begins, implying that each feature is effectively "off" before the first timestep. Each feature k for each actor i is given independent Markov dynamics, wherein if its current state is zero, the next value is distributed Bernoulli with a_k, otherwise it is distributed Bernoulli with the persistence parameter b_k for that feature. In other words, the transition matrix for actor i's kth feature is $\mathbf{Q}^{(ik)} = \left(\begin{smallmatrix} 1-a_k & a_k \\ 1-b_k & b_k \end{smallmatrix} \right)$. These Markov dynamics resemble the infinite factorial hidden Markov model (iFHMM) of Van Gael *et al.* (2009).

Note that $\mathbf{W}$ is not time-varying, unlike $\mathbf{Z}$. This means that the semantics of the features themselves do not evolve over time; rather, the network dynamics are determined by the changing presence and absence of the features for each actor.

In the model, the a_k's are given prior probability beta$(\frac{\alpha}{K}, 1)$, which is the same prior as for the features in the IBP. Importantly, this choice of prior allows for the number of introduced (i.e. "activated") features to have finite expectation when $K \to \infty$, with the expected number of "active" features being controlled by hyper-parameter α. The b_k's are drawn from a beta distribution. The prior on the latent variables corresponds to N iFHMM chains running in

parallel, one for each actor in the network, but sharing the same transition parameters a and b for all of the actors' iFHMM chains.

We also need to specify priors for the parameters involved in the likelihood. We use the same priors as for the LFRM. Specifically, the $w_{kk'}$'s are each assumed to be drawn from a univariate Gaussian with mean zero, as are the ρ, ξ and ϵ.

Complete generative model

Ignoring the intercept and effects terms for simplicity, the full generative model is

$$
\begin{aligned}
a_k &\sim \text{beta}(\frac{\alpha}{K}, 1) \\
b_k &\sim \text{beta}(\gamma, \delta) \\
z_{ik}^{(0)} &= 0 \\
z_{ik}^{(t)} &\sim \text{Bernoulli}(a_k^{1-z_{ik}^{(t-1)}} b_k^{z_{ik}^{(t-1)}}) \\
w_{kk'} &\sim \text{Gaussian}(0, \sigma_w) \\
y_{ij}^{(t)} &\sim \text{Bernoulli}(\sigma(\mathbf{z}_i^{(t)} \mathbf{W} \mathbf{z}_j^{(t)\intercal})) \ .
\end{aligned}
$$

Our proposed framework is illustrated with a directed graphical model in Figure 2.1. The model is a factorial hidden Markov model with a hidden chain for each actor-feature pair, and with the observed variables being the networks ($\mathbf{Y}$'s). It is also possible to include additional covariates by adding regression terms inside the logistic function (Miller *et al.* , 2009). These terms do not complicate inference, and this allows for the potential use of the model for operating in a "semi-parametric" fashion, where the nonparametric latent feature model is used to "absorb" any additional structure not captured by the covariates. In our experiments we do not consider covariates or sender and receiver effects terms, and only use the additional intercept term ϵ that determines the prior probability of an edge when no

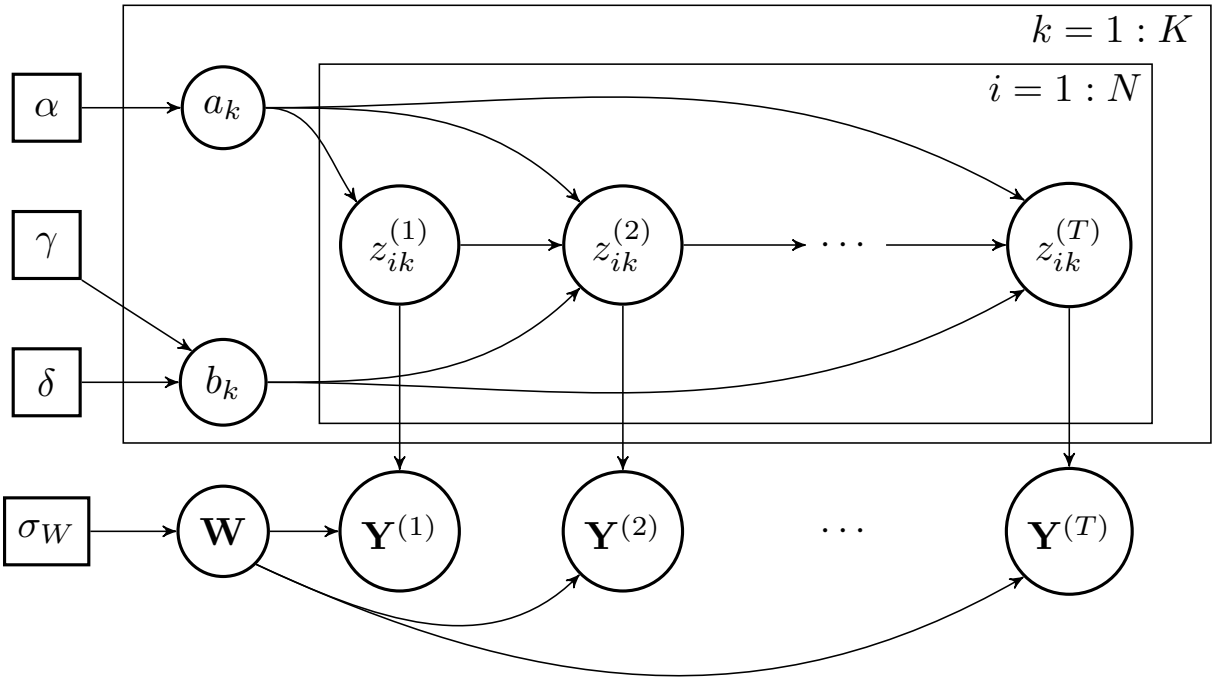


Figure 2.1: Graphical model for the finite version of DRIFT. The full model is defined to be the limit of this model as $K \to \infty$.

features are present. Note that including ϵ does not increase the generality of the model, as the same effect could be achieved by introducing an additional feature shared by all actors. However, it does free the latent variables from the responsibility of having to explain the base density of the network.

2.4.2 Taking the Infinite Limit

The full model is defined to be the limit of the above model as the number of features approaches infinity. Let c_k^{00}, c_k^{01}, c_k^{10}, c_k^{11} be the total number of transitions from $0 \to 0$, $0 \to 1$, $1 \to 0$, $1 \to 1$ over all actors, respectively, for feature k. In the finite case with K features, we can write the prior probability of $\mathbf{Z} = (\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \ldots, \mathbf{Z}^{(T)})$ as:

$$Pr(\mathbf{Z}|a,b) = \prod_{k=1}^{K} a_k^{c_k^{01}} (1-a_k)^{c_k^{00}} b_k^{c_k^{11}} (1-b_k)^{c_k^{10}} \ . \tag{2.19}$$

Before taking the infinite limit, using Equations 2.12 – 2.14 we integrate out the transition probabilities with respect to their priors,

$$Pr(\mathbf{Z}|\alpha,\gamma,\delta) = \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(\frac{\alpha}{K}+c_k^{01})\Gamma(1+c_k^{00})\Gamma(\gamma+\delta)\Gamma(\delta+c_k^{10})\Gamma(\gamma+c_k^{11})}{\Gamma(\frac{\alpha}{K}+c_k^{00}+c_k^{01}+1)\Gamma(\gamma)\Gamma(\delta)\Gamma(\gamma+\delta+c_k^{10}+c_k^{11})} \ . \tag{2.20}$$

Similarly to the construction of the IBP, we must compute the infinite limit for the probability distribution on *equivalence classes* of the binary matrices, rather than on the matrices directly. The next step is to define these equivalence classes.

Recall that $\mathbf{Z}$ is a length T sequence of $N \times K$ matrices $\mathbf{Z}^{(t)}$. We rewrite it as an $NT \times K$ matrix $\bar{\mathbf{Z}}$, where the sequences of feature values for each actor over time are concatenated one after another to form a single matrix, using the ordering of the actors in $\mathbf{Y}$. Define the *history* of a column k of $\bar{\mathbf{Z}}$ to be the binary number that it encodes when its entries are interpreted to be binary digits. As stated in Section 2.3, $lof(\mathbf{M})$ maps a binary matrix $\mathbf{M}$ to its *left-order form*, where the columns of $\mathbf{M}$ are permuted so that their histories are sorted in decreasing order. Note that the model is column-exchangeable so transforming $\bar{\mathbf{Z}}$ to lof does not affect its probability. We denote $[\bar{\mathbf{Z}}]$ to be the set of binary matrices that have the same left-order form as $\bar{\mathbf{Z}}$. Let K_h be the number of columns in $\bar{\mathbf{Z}}$ whose history has decimal value h. Then the number of elements of $[\bar{\mathbf{Z}}]$ equals $\binom{K}{K_0, K_1, \ldots, K_{2^{NT}-1}} = \frac{K!}{\prod_{h=0}^{2^{NT}-1} K_h!}$, yielding the following:

$$\begin{aligned} Pr([\bar{\mathbf{Z}}]) &= \sum_{\bar{\mathbf{Z}}' \in [\mathbf{Z}]} Pr(\bar{\mathbf{Z}}'|\alpha,\gamma,\delta) \\ &= \binom{K}{K_0, K_1, \ldots, K_{2^{NT}-1}} Pr(\bar{\mathbf{Z}}|\alpha,\gamma,\delta). \end{aligned} \tag{2.21}$$

The limit of $Pr([\bar{\mathbf{Z}}])$ as $K \to \infty$ can be derived similarly to the iFHMM model. Let K^+ be the number of features that have at least one non-zero entry for at least one actor. Using a lemma of Van Gael *et al.* (2009), we obtain

$$\lim_{K\to\infty} Pr([\bar{\mathbf{Z}}]) = \frac{\alpha^{K^+}}{\prod_{h=0}^{2^{NT}-1} K_h!} \exp(-\alpha H_{NT}) \prod_{k=1}^{K^+} \frac{(c_k^{01}-1)!c_k^{00}!\Gamma(\gamma+\delta)\Gamma(\delta+c_k^{10})\Gamma(\gamma+c_k^{11})}{(c_k^{00}+c_k^{01})!\Gamma(\gamma)\Gamma(\delta)\Gamma(\gamma+\delta+c_k^{10}+c_k^{11})}, \tag{2.22}$$

where $H_i = \sum_{k=1}^{i} \frac{1}{k}$ is the ith harmonic number.[6]

2.5 MCMC Inference Algorithm

We now describe how to perform posterior inference for DRIFT using a Markov chain Monte Carlo algorithm. The algorithm performs blocked Gibbs sampling updates on subsets of the variables in turn.

First, we consider the sampling procedure for the latent variables $\mathbf{Z}$. In practice, we cannot store the infinite dimensional matrices needed to represent these variables explicitly. Fortunately, we need only store the K^+ "active" features, with the columns for the non-represented features consisting of all zeros. Since the number of active features is not fixed, we need a sampler which allows K^+ to grow and shrink per sampling iteration as dictated by the posterior distribution of the model.

To this end, we adapt the "slice sampling" procedure originally derived by Teh *et al.* (2007b) for the IBP, which makes use of the stick-breaking construction of the IBP portion of DRIFT. Slice sampling (Neal, 2003) is an auxiliary variable MCMC sampling strategy which can be useful when it is difficult to specify an optimal proposal distribution for Metropolis-Hastings moves. Although the particular technique of Teh et al. is not a direct application of Neal's algorithm, it uses an auxiliary variable very much in the spirit of that algorithm. Teh et

[6] A thorough derivation of the lemma used here is given in Van Gael's Ph.D. thesis (Van Gael, 2011).

al. showed that this method mixes better than the naive sampling algorithm, which in non-conjugate models requires Metropolis-Hastings moves when introducing new features. The method is also straightforward to implement, making it useful as a general-purpose method for sampling IBP-based models in the non-conjugate case.

Since the distribution on the a_k's is identical to the feature probabilities in the IBP model, the "stick breaking" properties of these variables carry over to our model. Specifically, if we order the features so that they are strictly decreasing in a_k, Teh *et al.* (2007b) showed that we can write the a_k's in "stick-breaking" form as

$$v_k \sim \text{beta}(\alpha, 1) \tag{2.23}$$

$$a_k = v_k a_{k-1} = \prod_{l=1}^{k} v_l \ , \tag{2.24}$$

Here, we can metaphorically view this process for generating the a_k's as beginning with a "stick" a_0 of length 1. At iteration k, we take the stick of length a_{k-1} and break it into two pieces, retaining a portion v_k of it. This becomes our new stick length a_k, and we proceed recursively to break a_k further to obtain a_{k+1}, and so on.

To leverage this construction of the IBP in a sampling context, the technique is to introduce an auxiliary "slice" variable s to adaptively truncate the represented portion of $\mathbf{Z}$ while still performing correct MCMC inference on the infinite model. We first sample the slice variable s uniformly on the set of numbers between 0 and a_k for the active feature k that has the smallest a_k:

$$s|\mathbf{Z}, a \sim \text{Unif}(0, \min_{k:\exists t,i,\mathbf{Z}_{ik}^{(t)}=1} a_k) \ . \tag{2.25}$$

Having drawn s, we condition on it for the remainder of the MCMC iteration, which forces the features for which $a_k < s$ to be inactive. This ensures that we will not have to introduce such

features into the represented portion of $\mathbf{Z}$ at this time. We now extend the representation so that we have a and b parameters for all features k such that $a_k \geq s$. Here we are using the semi-ordered stick-breaking representation of the IBP feature probabilities (Teh *et al.* , 2007b), so we view the active features as being unordered, while the inactive features are in decreasing order of their a_k's. Consider the matrix whose columns each correspond to an inactive feature and consist of the concatenation of each actor's $\mathbf{Z}$ values at each time for that feature. Since each entry in each column is distributed Bernoulli(a_k), we can view this as the inactive portion of an IBP with $M = NT$ rows. So we can follow Teh *et al.* (2007b) to sample the a_k's for each of these features:

$$Pr(a_k|a_{k-1}, \mathbf{Z}^{:}_{:,>k} = 0) \propto \exp(\alpha \sum_{i=1}^{M} \frac{1}{i}(1-a_k)^i) a_k^{\alpha-1}(1-a_k)^M \mathbb{I}(0 \leq a_k \leq a_{k-1}) \ , \quad (2.26)$$

where $\mathbf{Z}^{:}_{:,>k}$ is the entries of $\mathbf{Z}$ for all timesteps and all actors, with feature index greater than k. We do this for each introduced feature k, until we find an a_k such that $a_k < s$, at which point we cease extending the representation. The $\mathbf{Z}$s for these features are initially set to $\mathbf{Z}^{(t)}_{ik} = 0$, and the other parameters ($\mathbf{W}$, b_k) for these are sampled from their priors, e.g. $b_k \sim \text{beta}(\gamma, \delta)$.

Having adaptively chosen the number of features to consider, we can now sample the feature values. The $\mathbf{Z}$s are sampled one $\mathbf{Z}_{ik}$ chain at a time via the forward-backward algorithm (Scott, 2002). In the forward pass, we create the dynamic programming cache, which consists of the 2×2 matrices $\mathbf{P}_2 \ldots \mathbf{P}_T$, where $\mathbf{P}_t = (p_{trs})$. Letting θ_{ik} be all other parameters and hidden variables not in $\mathbf{Z}_{ik}$, we have the following standard recursive computation,

$$\begin{aligned} p_{trs} &= Pr(\mathbf{Z}^{(t-1)}_{ik} = r, \mathbf{Z}^{(t)}_{ik} = s | \mathbf{Y}^{(1)} \ldots \mathbf{Y}^{(t)}, \theta_{ik}) \\ &\propto \pi_{t-1}(r|\theta) \mathbf{Q}^{(ik)}(r,s) Pr(\mathbf{Y}^{(t)} | \mathbf{Z}^{(t)}_{ik} = s, \theta_{ik}), \\ \text{where} \quad \pi_t(s|\theta) &= Pr(\mathbf{Z}^{(t)}_{ik} = s | \mathbf{Y}^{(1)} \ldots \mathbf{Y}^{(t)}, \theta_{ik}) = \sum_r p_{trs} \ . \end{aligned} \quad (2.27)$$

In the backward pass, we sample the states in backwards order via $\mathbf{Z}_{ik}^{(T)} \sim \pi_T(.|\theta_{ik})$, and $Pr(\mathbf{Z}_{ik}^{(t)} = s) \propto p_{t+1,r,\mathbf{Z}_{ik}^{(t+1)}}$. We drop all inactive columns, as they are relegated to the non-represented portion of $\mathbf{Z}$.

Next, we sample the IBP hyperparameter α, for which we assume a gamma(α_a, α_b) hyperprior, where α_a is the shape parameter and α_b is the inverse scale parameter. We temporarily integrate out the a_k's, after which $Pr(\mathbf{Z}|\alpha) \propto \alpha^{K^+} e^{-\alpha H_{NT}}$ from Equation 2.22. By Bayes' rule, $Pr(\alpha|\mathbf{Z}) \propto \alpha^{K^+ + \alpha_a - 1} e^{-\alpha(H_{NT} + \alpha_b)}$ is a gamma$(K^+ + \alpha_a, H_{NT} + \alpha_b)$.

Next, we sample the a's and b's for non-empty columns. Starting with the finite model, using Bayes' rule and taking the limit as $K \to \infty$, we find that $a_k \sim \text{beta}(c_k^{01}, c_k^{00} + 1)$. It is straightforward to show that $b_k \sim \text{beta}(c_k^{11} + \gamma, c_k^{10} + \delta)$.

We next sample $\mathbf{W}$, which proceeds similarly to Miller *et al.* (2009). Since it is non-conjugate, we use Metropolis-Hastings updates on each of the entries in $\mathbf{W}$. For each entry $w_{kk'}$, we propose $w_{kk'}^* \sim \text{Gaussian}(w_{kk'}, \sigma_w)$. When calculating the acceptance ratio, since the proposal distribution is symmetric, the transition probabilities cancel, leaving the standard acceptance probability

$$Pr(\text{accept } w_{kk'}^*) = \min\{\frac{Pr(\mathbf{Y}|w_{kk'}^*, \ldots)Pr(w_{kk'}^*)}{Pr(\mathbf{Y}|w_{kk'}, \ldots)Pr(w_{kk'})}, 1\} \ . \tag{2.28}$$

The intercept term ϵ and the effects terms ρ_i, ξ_j are also sampled using Metropolis-Hastings updates with a Gaussian proposal centered on the current location. Slice sampling (Neal, 2003) is an alternative option for sampling the real-valued parameters, which has the advantage that it is robust to its step size parameter and it does not reject proposed moves, unlike Metropolis-Hastings. On the other hand, Metropolis-Hastings is simpler and requires fewer evaluations of the likelihood per update.

2.6 Experimental Analysis

We analyze the performance of DRIFT on synthetic and real-world longitudinal networks. The evaluation tasks considered are predicting the network at time t given networks up to time $t-1$, and prediction of missing edges. For the forecasting task, we estimate the posterior predictive distribution for DRIFT,

$$Pr(\mathbf{Y}^t|\mathbf{Y}^{1:(t-1)}) = \sum_{\mathbf{Z}^t} \sum_{\mathbf{Z}^{1:(t-1)}} Pr(\mathbf{Y}^t|\mathbf{Z}^t) Pr(\mathbf{Z}^t|\mathbf{Z}^{t-1}) Pr(\mathbf{Z}^{1:(t-1)}|\mathbf{Y}^{1:(t-1)}), \tag{2.29}$$

in Monte Carlo fashion by obtaining samples for $\mathbf{Z}^{1:(t-1)}$ from the posterior, using the MCMC procedure outlined in the previous section. For each sample, we then repeatedly draw $\mathbf{Z}^t$ by incrementing the Markov chains one step from $\mathbf{Z}^{(t-1)}$, using the learned transition matrix. Averaging the likelihoods of these samples gives a Monte Carlo estimate of the predictive distribution. This procedure also works in principle for predicting more than one timestep into the future.

An alternative task is to predict the presence or absence of edges between pairs of actors when this information is missing. Assuming that edge data are missing completely at random, we can extend the MCMC sampler to perform Gibbs updates on missing edges by sampling the value of each pair independently using Equation 2.18. To make predictions on the missing entries, we estimate the posterior mean of the predictive density of each pair by averaging the edge probabilities of Equation 2.18 over the MCMC samples. This was found to be more stable than estimating the edge probabilities from the sample counts of the pairs.

In our experiments, we compare DRIFT to its static counterpart, the LFRM. Several variations of the LFRM were considered. LFRM (all) treats the networks at each timestep as i.i.d. samples. For forecasting, LFRM (last) only uses the network at the last time step $t-1$ to predict timestep t, while for missing data prediction LFRM (current) trains an LFRM

model on the training entries for each timestep. The inference algorithm for the LFRM is the algorithm for DRIFT with one time step. For both DRIFT and the LFRM, all variables were initialized by sampling them from their priors.

We also consider a baseline method which has a posterior predictive probability for each edge proportional to the number of times that edge has appeared in the training data (i.e. a multinomial), using a symmetric Dirichlet prior with concentration parameter set to increase with the amount of training data. This can be interpreted as an urn model, where each edge in the network is associated with balls of a certain color. Selecting an edge corresponds to drawing a ball from an urn, and then placing that ball back into the urn along with a new ball of the same color. The concentration parameter corresponds to the number of balls of each color in the urn initially. In this case, we set the concentration parameter to the number of timesteps divided by 5. We also consider a simpler method ("naive") whose posterior predictive probability for all edges is proportional to the mean density of the network over the observed time steps. In the experiments, hyperparameters were set to $\alpha_a = 3$, $\alpha_b = 1$, $\gamma = 3$, $\delta = 1$, and $\sigma_W = .1$. For the missing data prediction tasks, twenty percent of the entries of each data set, across all time points, were randomly chosen as a test set, and the algorithms were trained on the remaining entries.

2.6.1 Synthetic Data

We first evaluate DRIFT on synthetic data to demonstrate its capabilities. Ten synthetic datasets were each generated from a DRIFT model with 10 actors and 100 timesteps, using a $\mathbf{W}$ matrix with 3 features chosen such that the features were identifiable, and a different $\mathbf{Z}$ sampled from its prior for each dataset.

Given this data, our MCMC sampler draws 20 samples from the posterior distribution, with each sample generated from an independent chain with 100 burn in iterations. Figure 2.2

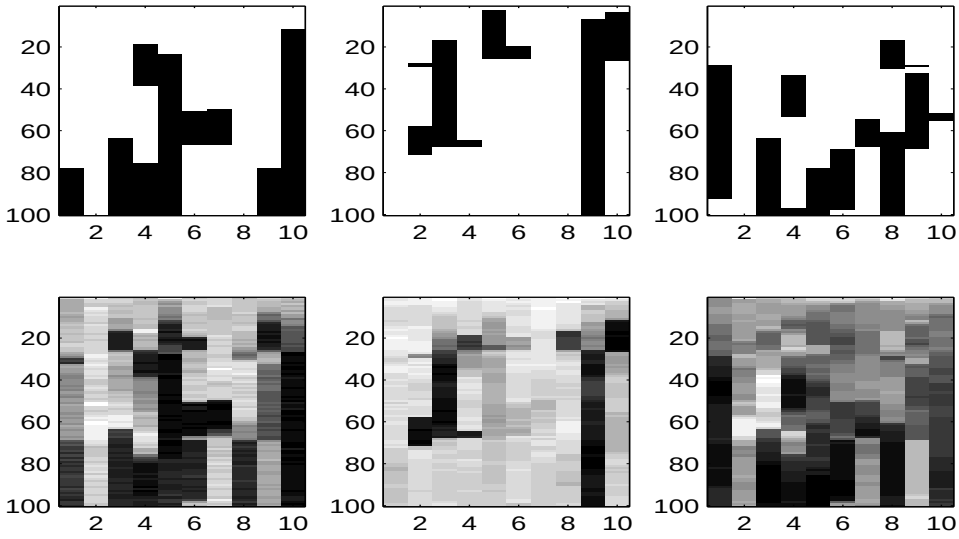
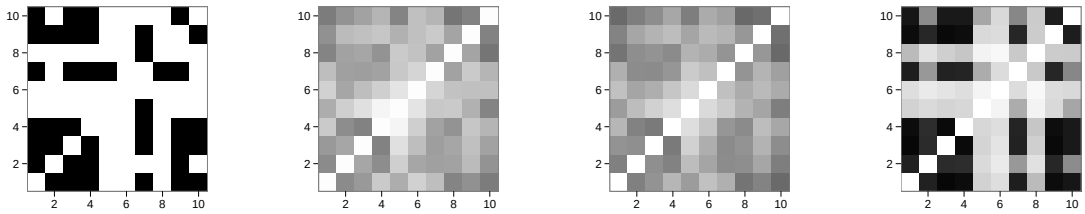


Figure 2.2: Ground truth (top) versus **Z**'s estimated by DRIFT (bottom) on synthetic data. Each image represents one feature, with rows corresponding to timesteps and columns corresponding to actors.

shows the **Z**s from one scenario, averaged over the 20 samples (with the number of features constrained to be 3, and with the features aligned so as to visualize the similarity with the true **Z**). This figure suggests that the **Z**s can be correctly recovered in this case, noting as in Miller *et al.* (2009) that the **Z**s and **W**s are not in general identifiable.

Table 2.1 shows the average AUC and log-likelihood scores for forecasting an additional network at timestep 101, and for predicting missing edges (the number of features was not constrained in these experiments). DRIFT outperforms the other methods in both log-likelihood and AUC on both tasks. This is because it is able to model the non-stationarity of the data.Figure 2.3 illustrates this with the held-out **Y** and the posterior predictive distributions for one forecasting task.



True $\mathbf{Y}$ Baseline LFRM (all) DRIFT

Figure 2.3: Held out $\mathbf{Y}$, and posterior predictive distributions for each method, on synthetic data.

Table 2.1: Log-likelihood and AUC on Enron

Synthetic Dataset	**Naive**	**Baseline**	**LFRM (last/current)**	**LFRM (all)**	**DRIFT**
Forecast LL	-31.6	-32.6	-28.4	-31.6	**−11.6**
Missing Data LL	-575	-490	-533	-478	**−219**
Forecast AUC	N/A	0.608	0.779	0.596	**0.939**
Missing Data AUC	N/A	0.689	0.675	0.691	**0.925**
Enron Dataset	**Naive**	**Baseline**	**LFRM last/current)**	**LFRM (all)**	**DRIFT**
Forecast LL	-141	-108	-119	-98.3	**−83.5**
Missing Data LL	-1610	-1020	-1410	-981	**−639**
Forecast AUC	N/A	0.874	0.777	0.891	**0.910**
Missing Data AUC	N/A	0.921	0.803	0.933	**0.979**

2.6.2 Enron Email Data

We also evaluate our approach on the widely-studied Enron email corpus (Klimt & Yang, 2004). The Enron data contains 34182 emails among 151 individuals over 3 years. We aggregated the data into monthly snapshots, creating a binary sociomatrix for each snapshot indicating the presence or absence of an email between each pair of actors during that month. In these experiments, we use a subset of the data involving interactions among the 50 individuals with the most emails.

For each month t, we train LFRM (all), LFRM (last), the baseline and DRIFT on all previous months 1 to $t-1$. In the MCMC sampler, we use 3 chains and a burn in length of 100,

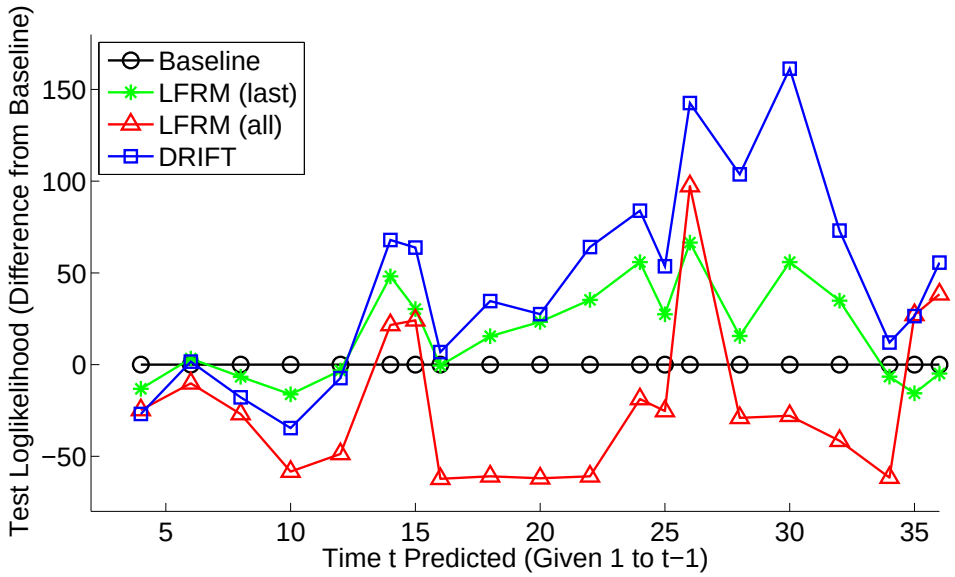
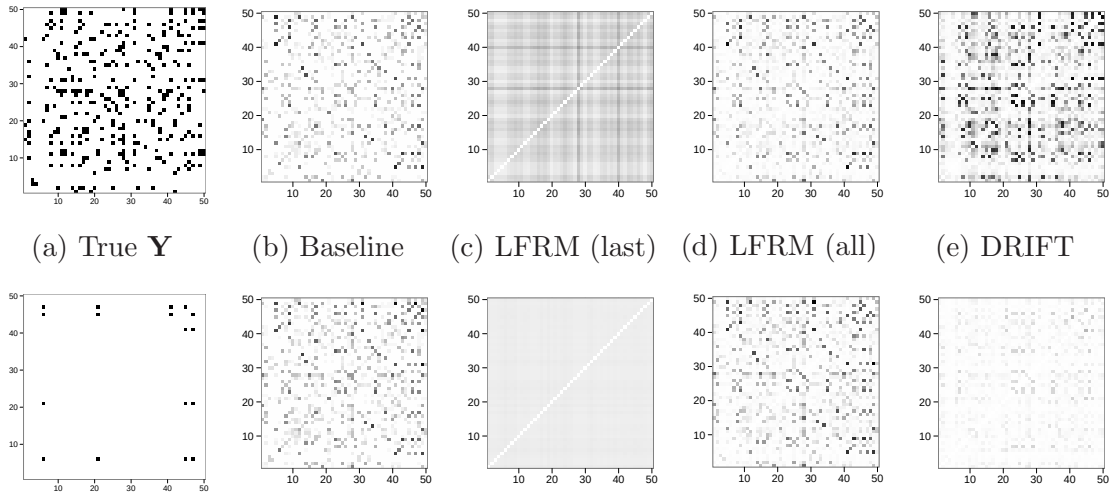


Figure 2.4: Test log-likelihood difference from baseline on Enron dataset at each time t.



(a) True $\mathbf{Y}$ (b) Baseline (c) LFRM (last) (d) LFRM (all) (e) DRIFT

Figure 2.5: Held out $\mathbf{Y}$ at time $t = 30$ (top row) and $t = 36$ (bottom row) for Enron, and posterior predictive distributions for each of the methods.

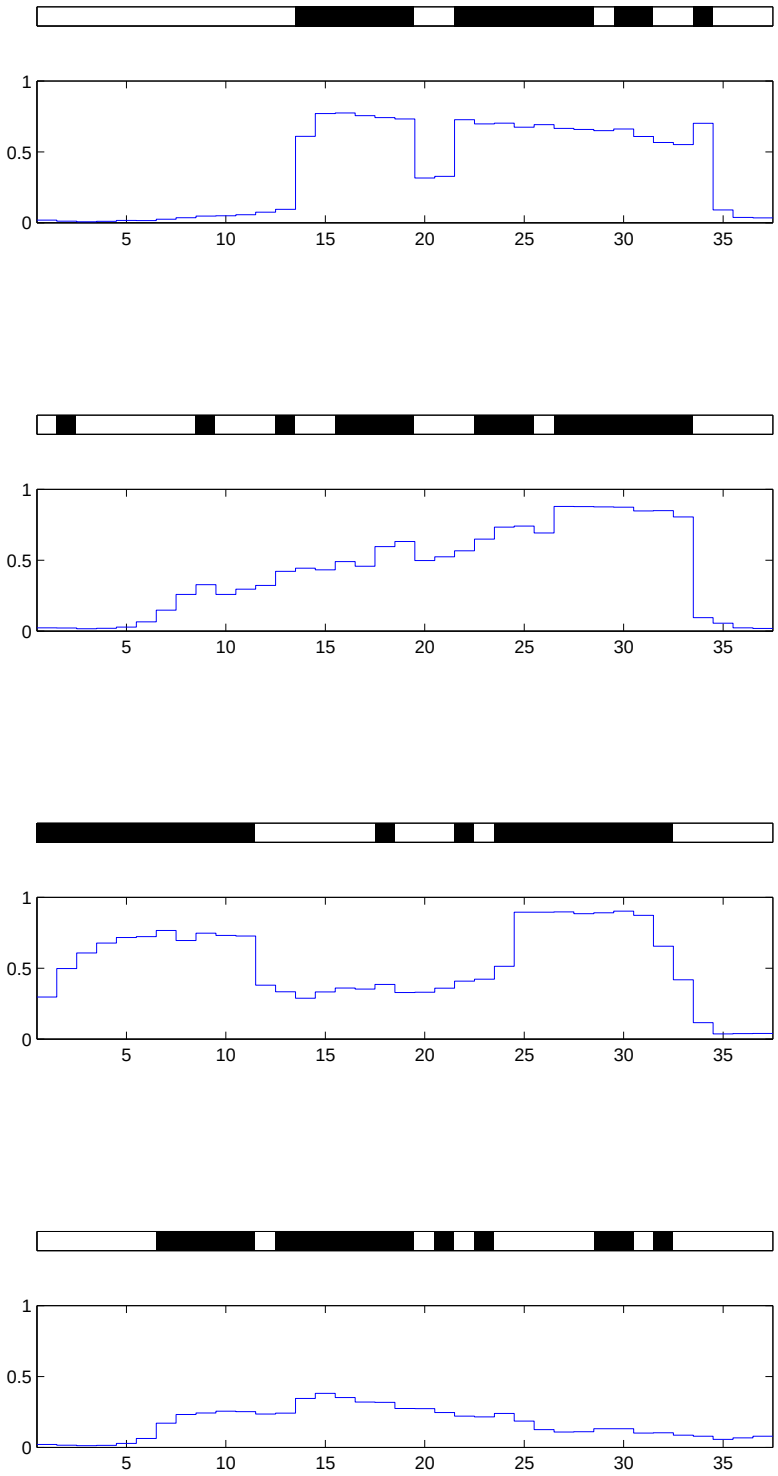


Figure 2.6: Estimated edge probabilities vs timestep for four pairs of actors from the Enron dataset. Above each plot the presence and absence of edges is shown, with black meaning that an edge is present.

k	**Baseline**	**LFRM** (current)	**LFRM** (all)	**DRIFT**
10	**10**	5	**10**	**10**
20	19	6	19	**20**
50	36	12	36	**48**
100	60	22	62	**90**
500	192	78	197	**301**
1000	285	142	290	**361**

Table 2.2: Number of true positives for the k missing entries predicted most likely to be an edge on Enron.

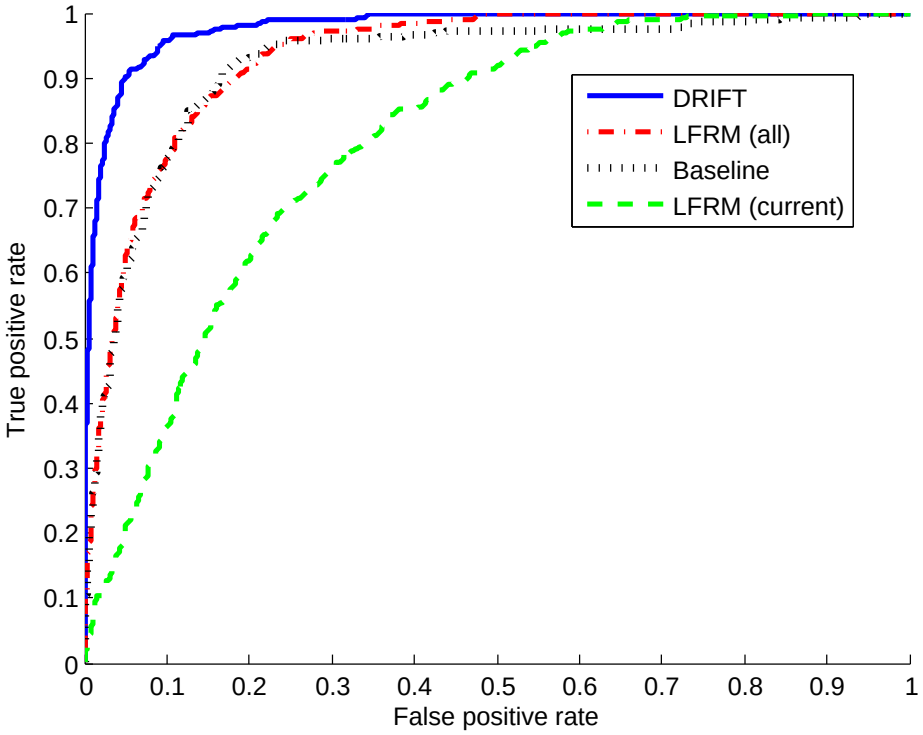


Figure 2.7: ROC curves for Enron missing data.

which we found to be sufficient. To compute predictions for month t for DRIFT, we draw 10 samples from each chain, and for each of these samples, we draw 10 different instantiations of $\mathbf{Z}^t$ by advancing the Markov chains one step. For LFRM, we simply use the sampled $\mathbf{Z}$'s from the posterior for prediction.

Table 2.1 shows the test log-likelihoods and AUC scores, averaged over the months from $t = 3$ to $t = 37$. Here, we see that DRIFT achieves a higher test log-likelihood and AUC than the LFRM models, the baseline and the "naive" method. Figure 2.4 shows the test log-likelihood for each time step t predicted (given 1 to $t - 1$). This plot suggests that all of the probabilistic models have difficulty beating the simple baseline early on (for $t < 12$). However, when t is larger, DRIFT performs better than the baseline and the other methods. For the last time step, LFRM (last) also does well relative to the other methods, since the network has become sparse at both that time step and the previous time step.

For the missing data prediction task, thirty MCMC samples were drawn for LFRM and DRIFT by taking only the last sample from each of thirty chains, with three hundred burn in iterations. AUC and log-likelihood results are given in Table 2.1. Under both metrics, DRIFT achieves the best performance of the models considered. Receiver operating characteristic curves are shown in Figure 2.7. Table 2.2 shows the number of true positives for the k most likely edges of the missing entries predicted by each method, for several values of k. As some pairs of actors almost always have an edge between them in each timestep, the baseline method is very competitive for small k, but DRIFT becomes the clear winner as k increases.

We now look in more detail at the ability of DRIFT to model the dynamic aspects of the network. Figure 2.5 shows the predictive distributions for each of the methods, at times $t = 30$ and $t = 36$. At time $t = 30$, the network is dense, while at $t = 36$, the network has become sparse. While LFRM (all) and the baseline method have trouble predicting a sparse network at $t = 36$, DRIFT is able to scale back and predict a sparser structure, since

it takes into account the temporal sequence of the networks and it has learned that the network has started to sparsify before time $t = 36$. Figure 2.6 shows the edge probabilities over time for four pairs of actors. The pairs shown were hand picked "interesting" cases from the fifty most frequent pairs, although the performance on these pairs is fairly typical (with the exception of the bottom plot). The bottom plot shows a rare case where the model has arguably underfit, consistently predicting low edge probabilities for all timesteps.

Before concluding our discussion on DRIFT, we note that if a network is changing very slowly relative to the time scale of the observations, it can potentially be well modeled by simpler static methods such as LFRM (all) or the baseline method. In another scenario, it is possible that the network, and any underlying latent representation, change much more rapidly than the time scale of the observations, which would also thwart dynamic modeling. However, the DRIFT model can be very useful in situations like the Enron data set where the underlying communication patterns systematically and smoothly vary throughout the observation period. In these cases, a dynamic model is needed, and we have shown that DRIFT is up to the task.

2.7 Interpreting Network Models by Leveraging Text

As we have seen, the LFRM and DRIFT can be used to recover low-dimensional binary vector representations of the actors in a social network. For example, Alice may email Bob due to their unobserved mutual interest in salsa dancing. By learning such latent variable models from observed communication patterns, we may be able to infer that Alice and Bob have a feature in common. However it is not possible to infer from the network alone that this feature corresponds to a salsa dancing hobby in particular.

Fortunately, digital social networks such as email and social media are designed to facilitate communication between members of the network. As such, they often have additional information associated with them, namely the text of the actors' communications. This can include messages between the actors (e.g. emails), or broadcast-style communications on social media websites, such as tweets or facebook posts. We would like to be able to make use of this text information together with the network data we have previously been considering, in order to automatically discover not only a latent feature representation of each of the actors, but the *semantics* of these latent features. Thus we may be able to infer that Bob and Alice share a salsa dancing hobby.

There are a multitude of potential applications for such a system. In a social science context, recovering the semantics of latent features aids in the sociological interpretation of the models. For internet technology companies, inferring the interests of users in a social network can potentially help to target advertisements to those users. It also increases the power of the models for exploratory data analysis applications. For instance, the data set of emails used above arose from a lawsuit after the financial collapse of the Enron corporation. In such a setting, it would be very useful for the legal teams to be able to determine automatically which employees communicated with each other due to reasons relevant to the lawsuit in question, in order to focus their investigations into the data.

2.7.1 A Joint Model for Networks and Text

To discover latent features and their semantics, we introduce a framework for statistical models of communication networks where both network and text data are observed. The key idea is to model the network and the text jointly with a combination of the LFRM and LDA, associating each LFRM binary latent feature to an LDA topic. The network model is connected to the LDA topic model by allowing the latent features to affect the Dirichlet

prior on each document's distribution over topics. We can summarize the generative process of the model, which we refer to as LFRM_LDA, as

- Generate binary latent features $\mathbf{Z}$ and network $\mathbf{Y}$ via the LFRM
- Draw K topics $\mathbf{\Phi}$, where K is the number of latent features in $\mathbf{Z}$
- For each node document $\omega^{(i)}$
  - Select $\alpha^{(i)}$ as a function of $\mathbf{z}_i$
  - Draw $\omega^{(i)} \sim LDA(\mathbf{\Phi}, \alpha^{(i)})$
- For each edge document $\omega^{(ij)}$
  - Select $\alpha^{(ij)}$ as a function of $\mathbf{z}_i$, $\mathbf{z}_j$
  - Draw $\omega^{(ij)} \sim LDA(\mathbf{\Phi}, \alpha^{(ij)})$

The Dirichlet parameters are chosen such that the topics corresponding to the latent features belonging to the actors associated with the document get the most weight in the prior. We assume that all documents have the same total Dirichlet prior concentration α^+, and that a proportion γ of the prior weight comes from the topics of the latent features of the entities associated with the document, with the remaining weight coming from a flat distribution over all the topics. For documents $\omega^{(ij)}$ on edges, the γ proportion of the prior weight is divided beween i's features and j's features with proportion λ going to i's features. This leads to K-dimensional Dirichlet priors over the K topics with parameters

$$\alpha_k^{(ij)} = \alpha^+\Big(\frac{\gamma\lambda}{\sum_{k'} z_{ik'}} z_{ik} + \frac{\gamma(1-\lambda)}{\sum_{k'} z_{jk'}} z_{jk} + \frac{(1-\gamma)}{K}\Big) \tag{2.30}$$

$$\alpha_k^{(i)} = \alpha^+\Big(\frac{\gamma}{\sum_{k'} z_{ik'}} z_{ik} + \frac{(1-\gamma)}{K}\Big)\ . \tag{2.31}$$

This modeling strategy is reminiscent of the Dirichlet-multinomial regression (DMR) model of Mimno & McCallum (2008), except that the Dirichlet parameters are selected based on latent features instead of observed features, and a different regression parameterization is used. We can make use of an interpretation of the Dirichlet-multinomial distribution used in LDA, in order to give an intuition regarding equations 2.30 and 2.31. The Dirichlet-multinomial (cf. Minka (2000)) is the distribution that results from drawing a multinomial parameter vector θ from a Dirichlet prior with parameter vector α, and then drawing a count vector from a Multinomial(θ, N). LDA uses this distribution to select the number of words per topic in each document.

We can also interpret the Dirichlet-multinomial as a multivariate generalization of Polya's urn scheme. In this interpretation, we begin by placing α_k colored balls into an urn for each $k, 1 \leq k \leq K$, and with each k corresponding to a different color. Then, we randomly draw one of the balls in the urn, and record its color. We then place it back into the urn, along with a new ball of the same color. This process is repeated N times, outputting the number of balls of each color which were drawn. In our case, this corresponds to the counts of each topic. For LDA, α is usually real-valued, so we must extend our intuition to drawing from an urn with "partial" balls in it. In our model, α^+ corresponds to the total number of balls in the urn initially, and Equations 2.30 and 2.31 specify how these initial balls are distributed. Thus, in Equation 2.31, a proportion γ of the α^+ colored balls are initially placed in the urn, colored so that they are distributed evenly across the topics corresponding to the actor i's active features. The remaining proportion $1 - \gamma$ of the α^+ balls are distributed across all of the topics evenly. The per-topic counts are then generated using the urn process.

More details on the model, including an MCMC inference algorithm, are given in Appendix A. It should be noted that a closely related model to the one proposed here was developed independently to us by Zhang & Carin (2012). After this competing work was published, we did not explore the model further. Therefore, this model is evaluated less comprehensively

than for other parts of this thesis. Nevertheless, we demonstrate the use of the model for an exploratory data analysis (EDA) task.

2.7.2 Exploratory Data Analysis

A key property of the model is that it assigns semantically meaningful topics to the discovered latent features, aiding the use of the LFRM as an (EDA) tool. In this section we explore its use in this setting on the Enron email corpus and a data set of Twitter accounts belonging to emergency response organizations.

Enron Email Corpus

The Enron corpus contains 150 email folders belonging mainly to senior management of the Enron corporation. In each email, we removed all words past an occurrence of the word "forwarded" as a simple attempt to get only the new text of each message. We removed a list of 571 common stopwords[7], the most popular 100 baby names from the last century according to the United States Social Security Administration[8], and words that occurred only once. After merging multiple folders and email addresses associated with the same people and duplicating multi-recipient emails for each recipient, we find 140 distinct actors, with 1379 within-network edges associated with text, after aggregating emails extracted from the users' "sent_items" folders, and with a dictionary of 15150 distinct words. We model this data set using LFRM_LDA with a Poisson link function on the counts of the emails between actors,

$$y_{ij} \sim \text{Poisson}(\exp(\mathbf{z}_i \mathbf{W} \mathbf{Z}_j^\intercal)) \ . \tag{2.32}$$

[7]From http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop .
[8]From http://www.ssa.gov/oact/babynames/decades/century.html .

Topic	Ten Most Probable Words
Management	program company blackberry committee purpose continue created important remains talent
Fantasy Football	draft good back big time team make game day week
Fuel	gas deal storage cycle day capacity OFO daily socal volumes
Meetings	meeting call gas time make enron pm good discuss today
California Energy	power state california energy utilities electricity commission market rate prices

Table 2.3: Top words for several topics from the Enron data set. The topic names were chosen manually. "OFO" stands for "operational flow order".

We ran the MCMC algorithm on the Enron data set with 20 topics for 3000 iterations of burn-in, then recorded 100 samples, computed the probability of each z_{ik} based on the samples and reported the most likely assignment of each of the z_{ik}'s. The topics were fairly stable after burn-in, and we verified that feature swapping, which would confound this analysis, did not occur. Some examples (from the last sample) are shown in Table 2.3. The mean number of active features per person across the samples was 4.5.

Figure 2.8 shows a graphical representation of a subset of the recovered latent features, labeled according to their topics. This demonstrates how the BMF_LDA framework can be used to automatically extract semantically meaningful latent structure in a text-augmented network data set. One interesting feature corresponds to a topic on a fantasy football league that some members of Enron participated in. The model has automatically identified the actors that participate in the league. Perhaps unsurprisingly, the majority of the Enron employees (81/140) are associated with the "meetings" topic/feature.

Government Emergency Response Twitter Accounts

The Enron corpus, being an email data set, contains text on the edges of the graph. We also explored the use of the model on a network where the text is instead associated with the nodes, using a data set extracted from the social media website Twitter. The data set, due

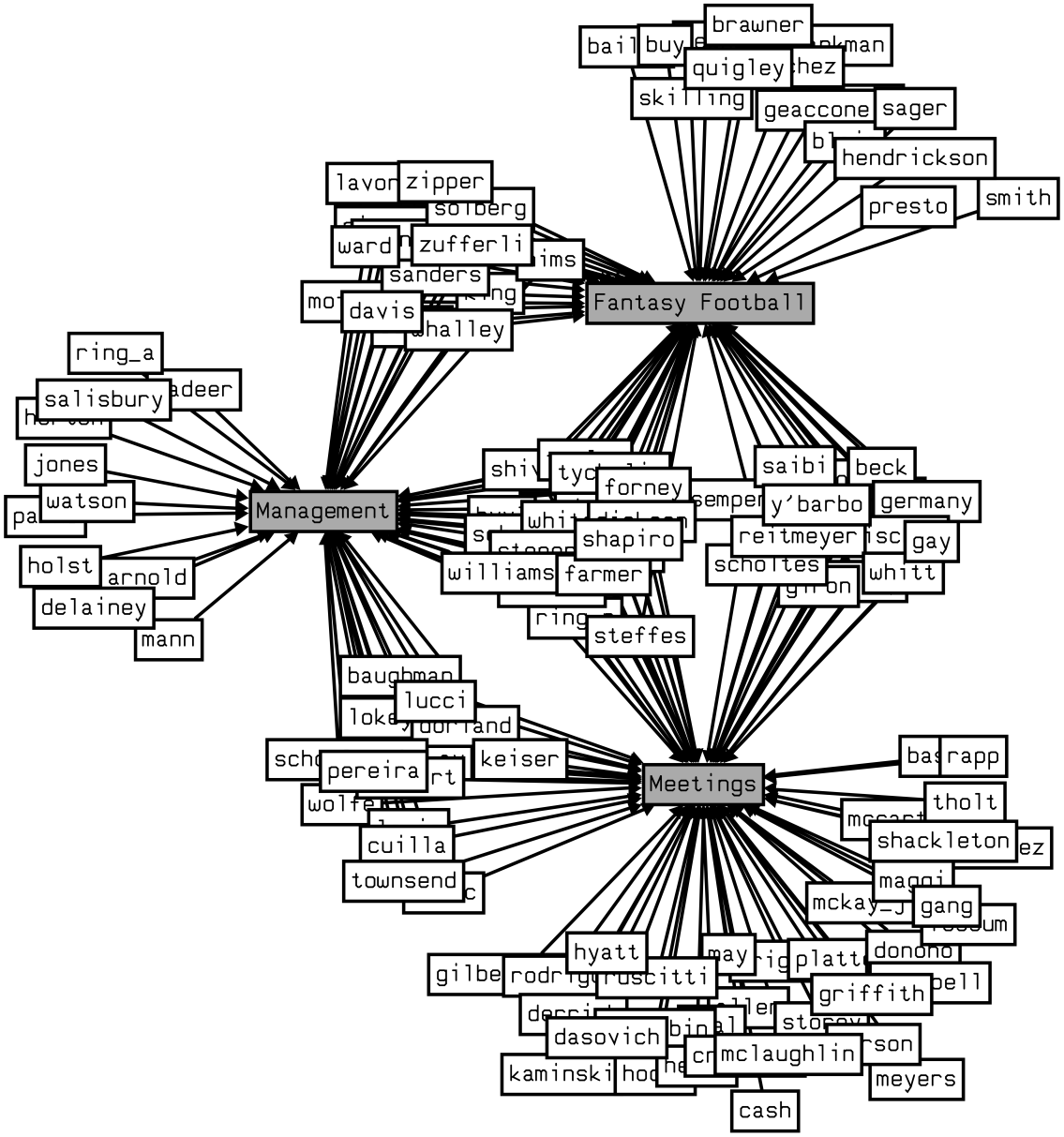


Figure 2.8: Bipartite graph of actors and a selected set of latent features that they possess. Arrows indicate that the feature is active for that actor in more than half of the recorded MCMC samples.

to Spiro *et al.* (2011), records the complete activity in the year 2011 of 216 Twitter accounts belonging to government organizations related to emergency response in the United States of America. The data include roughly 100,000 microblogging status updates ("*tweets*"), as well as the "*follows*" relation. On the website, if an account chooses to follow another account, it subscribes to the posts of that account. These Twitter accounts largely belong to organizations rather than to individuals and are designed primarily to disseminate information to the public, so it is unlikely that the edges in this graph exist for subscription purposes alone. Instead, it is more likely that the edges in the follows graph have been created to indicate affiliation or association. There are 2800 edges in the graph. The tweets were aggregated to create a single document for each account (i.e. each node in the graph), and the same text preprocessing was used as for the Enron corpus.

We modeled the network using LFRM_LDA with a logistic link function. The interaction matrix $\mathbf{W}$ was constrained to be diagonal, with an exponential prior distribution which further constrained it to have positive entries. This means that the semantics of the latent features correspond to communities of nodes which are more densely connected than the base rate of the network. The MCMC algorithm was performed for 3000 iterations of burn in with 100 samples subsequently recorded, using 20 topics.

Table 2.4 shows three manually selected topics, and the twitter accounts associated with the corresponding latent features. The emergency management topic was associated with a cluster of twitter accounts from the Federal Emergency Management Agency (FEMA), other regional EMA organizations, and other general emergency announcement organizations such as the emergency preparedness website "ready.gov."

The topic on storms was associated with a number of US Coastguard (USCG) accounts, three accounts belonging to the national oceanic and atmospheric organization (NOAA), NASA's hurricane webpage, and the US environmental protection agency (EPA). These are all accounts associated with organizations concerned with conditions in the ocean.

Topic	Most Probable Words	Twitter Accounts
Emergency Management	USA disaster FEMA gov weather emergency www preparedness recovery assistance	readydotgov fema craigatfema [*fema regional accounts 1 through 10*] femalro ntasalerts noradnorthcom cdcemergency alabamaema calema ...
Storms	tropical pacific full storm FBI NASA satellite NW man atlantic	usnoaagov uscoastguard noaacio nasahurricane usoceangov epagov uscgpacificnw uscgheartland icommandantuscg
Wildfires	fire Texas acres contained water Colorado firefighters south attorney acre	chpsouthern calema capublichealth fbipressoffice cagovernment jerrybrowngov cal_fire [*several state police departments*] [*several highway patrols*]

Table 2.4: Top words for several topics from the Twitter data set, and the Twitter accounts that the model associates with these topics.

A topic on wildfires was associated with the California Department of Forestry and Fire Protection (CAL FIRE), as well as many organizations relating to the state of California, state-level police departments, and highway patrol organizations. California is a state which is at risk of wildfires, so its association with the wildfire topic makes sense, even though this association is not a perfect match semantically. It should be noted that some topics were less coherent than the ones we have shown, and not all of the features corresponded to coherent communities of nodes. Nevertheless, we have seen that the model has been able to recover some useful semantically labeled clusters in an unsupervised way.

2.7.3 Related Work

A number of connections can be made between the LFRM_LDA model proposed here and other work in the literature. The idea of using an Indian buffet process (or the finite version of this, in our case) to select only a subset of the topics to be active for a given document, has been explored previously in the focused topic model (FTM) of Williamson *et al.* (2010). In the FTM, each document has a binary vector, a row of the IBP, specifying which topics

are present. The Dirichlet (or Dirichlet process, in the infinite case) prior weights are fixed for each topic, whereas in our case the weights, and the presence of the features, depend on the row and column entity's IBP feature vectors.

Other work has explored the idea of modeling matrix data with text associated with it. Balasubramanyan & Cohen (2011) model a network jointly with text documents in which the entities in the network are mentioned. Their Block-LDA model posits that a mixed membership stochastic blockmodel framework (Airoldi *et al.* , 2008) generates the network, and that the entity mentions are generated via LDA. The two components of the model are connected by using the same distributions over entities (i.e. topics) in each.

As previously mentioned, Zhang & Carin (2012) independently proposed a model closely related to the one proposed here. They build a joint model of networks and text using the binary matrix factorization (BMF) of Meeds *et al.* (2007) in conjunction with the focused topic model. BMF is the rectangular version of the LFRM, with feature matrices for both row and column entities. As in LFRM_LDA, they associate each topic with a binary latent feature. Each row of the matrix has a document associated with it, drawn using the focused topic model, with only the topics corresponding to the latent features being active in the document. The model follows the same high-level model structure as the one presented here, but differs in three ways: their model is designed for rectangular matrices only, the Dirichlet priors linking features to topics are specified differently, and they do not model documents associated with the edges of the network, but only with the nodes.

Wang & Blei (2011) presented collaborative topic regression (CTR), another related model which also uses a latent factorization model for the matrix data. In CTR, the latent representations of the row and column entities are real-valued instead of binary. Each feature once again is associated with a topic, and each column entity has an associated distribution over topics. The row entity's latent representation has a spherical multivariate Gaussian prior, while the column entity's representation's prior is it's topic distribution, plus multi-

variate Gaussian noise. The documents on each edge are drawn from the topic distribution of the column entity. This real-valued setup gives different semantics to the latent space – a real-valued embedding instead of an overlapping clustering. The preferred representation may depend on the task of interest. Note that CTR only handles documents associated with column entities, unlike our model which also handles documents on the row entities and on the edges, and also handles the square matrix case gracefully.

The author recipient topic model (ART) of McCallum *et al.* (2007) is also a topic model for documents on directed edges of a graph. In this model, each edge in the graph has its own distribution over topics, but with a flat prior, unlike BMF. If there are multiple recipients for an email, each word chooses a latent recipient from the list and the topic for that words is sampled from the resulting author-recipient pair. The entities themselves do not have latent representations. One can think of the text portion of LFRM_LDA as a modified version of ART, where the users themselves are modeled, allowing the latent representations of the users to influence the prior on the topic distribution of that edge.

Relational Topic Models (RTMs) (Chang & Blei, 2009) are models for networks with documents associated with nodes. Each node is generated via standard LDA, and links between documents are generated based on the similarity of their latent topic vectors. In other words, documents that are topically similar to each other are more likely to have an edge between them. Note that in this model, as in the other previous models discussed here, documents are generated before generating links, rather than generating documents conditional on the associated actors (such as senders and receivers) as in BMF_LDA.

The idea of learning the prior for each document's topic distribution from exogenous data has been explored before by Mimno & McCallum (2008). In their Dirichlet Multinomial Regression (DMR) model, LDA is extended such that the Dirichlet prior for the dth document's distribution over topics is parameterized by $\alpha_k^{(d)} = \exp(x^{(d)\intercal}\lambda^{(k)})$ for each topic k. Here, $x^{(d)}$ and $\lambda^{(k)}$ are feature vectors for the dth document and the kth topic, respectively, with the

$x^{(d)}$'s being observed and the latent $\lambda^{(k)}$'s being learned via optimization steps inside an MCMC sampler loop, similarly to our approach. Conditioned on the latent features, the text portion of BMF_LDA can be viewed as a variant of DMR using a different functional form for the α's.

2.8 Summary of Contributions

In this chapter, we have introduced a nonparametric Bayesian model for longitudinal social network data that models actors with latent features whose memberships change over time. We detailed an MCMC inference procedure that makes use of the IBP stick-breaking construction to adaptively select the number of features, as well as a forward-backward algorithm to sample the features for each actor at each time slice. Empirical results suggest that the proposed dynamic model can outperform static and baseline methods on both synthetic and real-world network data.

We also introduced a model for social network data with a text component, such as communication networks in online social media. The model leverages the text to aid in the interpretation of the latent features. By fitting the model to data using an MCMC algorithm, we showed how this method can be used for exploratory data analysis.

The primary contributions of this chapter are

- We proposed DRIFT, a nonparametric Bayesian latent variable model for social networks over time.
- An MCMC algorithm was proposed to fit the model to data. The MCMC algorithm uses several sophisticated techniques which have previously been shown to improve mixing properties over naive sampling algorithms.

- Unlike the straightforward MCMC algorithm proposed by Miller *et al.* (2009), the algorithm uses a slice sampling technique which exploits the stick-breaking construction of the IBP, potentially leading to better mixing properties (Teh *et al.* , 2007b). For instance, Miller et al.'s method needs a complex initialization scheme, unlike our method. Since the LFRM is a special case of DRIFT, this method can also be used for the LFRM.
- For the sequential aspects of the model, the algorithm also uses the forward-backward block Gibbs sampler instead of the simple direct Gibbs method. This technique is also known to improve mixing (Scott, 2002).

- We evaluated the model extensively, both qualitatively and quantitatively, on synthetic and real data, and on both forecasting and missing data tasks, showing an improvement over baseline approaches. We acknowledge and thank Christopher DuBois and Arthur Asuncion for assistance with running experiments. Dr. DuBois also implemented the baseline algorithm, and Dr. Asuncion improved the performance of the code. We also thank Dr. Asuncion, Dr. DuBois and Carter Butts for helpful discussions.

- We also proposed LFRM_LDA, which jointly models networks and text associated with them such as communications within the network.

- We derived an MCMC algorithm to fit LFRM_LDA (described in Appendix A).

- Finally, we demonstrated the application of the model by using it for exploratory data analysis on email and Twitter data sets. We are grateful to Spiro *et al.* (2011) for generously providing the Twitter data.

Chapter 3

Topic Models for Exploring Scientific Influence in Citation Networks

> Thou weigh'st thy words before thou givest them breath.
>
> *William Shakespeare, Othello*

In the latter part of the previous chapter we investigated data where network and text information occur together. The focus was on communication data, such as email networks and digital social media. This chapter continues the theme of modeling networks and text simultaneously, but in a different application domain. Here, we consider instead the analysis of corpora of scientific articles.

The key elements of the data in this domain are the text of the manuscripts, and the network of the citation relationships between the articles. These elements are frequently studied separately, but as we saw in the previous chapter, there is great potential for these two aspects of the data to be used in conjunction with each other to gain a better understanding of a data set.

Statistical latent variable models provide a flexible and extensible framework for analyzing networks and text data, both separately and together. In this chapter, we introduce a new latent variable model for leveraging these two aspects of scientific literature, called *topical influence regression* (TIR). The model is designed specifically to help answer the questions that arise uniquely in this domain. Specifically, we would like to be able to compute data driven assessments of scientific impact. We would also like to discover the nature of the relationships between scientific articles, including the spread of ideas along the citation graph and the extent of the influence that a cited article has on an article that cites it. Techniques that can answer these kinds of questions open the door to the development of exploratory analysis tools to help scientists quickly get their bearings in fields of study other than their own.

To this end, we develop the topical influence regression model, which posits that articles "coerce" the articles that cite them into having similar topical content to them. This is modeled in an unsupervised way, using a latent Dirichlet allocation framework (Blei *et al.* , 2003) and introducing latent variables which encode this influence. These latent variables represent a new bibliometric measure called *topical influence*, which we define in the context of our model. In the TIR model, articles with higher topical influence have a larger effect on the topics of the articles that cite them. We model this influence mechanism via a regression on the parameters of the Dirichlet prior over LDA topics.

This chapter shows how such models can be used to recover meaningful influence scores, both for articles and for specific citations. By looking not just at the citation graph but also taking into account the content of the articles, topical influence regression can provide a more complete picture of scientific impact than the simple citation-based scores used in traditional bibliometrics. A published version of the work in this chapter is available in Foulds & Smyth (2013).

The remainder of the chapter is structured in the following way. Section 3.1 motivates the problem of automatically inferring scientific impact, and Section 3.2 discusses background information on work related to this problem. In Section 3.3, we introduce the topical influence regression model, and we describe an inference algorithm for the model in Section 3.4. An experimental analysis of the model is performed in Section 3.5, and we summarize the contributions of this chapter in Section 3.6.

3.1 Motivation

Scientific articles are not created equal. Some articles generate entire disciplines or subdisciplines of research, or revolutionize how we think about a problem, while others contribute relatively little. When we are first introduced to a new area of scientific study, we may not be informed as to which are the most influential articles that we should focus our attention on, or the history of how ideas were built upon each other. In this situation, it would be useful to have tools which can automatically find the most important articles, and the relationships of influence between articles. Understanding the impact of scientific work is also crucial for hiring decisions, allocation of funding, university rankings and other tasks that involve the assessment of scientific merit. If scientific works stand on the shoulders of giants, we would like to be able to find the giants.

The most straightforward method for quantitatively assessing importance of a scientific work is to simply count the number of times that it has been cited. However, citation counts are not the whole story. The number of citations an article receives provides one indication of importance, but this is confounded by the unknown function of each citation. Many citations are made in passing, are relevant to only one section of an article, or make no impact on a work but are referenced out of "politeness, policy or piety" (Ziman, 1968).

In reality, scientific impact has many dimensions. Some articles are important because they describe scientific discoveries that alter our understanding of the world, while some develop essential tools and techniques which facilitate future research. Other articles are influential because they introduce the seeds of new ideas, which in turn inspire many other articles. We would like to take more of these dimensions into account when assessing scientific merit. This motivates more in-depth bibliometric techniques, such as the model-based approach which we take in this chapter, leveraging the *content* of the articles as well as the citation graph. To put our method in context, we first overview relevant bibliometric techniques and related work in the literature.

3.2 Bibliometrics and Related Work

We can categorize bibliometric techniques into those which are simply based on citation counts, those which exploit the citation graph, and machine learning approaches which frequently leverage additional data such as text. We overview the methods in each of these categories in turn.

3.2.1 Metrics Derived from Citation Counts

Going one step beyond simply counting citations to measure impact, it is also possible to computed other derived bibliometric measures based on these counts. For example, the *impact factor* of a publication venue for a given year is defined to be the average number of times articles from that venue, published in the previous two years, were cited in that year. However, the quality of articles in a given publication venue can vary wildly, and it is difficult to compare impact factors between different disciplines of study.

Another metric based on citation counts is the h-index of a scientist (Hirsch, 2005). A scholar is said to have "index h" if she has h papers with at least h citations, and this is not the case for any larger value of h. This metric is designed to measure both the "quality" (in terms of number of citations per paper) and quantity of the research of a given scholar. Other related metrics are the i10-index, attributed to Google,[1] which counts the number of publications with at least ten citations, and the g-index (Egghe, 2006), which is the largest number of publications such that the top g of them together receive at least g^2 citations. These metrics do not consider context such as the number of authors or their ordering in the author list, the publication venue, or the function of the citations. Another limitation of these metrics is that they apply at the level of the scholar, and not the individual publication (although they can readily be applied to other publication-producing entities such as institutions and journals).

3.2.2 Graph-Based Approaches

A more sophisticated approach than citation counting is to consider the structure of the citation graph, instead of just the raw counts of the citations. For example, measures of importance can be derived recursively from the citation graph, such as PageRank (Brin & Page, 1998). Such graph-based measures do not in general make use of the textual content of the articles, although it is possible to apply them to graphs where the edges between articles are determined based on the similarity of their content instead of the citation graph (Lin, 2008). In an information retrieval context, Hypertext-Induced Topic Search (HITS) (Kleinberg, 1999) and topic-sensitive PageRank (Haveliwala, 2002) use textual content to bias these recursive measures in a query-dependent way. These techniques output answers to information

[1] Cf. `www.googlescholar.blogspot.com/2011/11/google-scholar-citations-open-to-all.html`.

retrieval queries for search engines, which are not directly relevant to the analysis of scientific literature.

3.2.3 Machine Learning Approaches

Machine learning methods for the study of scientific literature take a data-driven approach, and thus tend to make use of other available sources of information such as text. These approaches can broadly be divided into supervised methods, which require labeled information regarding scientific impact, and unsupervised methods, which proceed without the use of such labels.

In the supervised category, Teufel *et al.* (2006) use supervised classification algorithms to predict the function of each citation. More closely related to the method presented in this chapter, and published slightly after this work, Zhu *et al.* (2014) predict the presence or absence of influence relationships along each citation edge, using support vector machines and logistic regression. Similarly to the work in this chapter, they use the results of the algorithms to adjust citation count-based metrics to take into account influence relationships. The overall goals of Zhu *et al.* are very similar to what is presented here. One key difference is that our approach uses unsupervised techniques instead of supervised methods, so it does not require expensive labeled data.

A variety of unsupervised methods using LDA-style probabilistic models have also been proposed to analyze both textual content and citation links. An early example is the work of Cohn & Hofmann (2001), which combines PLSA and PHITS to model the connections between words and citations. The passage impact model of Shaparenko & Joachims (2009) uses a mixture modeling strategy to attempt to identify which parts of scientific articles are novel, and which are "copied" from earlier work.

Chang & Blei (2009)'s relational topic model (RTM) is another related model. The RTM models the joint probability of citation links and document content, again via an extension to LDA. The text of each document is assumed to be generated using standard LDA and the links (e.g., citations) are then generated subsequently based on the text. Our goal and approach is different in that we wish to make direct inferences about the influence of cited articles on citing articles, rather than model the probability of citation links. Thus, we model the conditional distribution of latent topics and latent influences, conditioned on an observed citation graph, rather than a joint model of topics and links.

Another topic model for scientific corpora is TopicFlow (Nallapati *et al.* , 2011), a PLSA-based model for the flow of topics in a document network. In their model, citing articles "vote" on each cited article's topic distribution in retrospect, via a network flow model. Since this voting occurs in time-reversed order, it does not describe an influence mechanism and is not a generative model that can simulate or predict new documents.

The document influence model of Gerrish & Blei (2010) can be viewed as orthogonal to this work, in that it models the impact of documents on *topics* over time (specifically, how topics change over time) rather than how articles influence the specific *articles* that cite them.

The previous method which is perhaps the closest in goals and methodology to the present work is that of Dietz *et al.* (2007), who introduce the citation influence model (CIM). Building on the latent Dirichlet allocation (LDA) framework, CIM assumes that each word is drawn by first selecting either (a) the distribution over topics of a cited article (with probability proportional to the influence weight of that article on the present article) or (b) a novel topic distribution, and drawing a topic from the selected distribution, then finally drawing the word from the chosen topic.[2] In their approach, every word is assigned an extra latent variable, namely the cited article whose topic distribution the topic was drawn from. For the

[2]A model similar to Dietz et al.'s CIM was also proposed later by He *et al.* (2009), in the context of detecting the evolution of topics over time.

model proposed in this paper, we do not need to introduce these additional latent variables, which leads to a simpler latent representation and fewer variables to sample during inference. Dietz *et al.* (2007) also assume that the citation graph is bipartite, consisting of one set of citing articles and one set of cited articles—in contrast, our proposed models can handle arbitrary citation graphs in the form of directed acyclic graphs (DAGs). While both the CIM and our approach can identify the influence of specific citations between articles, our model can also infer how influential each article is overall, and provides a flexible modeling framework which can handle different assumptions about influence.

A more general topic model called Dirichlet-multinomial regression (DMR), due to (Mimno & McCallum, 2008), provides a framework for building topic models conditioned on arbitrary features. This model can also learn latent topics conditioned on citations. However, unlike our method it does not model influence directly, does not make use of the content of cited articles in the regression and does not model a full network of probabilistic dependence relationships between articles. Nevertheless, in this chapter we build upon the ideas of DMR to create models which have these properties.

3.3 Topical Influence Regression

Scientific research is seldom performed in a vacuum. New research builds on the research that came before it. Although there are many aspects by which the importance of a scientific article can be judged, in this work we are interested in the extent to which a given article has or will have subsequent articles that build upon it or are otherwise inspired by its ideas. We begin by defining *topical influence*, a quantitative measure for this type of influence.

3.3.1 Topical Influence

It is not immediately obvious how one might quantify such a notion of "idea-based" influence. However, the mechanism used in the scientific community for giving credit to prior work is citation. The presence of a citation from article b to article a therefore indicates that article b may have been influenced by the ideas in article a, to some unknown extent. We hypothesize that the extent of this influence manifests itself in the language of b. Using latent Dirichlet allocation (LDA) topics as a concrete proxy for the vague notion of "ideas", we define the *topical influence* of a to be the extent to which article a coerces the documents which cite it to have similar topic distributions to it. Topical influence will be made precise in the context of a generative model for scientific corpora, conditioned on the citation graph, called *topical influence regression* (TIR).

The proposed model extends the LDA framework of Blei *et al.* (2003) in order to model topical influence (see Section 1.5). In LDA, each word $w_i^{(d)}$ of each document d is assigned to one of K latent topics, $z_i^{(d)}$. Each topic $\Phi^{(k)}$ is a discrete distribution over words. Document d has a distribution over topics $\theta^{(d)}$, which can be viewed as a "location in topic space" summarizing its thematic content. The $\theta^{(d)}$'s have a Dirichlet prior distribution with parameters $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_K]^\intercal$. Although the α_k's are often set to be equal, representing a relatively uninformative prior over the θ's, a unique $\alpha^{(d)}$ for each document can also be used to encode prior information such as the effect of other variables on the topics of that document (Mimno & McCallum, 2008). In our case, we want to model the influence that a document has on the topic distributions of the documents that cite it. A natural way to encode such influence, then, is to allow documents to affect the value of $\alpha^{(d)}$ for each document d that cites them.

Accordingly, we model each article d as having a latent, non-negative "topical influence" value $l^{(d)}$. Let $n^{(d)}$ be number of words in article d, $n_k^{(d)}$ be the number of words assigned to

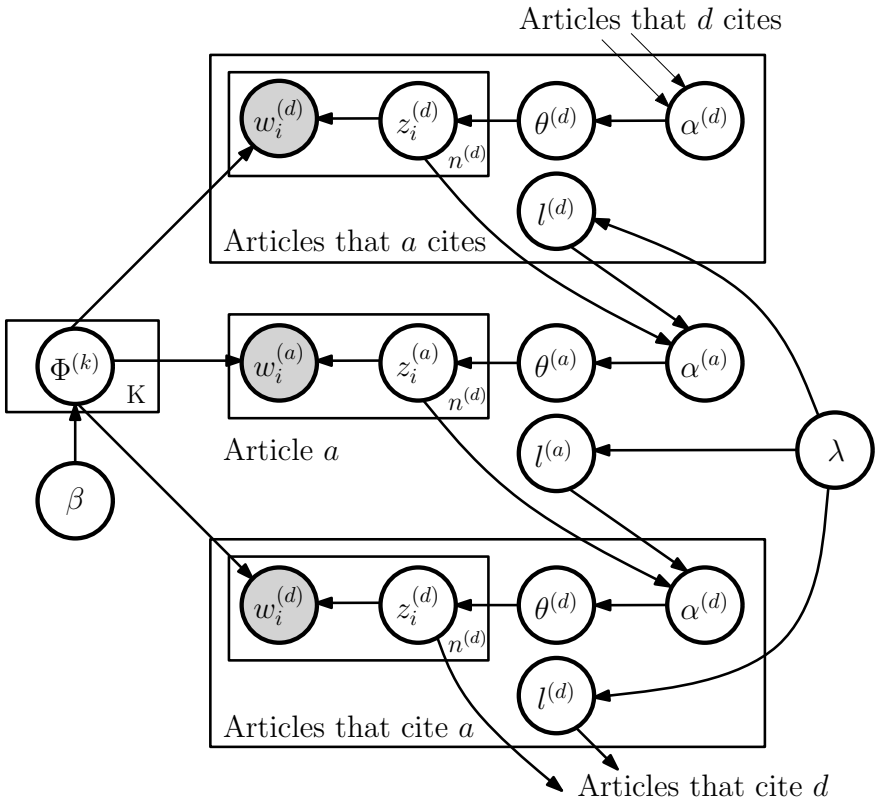


Figure 3.1: The graphical model for the portion of the TIR model connected to article a (the links from the $\mathbf{z}$'s and l's to the $\alpha^{(d)}$'s are deterministic). The full model applies this diagram recursively.

topic k, and let $C^{(d)}$ be the set of articles that d cites. We model $\alpha^{(d)}$ as

$$\alpha^{(d)} = \sum_{c \in C^{(d)}} l^{(c)} \bar{\mathbf{z}}^{(c)} + \alpha \ , \tag{3.1}$$

where $\bar{\mathbf{z}}^{(c)} = \frac{1}{n^{(c)}}[n_1^{(c)}, \ldots, n_K^{(c)}]^\intercal$ is the normalized histogram of topic counts for document c, and α is a constant for smoothing. Since the $\bar{\mathbf{z}}^{(c)}$'s sum to one, the topical influence $l^{(c)}$ of article c can be interpreted as the number of words of precision that it adds to the prior of the topic distributions of each document that cites it. As we increase $l^{(c)}$, the articles that cite c become more likely to have similar topic proportions to it. Thus, $l^{(c)}$ encodes the degree to which article c influences the topics of each of the articles that cite it.

3.3.2 Polya Urn Interpretation of Topical Influence

From another perspective, marginalizing out $\theta^{(d)}$, we can view the topic counts (in the standard LDA model) for document d as being drawn from a Polya urn scheme with $\alpha_k^{(d)}$ (possibly fractional) balls of each color $k \in \{1, \ldots, K\}$ initially in the urn. For each word, a ball is drawn randomly from the urn and the topic assignment is determined according to its color k. The ball is replaced in the urn, along with a new ball of color k. In our model, for each article c cited by article d we place $l^{(c)}$ balls, with colors distributed according to $\bar{\mathbf{z}}^{(c)}$, into article d's urn initially. Thus, article d's topic assignments are more likely to be similar to those of the more influential articles that it cites. The total number of balls that d added to other articles' urns,

$$T^{(d)} \triangleq \sum_{b:d \in C^{(b)}} l^{(d)} = l^{(d)} \left| \{b : d \in C^{(b)}\} \right| \tag{3.2}$$

measures the total impact (in a topical sense) of the article. We refer to this as *total topical influence*.

3.3.3 Generative Model

The full assumed generative process for articles in this model begins with a directed acyclic citation graph $G = \{V, E\}$. Intuitively, citation graphs are typically DAGs because articles can normally only cite articles that precede them in time. We assume that G is a DAG so that influence relationships are consistent with some temporal ordering of the articles, and so that the resulting model is a Bayesian network. Here, each vertex v_i corresponds to an article d_i, edge $e = (v_1, v_2) \in E$ IFF d_1 is cited by d_2, and vertices (articles) are numbered in a topological ordering with respect to G. Such an ordering exists because G is a DAG. We model each article d's word vector $w^{(d)}$ as being generated in topological

sequence, similarly to LDA but with its prior over topic distribution being Dirichlet($\alpha^{(d)}$), as given by Equation 3.1. Note that each $\alpha^{(d)}$ is a function of the topics of the documents that it cites, parameterized by their topical influence values. We therefore call this model *topical influence regression* (TIR). The graphical model for TIR is given in Figure 3.1, and the generative process for TIRE is given in Algorithm 4.

Algorithm 4 Generative process for TIR

- For each topic k
  - $\Phi^{(k)} \sim \text{Dirichlet}(\beta)$ *//Sample a topic*
- For each document d, in topological order
  - $l^{(d)} \sim \text{exponential}(\lambda)$ *//Sample an influence weight*
  - $\alpha^{(d)} = \sum_{c \in C^{(d)}} l^{(c)} \bar{\mathbf{z}}^{(c)} + \alpha$ *//Assign a prior over topics*
  - $\theta^{(d)} \sim \text{Dirichlet}(\alpha^{(d)})$ *//Sample a distribution over topics*
  - For each word i in document d
    - $z_i^{(d)} \sim \text{discrete}(\theta^{(d)})$ *//Sample a topic*
    - $w_i^{(d)} \sim \text{discrete}(\Phi^{(z_i^{(d)})})$ *//Sample a word*

3.3.4 Modeling Influence Along Citation Edges

The TIR model provides us with topical influence scores for each article, but it does not tell us about topical influence relationships between specific pairs of cited and citing articles. To model such relationships, we can consider a hierarchical extension to TIR, with edge-wise topical influences $l^{(c,d)}$ for each edge (c, d) of the citation graph, $l^{(c,d)} \sim \text{truncGaussian}(l^{(c)}, \sigma, l^{(c,d)} \geq 0)$.[3] In this case,

$$\alpha^{(d)} = \sum_{c \in C^{(d)}} l^{(c,d)} \bar{\mathbf{z}}^{(c)} + \alpha \ . \tag{3.3}$$

[3]We use a truncated Gaussian, rather than, say, a gamma distribution or a log-normal, because we desire a roughly bell-shaped distribution which is parameterized by a measure of central tendency. If the standard deviation is small relative to the un-truncated mean, the mean of the distribution will be close to the un-truncated mean.

This hierarchical setup allows us to continue to infer article-level topical influences, and provides a mechanism for sharing statistical strength between influences associated with one cited article. We shall refer to the model with influences on just the nodes (articles) as *TIR*, and the hierarchical extension with influences on the edges as *TIRE*. The TIRE model is detailed in Algorithm 5.

Algorithm 5 Generative process for TIRE

- For each topic k
  - $\Phi^{(k)} \sim \text{Dirichlet}(\beta)$ *//Sample a topic*
- For each document d, in topological order
  - $l^{(d)} \sim \text{exponential}(\lambda)$ *//Sample a node influence weight*
  - For each cited document $c \in C^{(d)}$
    - $l^{(c,d)} \sim \text{truncGauss}(l^{(c)}, \sigma, l^{(c,d)} \geq 0)$ *//Draw an edge influence weight*
  - $\alpha^{(d)} = \sum_{c \in C^{(d)}} l^{(c,d)} \bar{\mathbf{z}}^{(c)} + \alpha$ *//Assign a prior over topics*
  - $\theta^{(d)} \sim \text{Dirichlet}(\alpha^{(d)})$ *//Sample a distribution over topics*
  - For each word i in document d
    - $z_i^{(d)} \sim \text{discrete}(\theta^{(d)})$ *//Sample a topic*
    - $w_i^{(d)} \sim \text{discrete}(\Phi^{(z_i^{(d)})})$ *//Sample a word*

3.3.5 Relationship to Dirichlet-Multinomial Regression

The TIR model can be viewed as an adaption of the Dirichlet-multinomial regression (DMR) framework of Mimno & McCallum (2008) to model topical influence. DMR also endows each document with its own unique $\alpha^{(d)}$, but with $\alpha_k^{(d)} = \exp(\mathbf{x}^{(d)\intercal} \lambda^{(k)})$ being a function of the observed feature vector $\mathbf{x}^{(d)}$ parameterized by regression coefficients λ. The DMR model can also be applied to text corpora with citation information, by setting the feature vectors to be binary indicators of the presence of a citation to each article. TIR differs in that the functional form of the regression is parameterized in a way that directly models influence, and also differs in that the regression takes advantage of the content of the cited articles via their topic assignments.

Because an article's prior over topic distributions depends on the topic assignments of the articles that it cites, TIR induces a network of dependencies between the topic assignments of the documents. Specifically, if we collapse out Θ, the dependencies between the $\mathbf{z}$'s of each document form a Bayesian network whose graph is the citation graph. In contrast, DMR treats the documents as conditionally independent given their citations, and does not exploit their content in the regression.

To illustrate this, Figure 3.2 shows an example citation graph and the resulting Bayesian network. In the figure, an edge in *(a)* from c to d corresponds to a citation of c by d. Conditioned on the topics, the dependence relationships between $\mathbf{z}$ nodes in *(b)* follow the same structure as the citation graph.

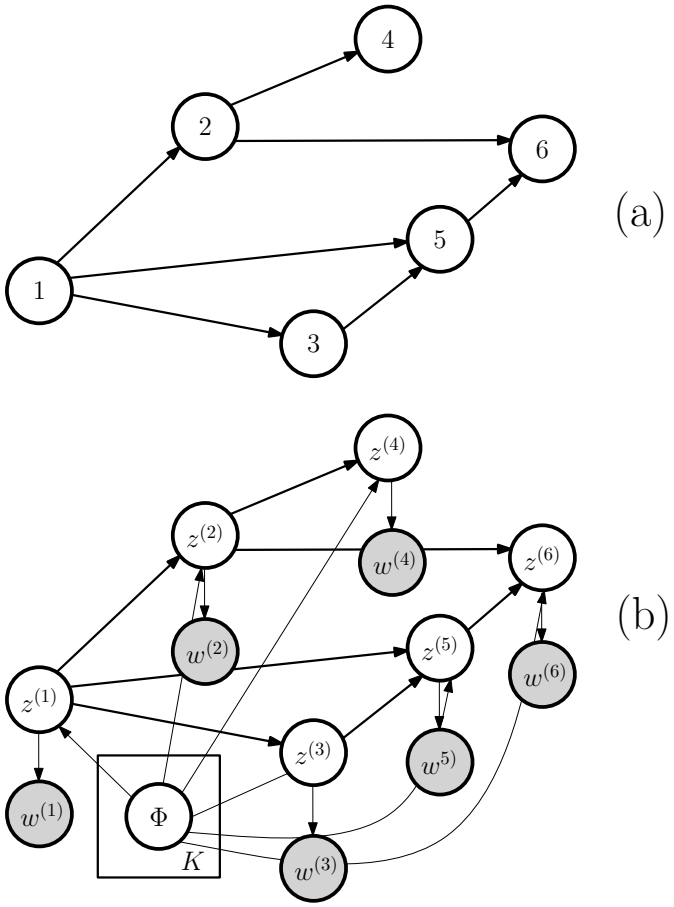


Figure 3.2: *(a)* An example citation network. *(b)* Graphical model for TIR on the example network, collapsing out Θ but retaining topics Φ. Influence variables and hyper-parameters not shown for simplicity.

3.4 Inference

We perform inference using a Markov chain Monte Carlo technique. We use a collapsed Gibbs sampling approach analogous to Griffiths & Steyvers (2004), integrating out $\mathbf{\Theta}$ and $\mathbf{\Phi}$. The update equation for the topic assignments is

$$Pr(z_i^{(d)} = k | z^{-(d,i)}, \ldots) \propto (n_k^{(d)-(d,i)} + \alpha_k^{(d)}) \frac{n_k^{(w_i^{(d)})-(d,i)} + \beta_{w_i^{(d)}}}{n_k^{-(d,i)} + \sum_w \beta_w} \times \prod_{d':d \in C(d')} \text{Polya}(\mathbf{z}^{(d')} | \alpha^{(d')} : z_i^{(d)} = k,\, \mathbf{z}^{-(d,i)},\, l) \tag{3.4}$$

where the n_k's are the counts of the occurrences of topic k over all of the entries determined by the superscript. The $-(d,i)$ superscript indicates excluding the current assignment for $z_i^{(d)}$. The update equation is similar to the update equations of Griffiths and Steyvers, but with a different α for each document d, and with multiplicative weights for each document that cites it. These weights $\text{Polya}(\mathbf{z}^{(d)} | \alpha^{(d)})$ are the likelihood for a multivariate Polya (a.k.a. Dirichlet-multinomial) distribution,

$$\text{Polya}(\mathbf{z}^{(d)} | \alpha^{(d)}) = \frac{\Gamma(\sum_k \alpha_k^{(d)})}{\Gamma(n^{(d)} + \sum_k \alpha_k^{(d)})} \prod_k \frac{\Gamma(n_k^{(d)} + \alpha_k^{(d)})}{\Gamma(\alpha_k^{(d)})} \; .$$

In the case of TIR, in the collapsed model the full conditional posterior for the topical influence values l is

$$Pr(l | \mathbf{z}, \lambda) \propto Pr(\mathbf{z} | l) Pr(l | \lambda) \; . \tag{3.5}$$

Here, we have that

$$Pr(\mathbf{z} | l) = \prod_{d=1}^{D} \text{Polya}(\mathbf{z}^{(d)} | l^{C^{(d)}}, \mathbf{z}^{C^{(d)}}) \; . \tag{3.6}$$

The topical influence values l can be sampled using Metropolis-Hastings updates, or slice sampling. An alternative is to perform stochastic EM, optimizing the likelihood or the posterior probability of l, interleaved within the Gibbs sampler, as in Mimno & McCallum (2008) and Wallach (2006). In experiments on synthetic data we found that maximum likelihood updates on l, obtained via stochastic EM using gradient ascent, resulted in the lowest L1 error from the true l, so we use this strategy for the experimental results in this chapter. The derivative of the log-likelihood with respect to the topical influence $l^{(a)}$ of article a is

$$\frac{\mathrm{d}Pr(z|l)}{\mathrm{d}l^{(a)}} = \sum_{d:a\in C(d)} \Big(\Psi(\sum_k \sum_{c\in C^{(d)}} l^{(c)}\bar{\mathbf{z}}_k^{(c)} + K\alpha) - \Psi(\sum_k \sum_{c\in C^{(d)}} l^{(c)}\bar{\mathbf{z}}_k^{(c)} + K\alpha + n^{(d)})\Big) + \sum_{d:a\in C(d)} \sum_{k=1}^{K} \bar{\mathbf{z}}_k^{(a)} \Big(\Psi(\sum_{c\in C^{(d)}} l^{(c)}\bar{\mathbf{z}}_k^{(c)} + \alpha + n_k^{(d)}) - \Psi(\sum_{c\in C^{(d)}} l^{(c)}\bar{\mathbf{z}}_k^{(c)} + \alpha)\Big) ,$$

where $\Psi(.)$ is the digamma function. For TIRE, the likelihood decomposes across documents and we can optimize the incoming edge weights for each document separately. We have

$$\frac{\mathrm{d}Pr(z^{(d)}|l)}{\mathrm{d}l^{(a,d)}} = \Psi(\sum_k \sum_{c\in C^{(d)}} l^{(c,d)}\bar{\mathbf{z}}_k^{(c)} + K\alpha) - \Psi(\sum_k \sum_{c\in C^{(d)}} l^{(c,d)}\bar{\mathbf{z}}_k^{(c)} + K\alpha + n^{(d)}) + \sum_{k=1}^{K} \bar{\mathbf{z}}_k^{(a)} \Big(\Psi(\sum_{c\in C^{(d)}} l^{(c,d)}\bar{\mathbf{z}}_k^{(c)} + \alpha + n_k^{(d)}) - \Psi(\sum_{c\in C^{(d)}} l^{(c,d)}\bar{\mathbf{z}}_k^{(c)} + \alpha)\Big) .$$

We optimize the node-level l's in TIRE via the least squares estimate (LSE),

$$\hat{l}^{(a)} = \frac{1}{|\{d : a \in C^{(d)}\}|} \sum_{d:a\in C^{(d)}} l^{(a,d)} . \tag{3.7}$$

Although the LSE for the mean of a truncated Gaussian is biased, it is widely used as it is more robust than the MLE (A'Hearn, 2004).

3.5 Experimental Analysis

In this section we experimentally investigate the properties of TIR and TIRE. We consider two scientific corpora: a collection of 3286 of articles from the Association for Computational Linguistics (ACL) conference[4] (Radev *et al.* , 2013) published between 1987 and 2011, and a corpus of articles from the Neural Information Processing Systems (NIPS) conference[5] containing 1740 articles from 1987 to 1999. The corpora both contained a small number (53, and 14, respectively) of citation graph loops due to insider knowledge of simultaneous publications. Some loops were removed by manual deletion of "insider knowledge" edges, and others were removed by deleting edges in the loop uniformly at random. For computational efficiency, we performed approximate Gibbs updates where we drop the multiplicative Polya likelihood terms in Equation 3.4. This corresponds to only transmitting influence information downward in the citation DAG, but not transmitting "reverse influence" information upwards. Preliminary experiments on synthetic data indicated that this did not significantly impact the ability of the model to recover the topical influence weights. As one might expect, LDA is already capable of inferring topic distributions which are good enough to perform the regression on, without fully exploiting the additional feedback from the regression. This algorithm has a similar running time to the standard collapsed Gibbs sampler for LDA, as the regression step is not a bottleneck.

In all experiments, we set the hyper-parameters to $\alpha = 0.1, \beta = 0.1$ and the σ parameter for the truncated Gaussian in TIRE to be 1. We interleaved regression steps every 10 Gibbs iterations. For exploratory data analysis experiments the models were trained for 500 burn-in iterations, and the samples from the final iterations were used for the analysis.

[4] `http://clair.eecs.umich.edu/aan/`

[5] `http://www.arbylon.net/resources.html`, published by Gregor Heinrich and based on an earlier collection due to Sam Roweis.

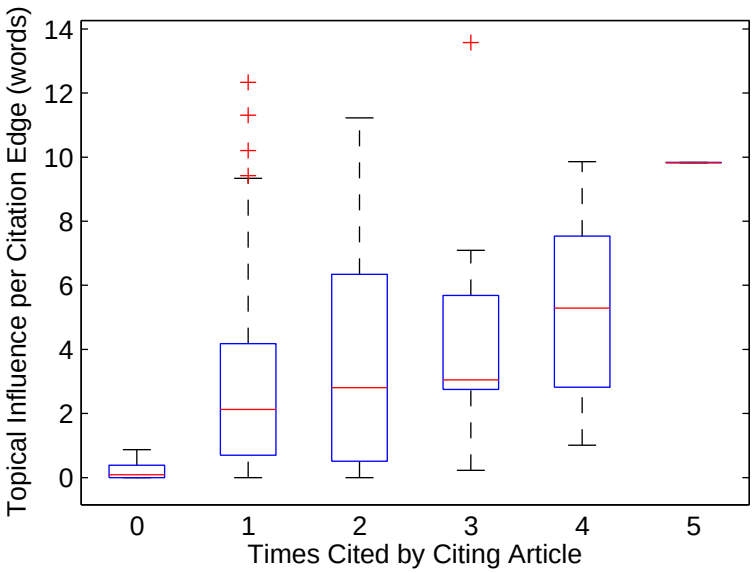


Figure 3.3: Topical influence per edge versus number of times cited by the citing article (NIPS). Several articles had zero in-text citations due to author or dataset errors.

3.5.1 Model Validation using Metadata

It is not immediately obvious how to best validate an unsupervised model of citation influence. Ground truth is not well-defined and human evaluation requires extensive knowledge of the individual papers in the corpora. Also note that citation counts are not useful for validating topical influence, as they measure the *number of articles* that cite a given article, and not the impact that the article has on the *articles that do cite it*. With this in mind, we explore how topical influence scores relate to several kinds of document metadata, which serve as a proxy for ground truth.

In many cases, if article c is repeatedly cited in the text of article d it may indicate that d builds heavily on c. We would therefore expect to see an association between the number of references to article c within the text of article d and edge-wise topical influence $l^{(c,d)}$. For each of the 106 papers in the NIPS corpus with at least three distinct references, we manually counted the number of repeated citations for the most influential and least influential

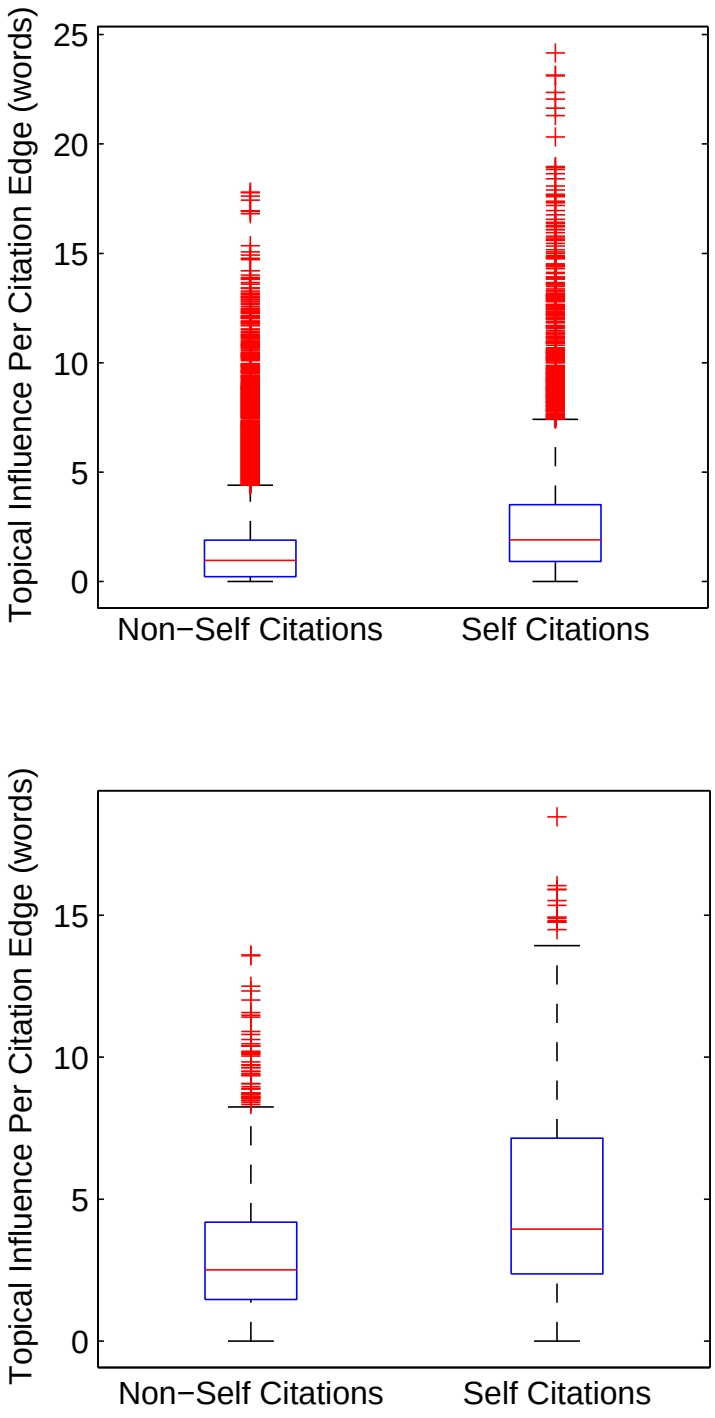


Figure 3.4: Topical influence for self and non-self citation edges. Top: ACL. Bottom: NIPS.

references according to the TIRE model. Box plots for the edge-wise topical influence scores, computed over the citation edges containing a given number of repeated references, is given in Figure 3.3. The figure shows that edge-wise topical influence tends to increase with the number of in-text citations by the citing article.

Overall, the "most influential" references were cited 171 times in the text of their citing articles, while the "least influential" references were cited 128 times. Of the 45 articles where the counts were not tied, the most influential references had the higher citation counts 33 times. A sign test rejects the null hypothesis that the median difference in citation counts between least and most influential references is zero at $\alpha = 0.05$, with p-value $\approx 5 \times 10^{-4}$. Note that repeated citations only occurred for 30 percent of the edges considered, and so they do not in general provide a precise influence metric which could be used in lieu of the proposed model-based topical influence metric.

Self-citations, where at least one author is in common between cited and citing articles, are also informative (Figure 3.4). Authors often build upon their own work, so we would expect self-citations to have higher edge-wise topical influence on average. For ACL the mean topical influence for a self citation edge is 2.80 and for a non-self citation is 1.40. For NIPS the means are 5.05 (self) and 3.15 (non-self). A two-sample t-test finds these differences are both significant at $\alpha = 0.05$.

3.5.2 Prediction Experiments

We also used a document prediction task to explore whether the posited latent structure is predictively useful. We selected roughly 10% of the articles in each corpus (170 and 330 documents for NIPS and ACL, respectively) for testing, chosen among the articles that made at least one citation. We held out a randomly selected set of 50% of their words and evaluated the log probability of the held out partial documents under each model. This is

	ACL			NIPS		
	Wins	Losses	Average Improvement	Wins	Losses	Average Improvement
TIR	297	33	65.7	150	20	38.2
TIRE	276	54	63.0	148	22	38.7
DMR	302	28	79.1	157	13	48.4

Table 3.1: Wins, losses and average improvement for log probabilities of held-out articles, versus LDA. Each "Win" corresponds to the model assigning a higher log probability score for the test portion of a held-out document than LDA assigned to that document.

equivalent to evaluating on a set of new documents with the same set of references as the held out set. Evaluation was performed using annealed importance sampling (Neal, 2001), as in Wallach *et al.* (2009b) except we used multiple samples per likelihood computation.

The TIR models were compared to LDA and an "additive" version of DMR with regression function $\alpha_k^{(d)} = \mathbf{x}^{(d)\intercal}\lambda_k + \alpha$, where the λs were constrained to be positive and given an exponential prior with mean one. In this DMR model, binary feature vectors encoded the presence or absence of each possible citation. The additive variant of DMR can be understood using the Polya urn interpretation of LDA, which we discussed in Section 3.3.2. If a feature j is present (i.e. article j was cited), $x_j^{(d)} = 1$ and λ_{kj} is added to entry k of the Dirichlet prior for the document. We can view this as adding λ_{kj} balls of color k into the urn for that document before beginning the Polya urn process.

For each algorithm, we burned in for 250 iterations, then executed 1000 iterations, optimizing topical influence weights/DMR parameters every 10th iteration. Held-out log probability scores were computed by performing AIS with every 100th sample, and averaging the results to estimate the posterior predictive probability,
Pr(held out article|training set, citations, model).

It was found that all of the regression methods had superior predictive performance to LDA on these corpora, demonstrating that topical influence has predictive value (Table 3.1). Although DMR performed slightly better than TIR predictively, TIR was competitive

despite the fact that it has a factor of K less regression parameters. Note that DMR does not provide an interpretable notion of influence.

3.5.3 Exploring Topical Influence

In this section we explore the inferred topical influence scores $l^{(d)}$, total topical influence scores $T^{(d)}$ and edgewise topical influence scores $l^{(c,d)}$ (recall their definitions in Equations 3.1, 3.2 and 3.3, respectively). Table 3.2 shows the most influential articles in the ACL corpus, according to citation counts, topical influence and total topical influence (the latter two inferred with the TIR model). The most frequently cited paper within the ACL corpus, written by Papineni et al., introduces BLEU, a technique for evaluating machine translation (MT) systems.[6] This paper is of great importance to the computational linguistics community because the method that it introduces is widely used to validate MT systems. However, the BLEU article has a relatively low *topical* influence value of 0.58, consistent with the fact that most of the papers that cite it use the technique as part of their *methodology* but do not *build upon its ideas*. We emphasize that topical influence measures a specific dimension of scientific importance, namely the tendency of an article to influence the ideas (as mediated by the topics) of citing articles; papers with low topical influence such as the BLEU article may be important for other reasons.

Ranking papers by their influence weights $l^{(d)}$ (Table 3.2, middle) has the opposite difficulty to ranking by citation counts — the papers with the highest topical influence were typically cited only once, by the same authors. This makes sense, given what the model is designed to do. The lone citing papers were certainly topically influenced by these articles.

A more useful metric, however, is the total topical influence $T^{(d)}$ (the bottom sub-table in Table 3.2). This is the total number of words of prior concentration, summed over all of

[6] Citations within the corpora are of course only a small fraction of the total set of citations for many of these papers.

Citation Count	**Top 5 Articles by Citation Count**
140	BLEU: a Method for Automatic Evaluation of Machine Translation. *K. Papineni, S. Roukos, T. Ward, W. Zhu.*
105	Minimum Error Rate Training in Statistical Machine Translation. *F. Och.*
64	A Hierarchical Phrase-Based Model for Statistical Machine Translation. *D. Chiang.*
64	Accurate Unlexicalized Parsing. *D. Klein, C. Manning.*
59	Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *D. Yarowsky.*

Topical Influence	**Top 5 articles by Topical Influence**
11.38	Refining Event Extraction through Cross-document Inference. *H. Ji, R. Grishman.*
11.37	Bayesian Learning of Non-compositional Phrases with Synchronous Parsing. *H. Zhang, C. Quirk, R. Moore, D. Gildea.*
10.48	A Plan Recognition Model for Clarification Subdialogues. *D. Litman, J. Allen.*
10.38	PCFGs with Syntactic and Prosodic Indicators of Speech Repairs. *J. Hale, I. Shafran, L. Yung, B. Dorr, and others.*
10.30	Referring as Requesting. *P. Cohen*

Total Topical Influence	**Top 5 Articles by Total Topical Influence**
111.46 (1.74 × 64)	A Hierarchical Phrase-Based Model for Statistical Machine Translation. *D. Chiang.*
101.12 (6.74 × 15)	Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. *D. Xiong, Q. Liu, S. Lin.*
98.56 (5.80 × 17)	A Logical Semantics for Feature Structures. *R. Kasper, W. Rounds.*
85.15 (2.18 × 39)	Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. *F. Och, H. Ney*
81.82 (0.58 × 140)	BLEU: a Method for Automatic Evaluation of Machine Translation. *K. Papineni, S. Roukos, T. Ward, and W. Zhu.*

Table 3.2: Most influential articles in the ACL corpus, according to citation counts (top), topical influence $l^{(d)}$ inferred by TIR (middle), and total topical influence $T^{(d)}$ inferred by TIR (bottom). For total topical influence, the breakdown of $T^{(d)} = l^{(d)} \times$ citation count is shown in parentheses.

Citation Count	**Top 5 Articles by Citation Count**
26	Handwritten Digit Recognition with a Back-Propagation Network. *Y. Le Cun, et al.*
19	Optimal Brain Damage. *Y. Le Cun, J. Denker, S. Solla.*
17	A New Learning Algorithm for Blind Signal Separation. *S. Amari, A. Cichocki, H. Yang.*
17	Efficient Pattern Recognition Using a New Transformation Distance. *P. Simard, Y. Le Cun, J. Denker.*
14	The Cascade-Correlation Learning Architecture. *S. Fahlman, C. Lebiere.*

Topical Influence	**Top 5 articles by Topical Influence**
29.7	Synchronization and Grammatical Inference in an Oscillating Elman Net. *B. Baird, T. Troyer, F. Eeckman.*
26.3	Learning the Solution to the Aperture Problem for Pattern Motion with a Hebb Rule. *M. Sereno.*
25.9	ALVINN: An Autonomous Land Vehicle in a Neural Network. *D. Pomerleau.*
25.1	Some Estimates of Necessary Number of Connections and Hidden Units for Feed-Forward Networks. *A. Kowalczyk.*
24.7	Complex- Cell Responses Derived from Center-Surround Inputs: The Surprising Power of Intradendritic Computation. *B. Mel, D. Ruderman, K. Archie.*

Total Topical Influence	**Top 5 Articles by Total Topical Influence**
84.7 (10.6 × 8)	Gaussian Processes for Regression. *C. Williams, C. Rasmussen.*
63.9 (7.1 × 9)	Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems. *T. Jaakkola, S. Singh, M. Jordan.*
57.9 (19.3 × 3)	Optimal Stopping and Effective Machine Complexity in Learning. *C. Wang, S. Venkatesh, J. Judd.*
54.7 (10.9 × 5)	Links Between Markov Models and Multilayer Perceptrons. *H. Bourlard, C. Wellekens.*
51.2 (3.7 × 14)	The Cascade-Correlation Learning Architecture. *S. Fahlman, C. Lebiere.*

Table 3.3: Most influential articles in the NIPS corpus, according to citation counts (top), topical influence $l^{(d)}$ inferred by TIR (middle), and total topical influence $T^{(d)}$ inferred by TIR (bottom).

A Hierarchical Phrase-Based Model for Statistical Machine Translation. D. Chiang.		
Most influential reference	1.48	Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. *F. Och and H. Ney.*
Least influential reference	0.00	BLEU: a Method for Automatic Evaluation of Machine Translation. *K. Papineni, S. Roukos, T. Ward, W. Zhu.*
Most influenced citer	2.54	Toward Smaller, Faster, and Better Hierarchical Phrase-based SMT. *M. Yang, J. Zheng.*
Least influenced citer	0.60	An Optimal-time Binarization Algorithm for Linear Context-Free Rewriting Systems with Fan-out Two. *C. Gómez-Rodríguez, G. Satta.*
Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. D. Yarowsky.		
Most influential reference	2.52	Subject-dependent Co-occurrence and Word Sense Disambiguation. *J. Guthrie, L. Guthrie, Y. Wilks, H. Aidinejad.*
Least influential reference	0.53	Word-sense Disambiguation using Statistical Methods. *P. Brown, S. Della Pietra, V. Della Pietra, R. Mercer.*
Most influenced citer	1.81	Discriminating Image Senses by Clustering with Multimodal Features. *N. Loeff, C. Alm, D. Forsyth.*
Least influenced citer	0.00	Semi-supervised Convex Training for Dependency Parsing. *Q. Wang, D. Schuurmans, D. Lin.*
Accurate Unlexicalized Parsing. D. Klein, C. Manning.		
Most influential reference	3.87	Parsing with Treebank Grammars: Empirical Bounds, Theoretical Models, and the Structure of the Penn Treebank. *D. Klein and C. Manning.*
Least influential reference	0.81	Efficient Parsing for Bilexical Context-Free Grammars and Head Automaton Grammars. *J. Eisner, G. Satta.*
Most influenced citer	1.67	Evaluating the Accuracy of an Unlexicalized Statistical Parser on the PARC DepBank. *T. Briscoe, J. Carroll.*
Least influenced citer	0.00	Finding Contradictions in Text. *M. de Marneffe, A. Rafferty, C. Manning.*

Table 3.4: Least and most influential references and citers, and the influence weights along these edges, inferred by the TIRE model for three example ACL articles.

Feudal Reinforcement Learning. P. Dayan, G. Hinton		
Most influential reference	5.47	Memory-based Reinforcement Learning: Efficient Computation with Prioritized Sweeping. *A. Moore, C. Atkeson.*
Least influential reference	0.00	A Delay-Line Based Motion Detection Chip. *T. Horiuchi, J. Lazzaro, A. Moore, C. Koch.*
Most influenced citer	3.36	The Parti-Game Algorithm for Variable Resolution Reinforcement Learning in Multidimensional State-Spaces. *A. Moore.*
Least influenced citer	1.71	Multi-time Models for Temporally Abstract Planning. *D. Precup, R. Sutton.*
Optimal Brain Damage. Y. Le Cun, J. Denker , S. Solla.		
Most influential reference	2.82	Comparing Biases for Minimal Network Construction with Back-Propagation. *S. Hanson, L. Pratt.*
Least influential reference	0.15	Skeletonization: A Technique for Trimming the Fat from a Network via Relevance Assessment. *M. Mozer, P. Smolensky.*
Most influenced citer	3.08	Structural Risk Minimization for Character Recognition. *I. Guyon, V. Vapnik, B. Boser, L. Bottou, S. Solla.*
Least influenced citer	0.64	Structural and Behavioral Evolution of Recurrent Networks. *G. Saunders, P. Angeline, J. Pollack.*
An Input Output HMM Architecture. Y. Bengio, P. Frasconi.		
Most influential reference	5.29	Credit Assignment through Time: Alternatives to Backpropagation. *Y. Bengio, P. Frasconi.*
Least influential reference	0.00	Induction of Multiscale Temporal Structure. *M. Mozer*
Most influenced citer	2.66	Learning Fine Motion by Markov Mixtures of Experts. *M. Meila, M. Jordan.*
Least influenced citer	1.47	Recursive Estimation of Dynamic Modular RBF Networks. *V. Kadirkamanathan, M. Kadirkamanathan.*

Table 3.5: Least and most influential references and citers, and the influence weights along these edges, inferred by the TIRE model for three example NIPS articles.

its citers, that the article has contributed, and is a measure of the total corpus-wide topical influence of the paper. In the Polya urn interpretation of LDA, this is the total number of balls that the article added to the urns of the other articles in the corpus. This metric ranks the BLEU paper at 5th place, down from 1st place by citation count. The ACL paper with the highest total topical influence, by David Chiang, won the ACL best paper award in 2005.

The behavior of the different metrics is echoed in the NIPS corpus (Table 3.3). The most cited paper, "Handwritten Digit Recognition," by Le Cun *et al.* (1990), is an early successful application of neural networks. The paper does not introduce novel models or algorithms, but rather, in the authors' words, "show[s] that large back propagation (BP) networks can be applied to real image recognition problems." Thus, although it is has an important role as a landmark neural network success story, it does not score highly in terms of *topical* influence. This paper is ranked 13th according to total topical influence, with a score of 1.6. The top two-ranked papers according to total topical influence, on Gaussian Process Regression and POMDPs respectively, were both seminal papers that spawned large bodies of related work. An interesting case is the third-ranked paper in the NIPS corpus, by Wang et al., on the theory of early stopping. It is only referenced three times, but has a very high topical influence of 19.3 words. All three citing papers are also on the theory of early stopping, and one of the papers, by Wang and Venkatesh, directly extends a theoretical result of this paper. Although it is easy to see why this paper scores highly on topical influence, in this case the metric has perhaps overstated its importance. A limitation of topical influence is that it can potentially give more credit than is due when an article is cited by a small number of topically similar papers, due to overfitting. This is likely to be an issue for any topic-based approach for modeling scientific influence. However, topics help to absorb lexical ambiguity and author-specific idiosyncracies, mitigating the problem relative to word-based approaches.

Using the TIRE model, we can also look at influence relationships between pairs of articles. Tables 3.4 and 3.5 show the most and least topically influential references, and the most and least influenced citing papers, for three example articles from ACL and NIPS, respectively. The model correctly assigns higher influence scores along the edges to and from relevant documents. For the ACL papers, the BLEU algorithm's article is inferred to have zero topical influence on Chiang's paper, consistent with its role in the paper as an evaluation technique. The paper most topically influenced by Chiang's paper, written by Yang and Zheng, aims to improve upon the ideas in that paper. In the NIPS corpus, the article by Bengio and Frasconi, on recurrent neural network architectures, extends previous work by the same authors, which is correctly assigned the highest topical influence. A particularly interesting case is the paper by Dayan and Hinton, which is heavily influenced by a paper by Moore, and in turn strongly influences a later paper by Moore, thus illustrating the interplay of scientific influence between authors along the citation graph. These three papers were on reinforcement learning, while the lowest scoring reference and citer were on other subjects.

3.6 Summary of Contributions

This chapter introduced a latent variable model called topical influence regression (TIR). The model is used to define the notion of topical influence, a quantitative measure of scientific impact. TIR builds upon the ideas of Dirichlet-multinomial regression to encode influence relationships between articles, creating a Bayesian network of dependencies between documents which follows the structure of the citation DAG. By training TIR, we can recover topical influence scores that give us insight into the impact of scientific articles. The model was applied to two scientific corpora, demonstrating the utility of the method both quantitatively and qualitatively.

The main contributions of this chapter are

- We introduced topical influence regression (TIR), a probabilistic model of corpora of scientific articles with citation links between them.
- In the context of this model, we defined *topical influence* and *total topical influence*, metrics of the importance of scientific works which make use of both text and citation network information.
- We proposed a hierarchical extension of the TIR model, TIRE, where influence weights are computed on the edges of the citation graph as well as the nodes, while sharing statistical strength between edges through inference on node-level parameters.
- It was shown how to perform inference on these models using a Markov Chain Monte Carlo technique, using a stochastic EM approach to learn the influence weights.
- We evaluated the models both qualitatively and quantitatively on two corpora of scientific articles. The models were validated by comparing the results with relevant metadata, namely within-article citation counts and self-citation relationships. We also validated the models on a prediction task, and qualitatively explored the output of the model on an exploratory data analysis task, demonstrating the utility of the approach.

Chapter 4

Fast Online Inference for Topic Models

> All we have to decide is what to do with the time that is given us.
>
> *J.R.R. Tolkien, the Fellowship of the Ring*

In the previous chapters of this thesis, we have seen that latent variable models can be powerful tools for finding meaningful hidden structure in our data. As discussed in Chapter 1, a key motivation for these models is to help make sense of the vast body of digital information on the internet. As the amount of available data continues to grow, so does the need for automatic tools to make sense of it. More data also brings with it the potential to improve the accuracy of our methods, and to support complex models which capture more aspects of the data.

There is a particularly clear need for tools which can learn topic models such as LDA at the "web scale," especially for web companies whose lifeblood is textual content on the internet. For example, news aggregator websites such as Yahoo! News publish a continually updated stream of online articles. These services need to analyze candidate articles for

topical diversity, relevance to current trends, and personalized recommendation, all of which can be facilitated by topic models (Ahmed *et al.* , 2011).

However, traditional inference techniques for these models such as Gibbs sampling and variational inference do not readily scale to such corpora, which frequently contain millions of documents or more. In such cases it is very time-consuming to perform even a single iteration of the collapsed Gibbs sampling (Griffiths & Steyvers, 2004) or variational inference algorithms (Blei *et al.* , 2003) for topic models, let alone run them until convergence. Clearly, more scalable approaches are needed.

A significant recent advance was made by Hoffman *et al.* (2010, 2013), who proposed a *stochastic* variational inference algorithm for LDA topic models. Because the algorithm does not need to see all of the documents before updating the topics, this method can often learn good topics before even a single iteration of the traditional batch inference algorithms is completed. The algorithm processes documents in an online fashion, so it can be applied to corpora of any size, or even to never-ending streams of documents. Thus, the stochastic approach is scalable in terms of the number of documents to process. A variant of the algorithm has been proposed which is also scalable in the size of the vocabulary and the number of topics (Mimno *et al.* , 2012).

A complementary direction that has been useful for improving inference in LDA is to take advantage of its *collapsed* representation, where parameters are marginalized out, leaving only latent variables. It is possible to perform inference in the collapsed space and recover estimates of the parameters afterwards. For inference techniques that operate in a batch setting, the algorithms that operate in the collapsed space are more efficient at improving held-out log probability than their uncollapsed counterparts, both per iteration and in wall-clock time per iteration (Griffiths & Steyvers, 2004; Teh *et al.* , 2007a; Asuncion *et al.* , 2009). For variational inference, perhaps the most important advantage of the collapsed rep-

	Variational Bayes	Collapsed Gibbs Sampling	Collapsed Variational Bayes
Batch	Blei *et al.* (2003)	Griffiths & Steyvers (2004)	Teh *et al.* (2007a)
Stochastic	Hoffman *et al.* (2010)	Mimno *et al.* (2012) (VB/ Gibbs hybrid)	**This chapter**

Table 4.1: LDA learning approaches can be divided into the algorithmic method used (*columns*), and by whether the approach is stochastic or batch mode (*rows*). This chapter fills in the bottom-right entry of the table by introducing a stochastic collapsed variational Bayes algorithm for LDA.

resentation is that the variational bound is strictly better than that for the uncollapsed representation, leading to the potential for collapsed variational algorithms to learn more accurate topic models than uncollapsed variational algorithms (Teh *et al.* , 2007a). Existing online inference algorithms for LDA do not fully take advantage of the collapsed representation.

In this chapter, we develop a stochastic algorithm for LDA that operates in the collapsed space, thus gaining the aforementioned advantages of both collapsed and online algorithms (see Table 4.1). This facilitates learning topic models both more accurately and more quickly on large datasets. The proposed algorithm is also very simple to implement, requiring only basic arithmetic operations. To validate the approach, we test its performance experimentally on three large web-scale datasets, with comparison to the previous uncollapsed approach. We also explore the benefit of our method on small problems, showing that it is feasible to learn human-interpretable topics in seconds.[1]

We begin the chapter with essential background material on the relevant prior work (Section 4.1). This includes introductions to variational inference, to the collapsed representation of LDA, to the collapsed variational approach, and to stochastic optimization. Section 4.2 introduces the proposed inference algorithm for topic models, which we refer to as *SCVB0*. In Section 4.3, we evaluate the SCVB0 algorithm on both large-scale and small-scale problems.

[1]Much of this chapter is published work, available in Foulds *et al.* (2013).

The remaining technical sections of this chapter are more theoretical in nature. In Section 4.4 we take a close look at the approximations made by the *CVB0* algorithm of Asuncion *et al.* (2009), which is the basis for our method, and provide a new justification for the approach. An alternative perspective is given in Section 4.5, where we show connections between SCVB0 and an algorithm which performs MAP estimation. Leveraging these connections, we prove the convergence of the algorithm in Section 4.6. We put our approach in context in Section 4.7, by discussing related work in the literature on the scalable learning of topic models. Finally, we conclude the chapter in Section 4.8 with a summary of the contributions made here.

4.1 Background

This chapter develops a new algorithm for learning LDA topic models which leverages several strands of research. The algorithm uses the framework of *variational inference* to cast the problem of training a model as an optimization problem. The inference is performed using the *collapsed* representation of LDA, where only the latent variables of the model are reasoned over, thus simplifying and streamlining the process. Finally, the approach leverages *stochastic* optimization techniques to make the algorithm scalable to large datasets. In this section, we provide a tutorial on each of these concepts, as well as an overview of the previous work on variational inference for collapsed LDA, before proceeding to detail our new approach in Section 4.2.

4.1.1 Variational Inference

Variational inference (Dayan *et al.* , 1995; Jaakkola & Jordan, 1997; Jordan *et al.* , 1999) is an optimization approach to solving inference problems. The word "*variational*" refers to

a field of mathematical analysis, dating back to Euler and Lagrange, called the *calculus of variations*, which concerns optimization over the space of functions. Variational methods are applicable in a probabilistic modeling context because probability distributions are functions, and inference problems can often be cast as optimization problems over these distributions.

The application of such variational methods to solving inference problems in a Bayesian context is called *variational Bayesian inference*, or *variational Bayes* (*VB*). Typically, VB corresponds to using variational inference to estimate a Bayesian posterior. The VB approach is fully Bayesian in the sense that it estimates the full posterior distribution. This should be contrasted to maximum likelihood estimation and maximum a posteriori probability (MAP) estimation, which find only point estimates of the parameters of interest. However, while VB is a Bayesian technique, variational methods are not inherently Bayesian, and they can be applied in other contexts as well. For example, variational inference may be used within an inner loop of an EM algorithm for performing maximum likelihood estimation, by estimating a distribution over a set of hidden variables. This strategy is known as *variational EM*. A well-known example of this is the original learning algorithm for LDA (Blei *et al.* , 2003).

In the context of variational inference, suppose we would like to compute a posterior distribution $p(\mathbf{z}|\mathbf{x})$ over hidden variables and parameters $\mathbf{z}$, having observed data $\mathbf{x}$.[2] It is assumed that the posterior is intractable, and so approximation techniques must be used. The key idea of variational inference is to approximate $p(\mathbf{z}|\mathbf{x})$ with a more tractable distribution $q(\mathbf{z})$, and minimize some distance (or divergence) between the two distributions. We refer to $q(\mathbf{z})$ as the *variational distribution*. This is a variational technique because we are optimizing over a function, in this case $q(\mathbf{z})$, which is a mapping from variable assignments to their probabilities (or probability densities).

[2]Here, we may also be implicitly conditioning on learned parameter values and/or hyper-parameters.

KL-divergence and the Evidence Lower Bound

Typically the divergence to minimize is chosen to be the KL-divergence from $q(\mathbf{z})$ to $p(\mathbf{z}|\mathbf{x})$,

$$\begin{aligned} D_{KL}(q(\mathbf{z})\|p(\mathbf{z}|\mathbf{x})) &= E_q\Big[\frac{\log q(\mathbf{z})}{\log p(\mathbf{z}|\mathbf{x})}\Big] \\ &= E_q[\log q(\mathbf{z})] - E_q[\log p(\mathbf{z}|\mathbf{x})] \\ &= E_q[\log q(\mathbf{z})] - E_q[\log p(\mathbf{z},\mathbf{x})] + \log p(\mathbf{x}) \ . \end{aligned} \tag{4.1}$$

The KL-divergence makes intuitive sense as such an objective function, as it is zero if the distributions are equal, and greater than zero otherwise.[3] We can think of it as the number of extra bits of information needed to encode both the approximate posterior and the data, beyond using the true posterior. This "bits back" motivation of the KL-divergence as a minimization objective in a Bayesian context is due to Hinton & Van Camp (1993), in what is perhaps the earliest paper on VB.[4]

The arg min of Equation 4.1 with respect to $q(\mathbf{z})$ does not depend on the constant $\log p(\mathbf{x})$. Minimizing it is therefore equivalent to maximizing

$$\mathcal{L}(q) \triangleq E_q[\log p(\mathbf{z},\mathbf{x})] - E_q[\log q(\mathbf{z})] = E_q[\log p(\mathbf{z},\mathbf{x})] + H(q) \ , \tag{4.2}$$

where $H(q)$ is the entropy (or differential entropy) of $q(\mathbf{z})$. Here, the entropy of $q(\mathbf{z})$ rewards simplicity, while $E_q[\log p(\mathbf{z},\mathbf{x})]$, the expected value of the complete data log-likelihood under the variational distribution, rewards accurately fitting to the data.

[3] Note that the KL-divergence is not symmetric, and we could instead to have chosen to optimize over $D_{KL}(p(\mathbf{z}|\mathbf{x})\|q(\mathbf{z}))$. This alternative variational inference strategy, known as *expectation propagation* (Minka, 2001), is sometimes used, but it results in a more difficult optimization problem. We will not consider it further here.

[4] See also the follow-up papers by Hinton and colleagues, (Hinton & Zemel, 1994; Dayan *et al.* , 1995).

As a side note, we can alternatively derive $\mathcal{L}(q)$ as a lower bound on the log of the marginal probability of the data $p(\mathbf{x})$ (the *evidence*),

$$\begin{aligned}
\log p(\mathbf{x}) &= \log(\sum_{\mathbf{z}} p(\mathbf{z}, \mathbf{x})) \\
&= \log(\sum_{\mathbf{z}} p(\mathbf{z}, \mathbf{x}) \frac{q(\mathbf{z})}{q(\mathbf{z})}) \\
&= \log(E_q\Big[\frac{p(\mathbf{z}, \mathbf{x})}{q(\mathbf{z})}\Big]) \\
&\geq E_q\Big[\log p(\mathbf{z}, \mathbf{x}) - \log q(\mathbf{z})\Big] \\
&= \mathcal{L}(q) \ ,
\end{aligned} \tag{4.3}$$

where we have made use of Jensen's inequality.[5] In the above, summations can be replaced with integrals for continuous $\mathbf{z}$. Due to Equation 4.3, $\mathcal{L}(q)$ is referred to as the *evidence lower bound* (ELBO). As we have seen, maximizing $\mathcal{L}(q)$ can be understood as either minimizing the KL-divergence from our approximating distribution $q(\mathbf{z})$ to the posterior (Equation 4.1), or as maximizing a lower bound on the evidence $p(\mathbf{x})$ (Equation 4.3). The ELBO $\mathcal{L}(q)$ will be our objective function in what follows.

Mean Field Variational Inference

So far, we have not made the problem any easier, as our objective function $\mathcal{L}(q)$ still contains expectations over the hidden variables $\mathbf{z}$. What we have gained, however, is the option to select a variational distibution $q(\mathbf{z})$ for which these expectations are tractable, in order to facilitate efficient optimization.

[5] In a probability context, Jensen's inequality states that for any concave function f such as log, $f(E[X]) \geq E[f(X)]$).

One strategy with a clear advantage in this regard is to use a fully factorized distribution,

$$q(\mathbf{z}) = \prod_i q_i(z_i) \ , \tag{4.4}$$

which leads to the entropy term decomposing into a much more tractable form,

$$-E_q[\log q(\mathbf{z})] = -E_q[\log \prod_i q_i(z_i)] = \sum_i E_{q_i}[-\log q_i(z_i)] = \sum_i H(q_i) \ . \tag{4.5}$$

The expected complete data log-likelihood also typically becomes much more tractable in this setup. This factorization strategy for selecting a variational family q is known as *mean field*, by analogy to behaviors of particles in statistical physics.

The Mean Field Update Equations

The mean field independence assumption in the variational distribution implies that each factor is relatively tractable, individually. This suggests a coordinate ascent approach to the optimization of the ELBO for mean field VB, in which each factor is optimized in turn. It is possible to derive the general structure of the updates for this algorithm without making any further distributional assumptions or assuming a specific parameterization for $q(\mathbf{z})$. We will derive this below. It is important to note that because we have not yet specified any parameterization for $q(\mathbf{z})$, the optimization we are performing is with respect to the *function* $q(\mathbf{z})$, rather than being with respect to specific *parameters*. This is sometimes referred to as *free-form optimization*. Later, we will describe how this applies once we have specified a parametric form for $q(\mathbf{z})$ for the specific problem at hand.

To derive the coordinate ascent updates, we first isolate the terms in $\mathcal{L}(q)$ associated with factor i,

$$\begin{aligned}
\mathcal{L}(q) &= E_q[\log p(\mathbf{z}, \mathbf{x})] + \sum_j H(q_j) \\
&= \sum_{\mathbf{z}} \prod_j q_j(z_j) \Big(\log p(\mathbf{z}, \mathbf{x}) \Big) + H(q_i) + \text{ const} \\
&= \sum_{z_i} q_i(z_i) \sum_{\mathbf{z}_{\neg i}} \prod_{j \neq i} q_j(z_j) \Big(\log p(\mathbf{z}, \mathbf{x}) \Big) + H(q_i) + \text{ const} \\
&= \sum_{z_i} q_i(z_i) E_{q_{\neg i}}[\log p(\mathbf{z}, \mathbf{x})] + H(q_i) + \text{ const.} \qquad (4.6)
\end{aligned}$$

Notice that this is starting to look like a negative KL-divergence from q to some "distribution" $E_{q_{\neg i}}[\log p(\mathbf{z}, \mathbf{x})]$, except that $E_{q_{\neg i}}[\log p(\mathbf{z}, \mathbf{x})]$ is not a normalized distribution. If we can rewrite this with some normalized distribution, we can interpret the equation as a KL-divergence and we will be on familiar ground. To this end, let us construct a normalized probability distribution

$$\begin{aligned}
f_i(z_i) &= \frac{\exp(E_{q_{\neg i}}[\log p(\mathbf{z}, \mathbf{x})])}{\exp A_i} \\
&= \exp(E_{q_{\neg i}}[\log p(\mathbf{z}, \mathbf{x})] - A_i) \ , \qquad (4.7)
\end{aligned}$$

with $A_i = \log \sum_{i'} \exp(E_{q_{\neg i'}}[\log p(\mathbf{z}, \mathbf{x})])$ being the log of the normalizing constant, often referred to as the *log partition function*. Then we have

$$\begin{aligned}
\mathcal{L}(q) &= \sum_{z_i} q_i(z_i)(\log f_i(z_i) + A_i) + H(q_i) + \text{ const} \qquad (4.8) \\
&= \sum_{z_i} q_i(z_i) \log f_i(z_i) + H(q_i) + \text{ const} \qquad (4.9) \\
&= -D_{KL}(q_i(z_i) \| f_i(z_i)) + \text{const} \ . \qquad (4.10)
\end{aligned}$$

KL-divergences are minimized by setting the two distributions equal, in which case the divergence is zero. So we can maximize the above by setting $q_i(z_i) = f_i(z_i)$, which leads to the update

$$q_i(z_i) :\propto \exp(E_{q_{\neg i}}[\log p(\mathbf{z}, \mathbf{x})]) \ , \tag{4.11}$$

where $:\propto$ means "assigned to be proportional to."

When implementing this in practice, we typically will parameterize each $q_i(z_i)$ using some parametric form with *variational parameters* γ_i,

$$q(\mathbf{z}) = \prod_i q_i(z_i|\gamma_i) \ . \tag{4.12}$$

For example, if $\mathbf{z}$ is a categorical variable, then γ_i is the parameter vector for a discrete distribution, which sums to one. In this case, parameterizing by γ_i has not required any further assumptions on $q(\mathbf{z})$, since any categorical random variable can be written this way. We can obtain the update equations for the specific problem at hand by plugging our specific $p(\mathbf{z}, \mathbf{x})$ into Equation 4.11 to deduce the update for each γ_i. Regardless of the parameterization for the γ_i's, the coordinate ascent algorithm optimizes $\mathcal{L}(q)$ with respect to γ by updating each of the variational parameters γ_i for each factor $q_i(z_i)$ in turn. The updates are iterated until convergence. Each update monotonically improves $\mathcal{L}(q)$ so the algorithm is guaranteed to converge.

4.1.2 Collapsed LDA

We now return our discussion to LDA models. In the collapsed representation of LDA, due to (Griffiths & Steyvers, 2004), we marginalize out topics $\mathbf{\Phi}$ and distributions over topics $\mathbf{\Theta}$, and perform inference only on the topic assignments $\mathbf{z}$. This is possible due to the conjugacy

of the Dirichlet distribution and the multinomial distribution. Let us begin with the full joint distribution, and then proceed by marginalization. Referring back to the graphical model (Figure 1.4) and the generative process (Algorithm 2) for LDA, the joint distribution for the parameters and latent variables can be written as

$$Pr(w, \mathbf{z}, \mathbf{\Phi}, \mathbf{\Theta}|\alpha, \beta) = \prod_{k=1}^{K} \Big(\text{Dirichlet}(\Phi^{(k)}|\beta)\Big) \prod_{d=1}^{D} \Big(\text{Dirichlet}(\theta^{(d)}|\alpha)\Big) \times \prod_{d=1}^{D} \prod_{i=1}^{N_d} \Big(\text{Discrete}(z_i^{(d)})|\theta^{(d)}) \text{Discrete}(w_i^{(d)}|\Phi^{(z_i^{(d)})})\Big) \ . \tag{4.13}$$

The discrete and Dirichlet distributions used here are

$$\text{Discrete}(x|\pi) = \pi_x \tag{4.14}$$

$$\text{Dirichlet}(\pi|\mathbf{a}) = \frac{1}{B(\mathbf{a})} \prod_{k=1}^{K} \pi_k^{a_k - 1} \tag{4.15}$$

$$B(\mathbf{a}) = \int \prod_{k}^{K} \pi_k^{a_k - 1} d\pi = \frac{\prod_{k=1}^{K} \Gamma(a_k)}{\Gamma(\sum_k a_k)} \ , \tag{4.16}$$

where $B(\mathbf{a})$ is the multivariate beta function, $\Gamma(n)$ is the gamma function, and π sums to one. Before deriving collapsed LDA, as a warm-up we will first pause to consider a simpler model which can be constructed using these two distributions. The model is specified by the following generative process:

- $\pi \sim \text{Dirichlet}(\mathbf{a})$
- For $i = 1$ to N
  - $x_i \sim \text{Discrete}(\pi)$.

We can write down the joint distribution of this simple model as

$$Pr(\mathbf{x}, \pi|\mathbf{a}) = \text{Dirichlet}(\pi|\alpha) \prod_{i=1}^{N} \text{Discrete}(x_i|\pi) \tag{4.17}$$

$$= \frac{1}{B(\mathbf{a})} \prod_{k=1}^{K} \pi_k^{a_k+n_k-1} \ , \tag{4.18}$$

where n_k is the number of times that $x_i = k$. We can compute $Pr(\mathbf{x}|\mathbf{a}) = \int Pr(\mathbf{x}, \pi|\mathbf{a}) d\pi$ using an application of Equation 4.16,

$$Pr(\mathbf{x}|\mathbf{a}) = \frac{B(\mathbf{a}+\mathbf{n})}{B(\mathbf{a})} \ . \tag{4.19}$$

The distribution $Pr(\mathbf{x}|\mathbf{a})$, known as the *multivariate Polya distribution*, corresponds to the urn model which we discussed in Section 1.5 of the introduction to this thesis. Expanding out the beta functions, we can write this as

$$\text{Polya}(\mathbf{x}|\mathbf{a}) \triangleq Pr(\mathbf{x}|\mathbf{a}) = \frac{\Gamma(\sum_k a_k)}{\Gamma(\sum_k n_k + \sum_k a_k)} \prod_{k=1}^{K} \frac{\Gamma(n_k + a_k)}{\Gamma(a_k)} \ .$$

Returning to the LDA model, we can recover the collapsed model, marginalizing out $\mathbf{\Theta}$ and $\mathbf{\Phi}$, by applying the above argument once for every document d, with $Pr(\mathbf{z}^{(d)}|\alpha) = \int Pr(\mathbf{z}^{(d)}, \theta^{(d)}|\alpha) d\theta^{(d)}$ and once for each topic, $Pr(\{w_i^{(d)}|z_i^{(d)} = k\}|\beta) = \int Pr(\{w_i^{(d)}|z_i^{(d)} = k\}, \Phi^{(k)}|\beta) d\Phi^{(k)}$. The resulting marginal distribution is

$$\begin{aligned} Pr(w, \mathbf{z}|\alpha, \beta) &= \prod_{d=1}^{D} \Big(\text{Polya}(\mathbf{z}^{(d)}|\alpha) \Big) \prod_{k=1}^{K} \Big(\text{Polya}(\{w_i^{(d)}|z_i^{(d)} = k\}|\beta) \Big) \\ &= \prod_{d=1}^{D} \Big(\frac{\Gamma(\sum_k \alpha_k)}{\Gamma(n^{(d)} + \sum_k \alpha_k)} \prod_{k=1}^{K} \frac{\Gamma(n_k^{(d)} + \alpha_k)}{\Gamma(\alpha_k)} \Big) \times \\ &\prod_{k=1}^{K} \Big(\frac{\Gamma(\sum_w \beta_k)}{\Gamma(n_k + \sum_w \beta_w)} \prod_{w=1}^{W} \frac{\Gamma(n_k^{(w)} + \beta_w)}{\Gamma(\beta_w)} \Big) \ , \end{aligned} \tag{4.20}$$

where $n^{(d)}$ is the length of document d, $n_k^{(d)}$ is the number of times that words in document d are assigned to topic k, n_k is the number of words assigned to topic k, and $n_k^{(w)}$ is the number of times words with index w are assigned to topic k.

The *collapsed Gibbs sampler* (CGS) of Griffiths & Steyvers (2004) operates in this marginalized (a.k.a. *collapsed*) representation, updating each $z_i^{(d)}$ in turn. Starting from Equation 4.20, after some algebra we arrive at the Gibbs update equation for the CGS algorithm,

$$Pr(z_i^{(d)} = k | z^{-(d,i)}, \ldots) \propto (n_k^{(d)-(d,i)} + \alpha_k) \frac{n_k^{(w_i^{(d)})-(d,i)} + \beta_{w_i^{(d)}}}{n_k^{-(d,i)} + \sum_w \beta_w} \ . \tag{4.21}$$

The CGS algorithm is perhaps the most widely used training algorithm for LDA. Reasons for its popularity include the fact that it mixes much better than the naive Gibbs sampler, it is computationally efficient and that it is also easy to implement. It is also amenable to speed improvements using clever implementations (Porteous *et al.* , 2008; Yao *et al.* , 2009), and is robust to parallel approximations (Newman *et al.* , 2009; Smola & Narayanamurthy, 2010). A high quality implementation using all of these optimizations is available in the MALLET software package (McCallum, 2002).

4.1.3 Collapsed Variational Bayesian Inference for LDA

As well as facilitating an effective Gibbs sampling algorithm, the collapsed representation for LDA is also advantageous for the variational approaches we will consider in this section. The Gibbs sampling and variational strategies for performing inference each have different strengths. The Gibbs sampler is unbiased, and so its long-run performance is better than variational methods, at least if we are willing accept the cost of averaging over many samples at prediction time. On the other hand, the variational method is an optimization procedure. This allows variational algorithms to travel more directly "uphill" in the search space, instead of the random walk behavior exhibited by MCMC algorithms, which often results in faster convergence. An optimization framework such as VB also enables the use of stochastic optimization techniques, as explored in the new technique introduced in this chapter.

For the LDA model, Teh *et al.* (2007a) and Asuncion *et al.* (2009) showed how collapsing can improve performance over the original variational approach of Blei *et al.* (2003), by simplifying the algorithms and by improving the tightness of the variational bound. We will sketch the derivation of this technique below.

The collapsed variational Bayesian inference (CVB) approach of Teh *et al.* (2007a) begins by marginalizing out the topic $\mathbf{\Phi}$ and distributions over topics $\mathbf{\Theta}$. In the collapsed space, the method performs a mean field variational approximation on the topic assignments $\mathbf{z}$,

$$q(\mathbf{z}) = \prod_{d=1}^{D} \prod_{i=1}^{n^{(d)}} q_{id}(z_i^{(d)}|\gamma_{id}) \ , \tag{4.22}$$

where $q_{id}(z_i^{(d)}|\gamma_{id}) = \text{Discrete}(z_i^{(d)}|\gamma_{id})$, and γ_{id} is a K-dimensional vector of variational parameters which sums to one. Here, γ_{idk} is the probability that $z_i^{(d)} = k$ according to the variational distribution. Since $\mathbf{z}_i^{(d)}$ is a categorical variable, parameterizing $q_{id}(z_i^{(d)})$ via a discrete distribution with parameters γ_{id} does not correspond to any further assumption on the variational distribution $q(\mathbf{z})$ beyond the mean field assumption.

Motivating this mean field collapsed VB approach, Teh et al. note that the CGS update of Equation 4.21 shows us that the $z_i^{(d)}$'s affect each other only through aggregate counts of the topic assignments. Thus, the dependence between any variable $z_i^{(d)}$ and any other variable $z_{i'}^{(d')}$ is weak, which suggests that this is a scenario where a mean field assumption is likely to be reasonable. Furthermore, Teh et al. show that the variational bound (i.e. Equation 4.3) which results from Equation 4.22 is strictly better than the variational bound for the standard VB algorithm over all of the variables, as used in the algorithm of Blei *et al.* (2003). This is because the standard VB approach assumes a factorized representation over each of the parameters in $\mathbf{\Phi}$ and $\mathbf{\Theta}$, while this assumption is not made when mean field VB is performed in the collapsed space.

To optimize the evidence lower bound (Equation 4.3) of collapsed VB, Equation 4.11 gives us the mean field coordinate ascent update:

$$\gamma_{idk} :\propto \exp(E_{q_{\neg id}}[\log Pr(\mathbf{z}_{\neg id}, z_i^{(d)} = k, w|\alpha, \beta)]) \ . \tag{4.23}$$

After plugging Equation 4.20 into Equation 4.23 and performing some algebraic manipulation, Teh et al. arrive at

$$\gamma_{idk} :\propto \exp\Big(E_{q_{\neg id}}[\log(n_k^{(d)-(d,i)} + \alpha_k)] + E_{q_{\neg id}}[\log(n_k^{(w_i^{(d)})-(d,i)} + \beta_{w_i^{(d)}})] \\ - E_{q_{\neg id}}[\log(n_k^{-(d,i)} + \sum_w \beta_w)]\Big) \ . \tag{4.24}$$

To implement the algorithmic update step corresponding to this equation, Teh et al. show that each of the expectations in Equation 4.24 can be calculated with a running time quadratic in the number of words involved in the expectation, using a convolution technique. Unfortunately, this algorithm is not efficient enough to be practical.

Nonetheless, Teh et al. introduce an algorithm based on an approximation to Equation 4.24. They find that this approximate algorithm works well in practice, outperforming the standard VB algorithm in terms of predictive performance. The method approximates the expectation terms in Equation 4.24 by

$$E_{q_{\neg id}}[\log(n_k^{(d)-(d,i)} + \alpha_k)] \approx \log(E_{q_{\neg id}}[n_k^{(d)-(d,i)}] + \alpha_k) - \frac{\mathrm{Var}_{q_{\neg id}}[n_k^{(d)-(d,i)}]}{2(\alpha_k + E_{q_{\neg id}}[n_k^{(d)-(d,i)}])^2} \tag{4.25}$$

and similarly for the other terms, $E_{q_{\neg id}}[\log(n_k^{(w_i^{(d)})-(d,i)} + \beta_{w_i^{(d)}})]$ and $E_{q_{\neg id}}[\log(n_k^{-(d,i)} + \sum_w \beta_w)]$.

This approximation is motivated by observing that in the variational distribution q, the count variables, e.g. $n_k^{(w_i^{(d)})-(d,i)}$, are sums of independent Bernoulli variables,

$$n_k^{(w_i^{(d)})-(d,i)} = \sum_{i\prime \neq i} \mathbf{1}(z_{i\prime}^{(d)} = k) \ , \tag{4.26}$$

where $\mathbf{1}(a)$ is an indicator function which is equal to one if a is true, and zero otherwise. The independence of the Bernoulli variables follows from the mean field assumption. By the central limit theorem, if the Bernoulli variables are numerous enough, the count variables are well approximated by a Gaussian. Teh et al. arrive at Equation 4.25 by evaluating the expectation under the Gaussian approximation, after first approximating the term with a second-order Taylor expansion about the mean of the Gaussian.

CVB0

Asuncion *et al.* (2009) later showed that a simpler version of the CVB method, based on an additional approximation, is much faster and easier to implement while still maintaining its accuracy. This algorithm is derived by dropping the second order information in the Taylor expansion used to derive the approximate update, thereby replacing Equation 4.25 with

$$E_{q_{\neg id}}[\log(n_k^{(d)-(d,i)} + \alpha_k)] \approx \log(E_{q_{\neg id}}[n_k^{(d)-(d,i)}] + \alpha_k) \ . \tag{4.27}$$

The resulting algorithm is referred to by Asuncion *et al.* (2009) as CVB0, since only the "zeroth order" information in the Taylor expansion is used (i.e. the Taylor series is not expanded at all). However, the first order term in the Taylor series is zero, so the approximation used to derive Equation 4.27 can be understood as a first order Taylor series expansion.

By plugging the approximation from Equation 4.27 into the CVB update of Equation 4.24, Asuncion et al. arrive at approximate coordinate ascent updates for each γ_{id},

$$\gamma_{idk} :\propto (N_{dk}^{\Theta\neg id} + \alpha_k)\frac{N_{w_{id}k}^{\Phi\neg id} + \beta_{w_{id}}}{N_k^{Z\neg id} + \sum_w \beta_w} \tag{4.28}$$

with w_{id} corresponding to the word index for the dth document's ith word, and where $a :\propto b$ denotes that a is assigned to be proportional to b. The $\mathbf{N}^Z$, $\mathbf{N}^\Theta$ and $\mathbf{N}^\Phi$ variables, henceforth referred to as the CVB0 statistics, are variational expected counts corresponding to their indices, and the $\neg id$ superscript indicates the exclusion of the current value of γ_{id}. Specifically, $\mathbf{N}^Z$ is the vector of expected number of words assigned to each topic, $\mathbf{N}_d^\Theta$ is the equivalent vector for document d only, and each entry w, k of matrix $\mathbf{N}^\Phi$ is the expected number of times word w is assigned to topic k across the corpus,

$$N_k^Z \triangleq \sum_{id} \gamma_{idk} \qquad N_{dk}^\Theta \triangleq \sum_i \gamma_{idk} \qquad N_{wk}^\Phi \triangleq \sum_{id:w_{id}=w} \gamma_{idk} \ . \tag{4.29}$$

Note that normalizing $\mathbf{N}_d^\Theta + \alpha$ results in a Rao-Blackwellized estimate of the document d's distribution over topics $\theta^{(d)}$, and normalizing $\mathbf{N}_{:,k}^\Phi + \beta$ gives a Rao-Blackwellized estimate of the topic $\Phi^{(k)}$ (Griffiths & Steyvers, 2004; Teh *et al.* , 2007a). Also observe that the update for CVB0 closely resembles the collapsed Gibbs update for LDA (Equation 4.21), but is deterministic.

CVB0 is currently the fastest known technique for LDA inference for single-core batch inference in terms of convergence rate (Asuncion *et al.* , 2009). It is also as simple to implement as collapsed Gibbs sampling, and has a very similar update procedure except that the update is deterministic. Sato & Nakagawa (2012) showed that the terms in the CVB0 update can be understood as optimizing the α-divergence, with different values of α for each term. The α-divergence is a generalization of the KL-divergence that variational Bayes minimizes, and optimizing it is known as power expectation propagation (Minka, 2004). A disadvantage of

CVB0 is that the memory requirements are large as it needs to store a variational distribution γ for every token in the corpus (although this can be improved slightly by "clumping" every occurrence of a specific word in each document together and storing a single γ for them).

4.1.4 Stochastic Optimization

Having discussed variational methods and their application to collapsed LDA, we will now describe stochastic algorithms, which are the final puzzle piece needed to build our approach. Stochastic optimization methods (Robbins & Monro, 1951; Bottou, 1998) are a class of optimization algorithms which can tolerate, or even exploit, randomness. Such randomness may arise from the environment, due to noise in the measurement of the input data. Alternatively, the stochasticity may be introduced deliberately by the algorithm in order to improve optimization performance. In a machine learning context, the latter is an important case, with randomness being used to rapidly approximate expensive update steps in order to scale the algorithms to very large data sets. Stochastic algorithms for machine learning are often called *online learning* algorithms, particularly when applied in a streaming setting (cf. Bottou (1998)). In a typical machine learning application of stochastic algorithms, the learning problem is framed as the minimization of a function

$$g(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{w}) \ , \tag{4.30}$$

where $\mathbf{w}$ is a vector of the parameters of interest, and each $f_{\mathbf{x}_i}(\mathbf{w})$ is a cost function associated with each observed data point $\mathbf{x}_i$, such as the squared error loss for a prediction, minus a log-likelihood, or a variational objective function. To motivate the stochastic approach, let

us first consider instead a standard iterative *batch mode* algorithm for optimizing $g(\mathbf{w})$. Such an algorithm generally has the following form:

- while (*not converged*)
    - Process each input vector $\mathbf{x}_1, \ldots, \mathbf{x}_n$
    - Update the parameters $\mathbf{w}$ based on this processing.

For example, in the gradient descent algorithm, the gradient of $g(\mathbf{w})$ is computed by summing over the gradients for each $f_{\mathbf{x}_i}(\mathbf{w})$. In these batch mode algorithms, the first step in this loop is $O(n)$, where n is the number of data points. Unfortunately, many modern web-scale datasets have very large n (e.g., at the time of writing, the free online encyclopedia Wikipedia has around 14 million articles). In this case, it may take a long time to perform even a single update of the parameters.

To make our algorithms scalable to such datasets, we would prefer to have algorithms whose iteration cost is *independent of* n. This is accomplished by stochastic algorithms, which generally have the form:

- while (*not converged*)
    - Process a small subset of the input vectors
    - Approximate the update step based on this subset
    - Update the parameters $\mathbf{w}$ based on this processing.

This is a stochastic approach, in the sense that randomness arises in the selection of the subset of the input vectors. Typically the approximate update is chosen so that its expectation under the subset selection process is the exact update, and thus the stochastic updates will

take us in the correct direction on average. Thus, stochastic algorithms buy us an iteration running time which does not depend on n, at the cost of introducing noise in the update.

Despite the stochasticity, convergence guarantees exist for most standard stochastic optimization algorithms, as long as the updates are tempered with an appropriate sequence of step sizes (Bottou, 1998; Andrieu *et al.* , 2005). However, the *rate* of convergence with respect to the amount of data processed is provably slower for most stochastic algorithms than their deterministic counterparts. For large data sets, this is a price we are often willing to pay in order to be able to quickly begin making progress when on a computational budget.[6] Furthermore, in a learning context, we are interested in *generalization* performance, rather than in finding the optimal fit to the training data. Stochastic algorithms are often very efficient in the early phase of optimization, which is where the most generalization performance is gained, making them a good fit for learning problems (Bottou & LeCun, 2003). We will consider several examples of stochastic algorithms below.

Robbins-Monro Stochastic Approximation

The original stochastic optimization method is the *stochastic approximation* (SA) algorithm of Robbins & Monro (1951), which aims to find the roots of an equation

$$h(\mathbf{w}) = \mathbf{0} \ , \tag{4.31}$$

in the scenario where we can only observe noisy measurements $y_t(\mathbf{w})$ of it:

$$y_t(\mathbf{w}) = h(\mathbf{w}) + \xi_t, \ E[\xi_t] = 0 \tag{4.32}$$

[6] An exception to the convergence rate limitation of stochastic algorithms is the recently proposed stochastic average gradient (SAG) technique (Le Roux *et al.* , 2012), which maintains the convergence rate of batch mode gradient descent, although it has a memory requirement which is $O(n)$.

at each iteration t. This may occur if $h(\mathbf{w})$ is a function of a physical quantity measured using a noisy instrument. Alternatively, we can select $h(\mathbf{w})$ to be the average over data points $g(\mathbf{w})$ in Equation 4.30. A cheap, noisy measurement of $g(\mathbf{w})$ is obtainable by estimating it based on evaluations of $f_{\mathbf{x}_i}(\mathbf{w})$ at a randomly selected subset of data points, and we can then use the algorithm to find $g(\mathbf{w}) = \mathbf{0}$.

The Robbins-Monro SA algorithm proceeds iteratively, performing the following update after each measurement y_t at iteration t,

$$\mathbf{w} := \mathbf{w} + \rho_t y_t(\mathbf{w}). \tag{4.33}$$

Here, ρ_t is a step size at iteration t, which is typically annealed towards zero. Robbins and Monro showed that this algorithm converges to a correct solution under certain conditions and with an appropriately selected sequence of step sizes. To illustrate the method, we show a simple one-dimensional demonstration in Figure 4.1, where $h(w) = \sin(w)$, measured with standard Gaussian noise added. In this case, the update equation is $w := w + \rho_t(\sin(w) + \xi_t)$, where $\xi_t \sim \text{Gaussian}(0, 1)$ has perturbed our measurement of $\sin(w)$.

It is worth pausing to reflect on the meaning of the update equation for Robbins-Monro SA, while considering our example of the sine function. In each step, the algorithm measures the target function, with some error in the measurement. If the measurement is above zero, the algorithm increases w, although there is no particular reason to suspect that this is the best choice. However, the algorithm makes consistent choices, in that when the measurement is below zero it travels in the opposite direction, and w is decreased. The size of the step that the algorithm takes is smaller if the measurement is closer to zero. Finally, if the measurement is exactly zero, corresponding to having found a solution, the algorithm will not move. The step size is annealed towards zero, allowing it to eventually ignore the noisy readings and settle at a correct solution.

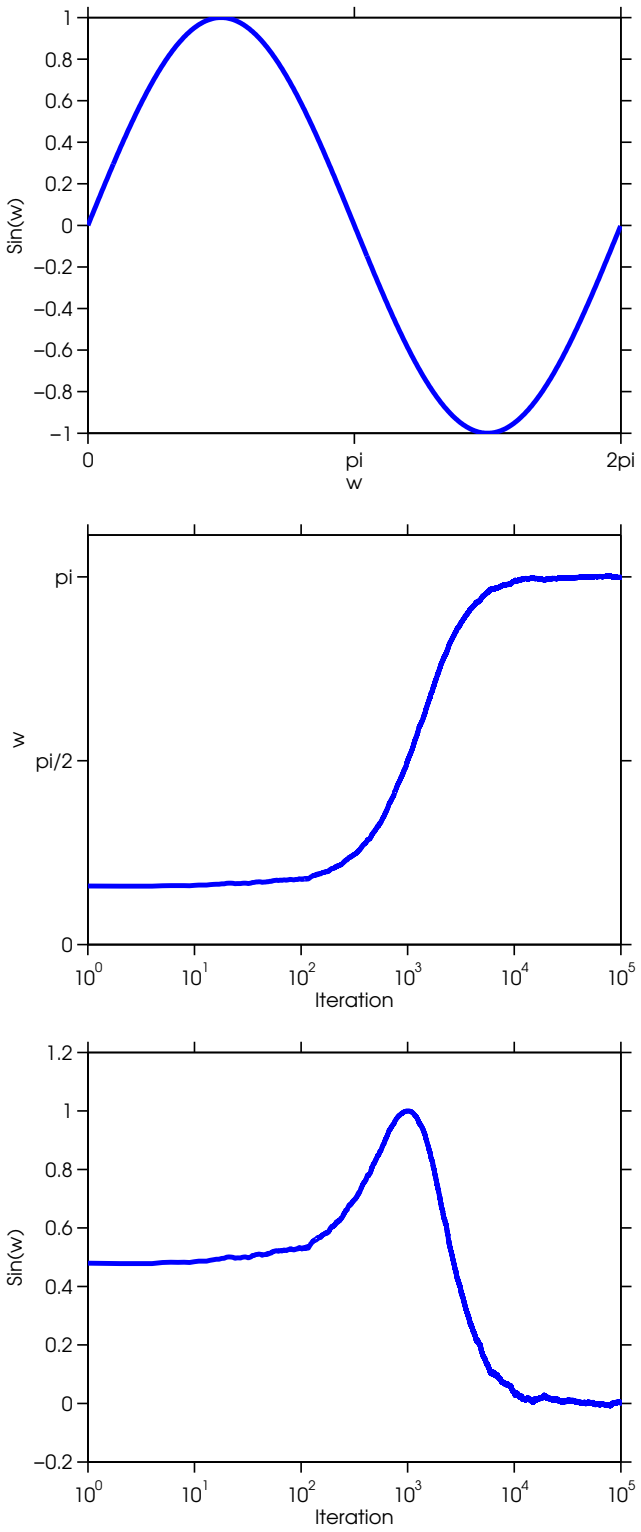


Figure 4.1: Demo of the Robbins-Monro stochastic approximation algorithm. The algorithm finds a root of the sine function at π, despite measuring the function with standard Gaussian noise added to it. **Top**: The sine function. **Middle**: The state of the algorithm w per iteration. **Bottom**: The target function, evaluated per iteration.

A consequence of the update is that if w "overshoots" a zero and lands on a positive region of the curve, it will not return towards the root but instead continue onwards. This is not a problem for the sine function, which has the property that there is always another zero for it to find at a larger value of w. However, if the target is the quadratic function $h(w) = w^2$ and the algorithm overshoots the root at zero, it will diverge. For this reason, a boundedness condition is usually required to ensure convergence (Andrieu *et al.* , 2005).

Stochastic gradient descent

The *stochastic gradient descent* (SGD) method (cf. Bottou (1998)), sometimes referred to in a learning context as *online learning*, is a stochastic variant of gradient descent. The elementary version of this algorithm optimizes $g(\mathbf{w})$ in each iteration t by selecting a random data point z_t, and performing the update

$$\mathbf{w} := \mathbf{w} - \rho_t \nabla_w f_{\mathbf{x}_{z_t}}(\mathbf{w}) \ , \tag{4.34}$$

where $\nabla_w f_{\mathbf{x}_{z_t}}(\mathbf{w})$ is the gradient of $f_{\mathbf{x}_{z_t}}(\mathbf{w})$ with respect to $\mathbf{w}$. Importantly, the average of the gradients selected by this procedure is the gradient of the full objective function, $E_z[\nabla_w f_{\mathbf{x}_{z_t}}(\mathbf{w})] = \nabla g(\mathbf{w})$.

In a more general framework, the objective function can instead be set to be $E_z[f_z(\mathbf{w})]$ directly, for some distribution of events z. This means that we do not have to define $g(\mathbf{w})$ over a finite set of data points, allowing the algorithm to be used in a streaming setting, with an infinite number of incoming data points. Furthermore, any unbiased estimate $H(z, \mathbf{w})$ of $E_z[\nabla_w f_z(\mathbf{w})]$ may be used as a noisy gradient in each iteration, such as a *minibatch* estimate computed over multiple data points.

In this general setting, the update of the algorithm becomes

$$\mathbf{w} := \mathbf{w} - \rho_t H(z_t, \mathbf{w}) \ . \tag{4.35}$$

The SGD algorithm is guaranteed to converge to a local minimum, for an appropriate sequence of step sizes ρ (Bottou, 1998). Of course, the algorithm can also be used to solve maximization problems by reversing the sign of the update,

$$\mathbf{w} := \mathbf{w} + \rho_t H(z_t, \mathbf{w}) \ . \tag{4.36}$$

To illustrate the method with an example, let us return to the case of the sine function, which we previously explored in Figure 4.1. The derivative of $\sin(x)$ is $\cos(x)$. Suppose we can measure the derivative with standard Gaussian noise added to it, and we are interested in maximizing over the function. Then the update becomes $w := w + \rho_t(\cos(w) + \xi_t)$, where $\xi_t \sim \text{Gaussian}(0, 1)$. The behavior of the algorithm is demonstrated in Figure 4.2.

Interestingly, the update $w := w + \rho_t(\cos(w) + \xi_t)$ is identical to the update step of a Robbins-Monro SA algorithm for finding zeros of $\cos(w)$. This is not a coincidence. By comparing Equation 4.36 and Equation 4.33, we can see that stochastic gradient ascent is a Robbins-Monro algorithm for finding the zeros of the gradient, i.e. the stationary points of the objective function. Similarly, the descent version of the algorithm in Equation 4.35 finds the zeros of the negative of the gradient, which are also the stationary points, however the Robbins-Monro algorithm takes steps in the reverse direction to the ascent algorithm.

As an aside, the name "stochastic gradient descent" is arguably a misnomer, because the stochastic algorithm is no longer a *descent* method, as the random behavior may cause the cost function to increase in some cases.[7]

[7]This observation was made by Stephen Wright, in a lecture at the IPAM Stochastic Gradient Methods Workshop, held at UCLA in February 2014. As an organizer of the workshop, Wright changed the name of the workshop from its original title, "Stochastic Gradient Descent Workshop" for this reason.

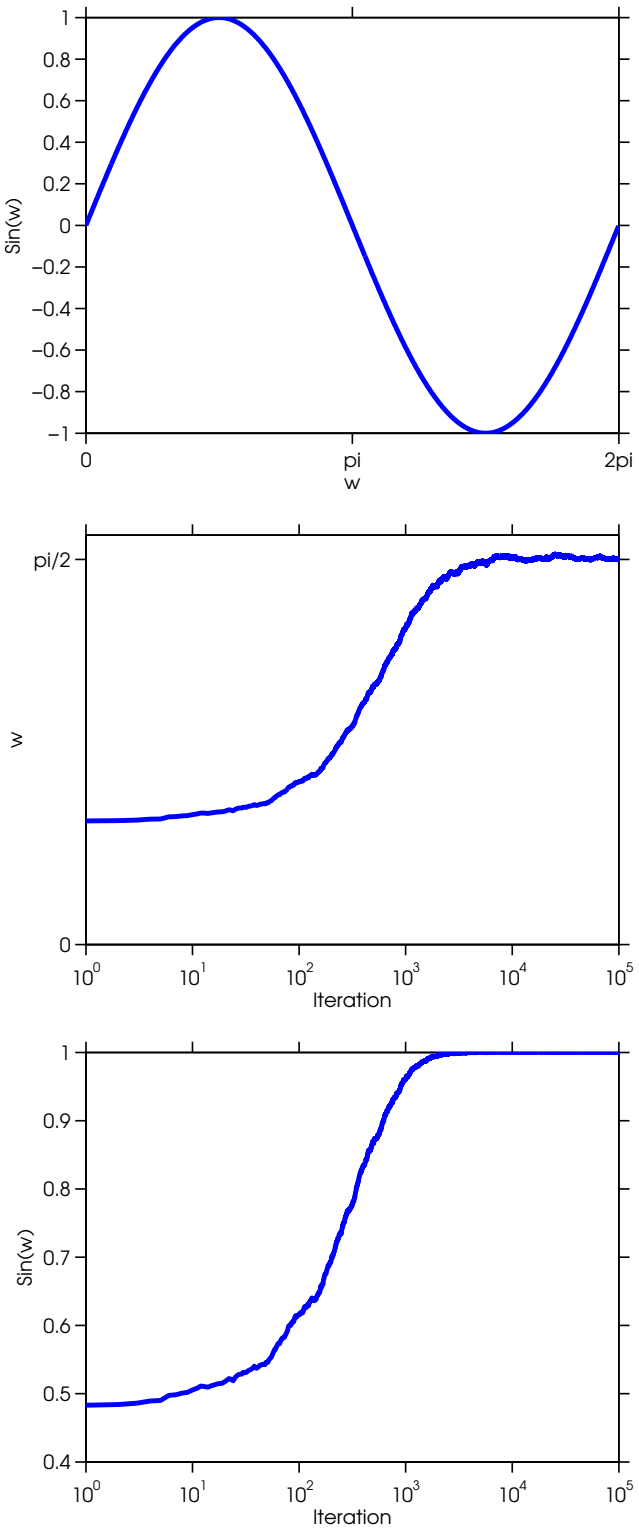


Figure 4.2: Demo of the stochastic gradient algorithm. The algorithm finds a maximum of the sine function at $\pi/2$, despite measuring the derivative with standard Gaussian noise added to it. **Top**: The sine function. **Middle**: The state of the algorithm w per iteration. **Bottom**: The objective function, evaluated per iteration.

Stochastic Variational Inference

A stochastic algorithm for performing variational inference on very large data sets was proposed by Hoffman *et al.* (2013). This algorithm applies to graphical models whose parameters can be split into "global" parameters $\mathbf{G}$ and "local" parameters $\mathbf{L}_j$ pertaining to each data point $\mathbf{x}_j$, and whose complete conditional distributions for each variable are exponential family distributions. The LDA model is an important example of a model with this property, and indeed the stochastic variational inference algorithm was originally designed specifically for LDA (Hoffman *et al.* , 2010).

The stochastic VB algorithm proceeds as follows in each iteration. First, it examines one randomly selected data point and optimizes that data point's local variational parameters (such as $\theta^{(j)}$ in LDA). It then updates the global variational parameters, such as topics $\Phi^{(k)}$, via a stochastic estimate of the *natural gradient*. In the field of information geometry, the natural gradient is an alternative to the usual gradient which gives the steepest ascent direction according to Riemannian geometry instead of Euclidean geometry (Amari, 1998). In the case of stochastic VB, the stochastic natural gradient update ends up being a simple convex combination of the current global parameters and an estimate of the global parameters based on current data point. We can think of the algorithm as performing stochastic gradient descent, but using the natural gradient instead of the usual Euclidean one. The general scheme of stochastic VB is given in Algorithm 6.

For an appropriate local update and sequence of step sizes ρ, this algorithm is guaranteed to converge to the optimal variational solution (Hoffman *et al.* , 2013). In the case of LDA, let λ_k be the parameter vector for a variational Dirichlet distribution on topic $\Phi^{(k)}$. For each document j, the method computes variational distributions for both the topic assignments and the document's distribution over topics using regular VB updates. These values are then used to update the topics. Specifically, for each topic k the algorithm computes $\hat{\lambda}_k$,

Algorithm 6 Stochastic variational inference (Hoffman et al.)

- Input: Data points $\mathbf{x}_1, \ldots, \mathbf{x}_D$ (e.g. word count histograms for documents), step sizes ρ_t, $t = 1 : m$ (where m is the maximum number of iterations)
- Randomly initialize "global" (e.g. topic) parameters G
- For $t = 1 : m$
  - Select a random data point $\mathbf{x}_j$, $j \in \{1, \ldots, D\}$
  - Compute "local" (e.g. document-level) variational parameters $\mathbf{L}_j$
  - $\hat{\mathbf{G}} = D\mathbf{L}_j$
  - $\mathbf{G} := (1 - \rho_t)\mathbf{G} + \rho_t \hat{\mathbf{G}}$

an estimate of what λ_k would be if all D documents were identical to document j. The algorithm then updates the λ_k's via a natural gradient update, which takes the form

$$\lambda_k := (1 - \rho_t)\lambda_k + \rho_t \hat{\lambda}_k \ . \tag{4.37}$$

The form of this update provides another insight into the advantages of the stochastic approach. In the standard VB algorithm for LDA, the topics are updated after a complete pass through the data, by summing up the local parameters $\mathbf{L}_j$. However, the parameters are initialized randomly, and so in the early iterations these values will not be very good. At the end of the iteration, the topics are finally updated, using an aggregation of these largely random values. It takes a number of iterations to bootstrap away from the random initialization. On the other hand, the online LDA algorithm updates the topics after every document, meaning that it has an earlier chance to improve on the random initialization. Furthermore, the update in Equation 4.37 uses the step size ρ_t as a "forgetting rate," allowing it to overwrite the old, poorly estimated values with new improved information. Typically the step size is set so that the forgetting rate is high in the beginning, allowing it to quickly escape from the random initialization.

Online EM

In a somewhat broader context, the online EM algorithm of Cappé & Moulines (2009) is another general-purpose method for learning latent variable models in an online setting. Suppose there are independent and identically distributed observed variables $\mathbf{x}^{(i)}$, each associated with latent variables $\mathbf{z}^{(i)}$ and we are interesting in performing maximum likelihood estimation over parameters θ. It is assumed that the complete data likelihood can be written in exponential family form,

$$Pr(\mathbf{x}, \mathbf{z}|\theta) \propto h(\mathbf{x}, \mathbf{z}) \exp(S(\mathbf{x}, \mathbf{z})^\intercal \eta(\theta)) \ , \tag{4.38}$$

where $S(\mathbf{x}, \mathbf{z})$ maps the observed and hidden variables to a vector of sufficient statistics, and $\eta(\theta)$ is a vector-valued function of θ. In each iteration, a new data point $\mathbf{x}^{(i)}$ is observed. The online EM algorithm consists of an E-step and an M-step, as in traditional EM (Dempster *et al.* , 1977). However, these steps are performed *for each data point* $\mathbf{x}^{(i)}$ instead of involving a pass through the entire data set.

The E-step operates on a running estimate $\mathbf{s}$ of the expected value of the complete data sufficient statistics $E_{Pr(\mathbf{z}|\mathbf{x},\theta)}[S(\mathbf{x}, \mathbf{z})]$. A stochastic estimate $\hat{\mathbf{s}}(\mathbf{x}^{(i)}; \theta) = E_{Pr(\mathbf{z}^{(i)}|\mathbf{x}^{(i)},\theta)}[S(\mathbf{x}^{(i)}, \mathbf{z}^{(i)})]$ is made based on the current data point $\mathbf{x}^{(i)}$, and then the running estimate $\mathbf{s}$ is updated via an online average,

$$\mathbf{s} := (1 - \rho_t)\mathbf{s} + \rho_t \hat{\mathbf{s}}(\mathbf{x}^{(i)}; \theta) \ . \tag{4.39}$$

Finally, a standard M-step is performed, which maximizes the EM lower bound $E_{Pr(\mathbf{z}|\mathbf{x},\theta)}[\log Pr(\mathbf{x}, \mathbf{z})]$, and hence the log-likelihood, with respect to parameters θ. Due to the exponential family assumption, it is possible to perform the M-step update based on the estimated expected sufficient statistics $\mathbf{s}$.

4.2 Stochastic CVB0

Our goal in this chapter is to develop a fast, scalable and accurate algorithm for training topic models on large data sets. Given the discussion above, a desirable algorithm for training LDA on web-scale corpora should have the following properties, each of which has been attained by at least one algorithm for LDA in the literature:

1. An efficient, simple update.
   (*Collapsed Gibbs sampling, CVB0*)

2. The improved variational bound of the collapsed representation.
   (*CVB, CVB0. Not applicable to sampling approaches*)

3. An update whose execution time does not depend on the size of the dataset.
   (*Collapsed Gibbs sampling, CVB0, stochastic VB*)

4. Memory requirements that do not depend on the size of the dataset.
   (*Stochastic VB*)

5. The ability to quickly "forget" the random initialization.
   (*Stochastic VB*)

6. The ability to estimate the topics when only a subset of the data have been visited.
   (*Stochastic VB*)

The first three conditions are met by the CVB0 algorithm of Asuncion *et al.* (2009), with condition 3 holding after caching the CVB0 statistics, and notwithstanding a $O(N)$ initialization step. However, being a batch algorithm, it fails on conditions 4 – 6. This is because it stores variational parameters γ for every word in every document, and these are updated one at a time, with the updates being initially confounded by the original random values of

the other γ's. Fortunately, the remaining conditions 4 – 6 (as well as 3) are key properties of stochastic optimization methods.

In this chapter, we show how to gain all of these properties in a single algorithm, *stochastic CVB0 (SCVB0)*, which performs stochastic variational inference in the collapsed representation of LDA. The proposed algorithm is inspired by the ideas of both stochastic VB and online EM. The algorithm can also be understood both as a Robbins-Monro SA algorithm and as an online EM algorithm, and we will make use of these connections to demonstrate the convergence of the algorithm. For reference, the set of notation used to describe the SCVB0 algorithm is provided in Table 4.2.

4.2.1 Estimating the CVB0 Update

Before building our new stochastic algorithm, let us pause to consider the anatomy of typical stochastic algorithms for machine learning. These techniques begin with a batch-mode algorithm, and modify it to *estimate the update* which the batch algorithm would take, based on subsets of the data, in order to free the update from dependence on the full data set. For example, stochastic gradient descent estimates the gradient of the objective function based on one or more data points, and then takes a step in the direction of minus the estimated gradient.

We will take this approach to make a stochastic version of the CVB0 algorithm. Specifically, we would like to be able to *estimate the CVB0 update* from subsets of the data. Recall the form of the CVB0 update from Equation 4.28:

$$\gamma_{idk} :\propto (N_{dk}^{\Theta \neg id} + \alpha_k) \frac{N_{w_{id}k}^{\Phi \neg id} + \beta_{w_{id}}}{N_k^{Z \neg id} + \sum_w \beta_w} \ .$$

We can see that the CVB0 statistics $\mathbf{N}^Z$, $\mathbf{N}^\Theta$ and $\mathbf{N}^\Phi$, terms which are dependent on the entire dataset, are what is needed to perform a CVB0 update. Thus, they are good candidates for being estimated stochastically based only on the subset of tokens we have observed. They are also sufficient to recover a Rao-Blackwellized estimate of the topics, as in Griffiths & Steyvers (2004) and Teh *et al.* (2007a). An algorithm which stochastically estimates the CVB0 statistics will thereby obtain the desired properties 5 and 6. This follows from being unencumbered of a reliance on the full dataset, and specifically on the initial random values for the γ's associated with tokens which we may not even have examined yet. If the algorithm does not maintain the γ's, it will also gain property 4, and so it will have all the properties which we have identified as necessary for scalability. In any case, it would not make sense to maintain the γ's when using stochastic estimates of the CVB0 statistics, as the stochastic estimates would not match the values implied by the γ's.

This strategy of performing stochastic estimation of the CVB0 statistics, in order to perform a CVB0 update, is reminiscent of the online EM algorithm. That algorithm also estimates a set of statistics, namely the expected sufficient statistics of the E-step of EM, in order to perform an M-step update. We place this approach in the context of the other stochastic algorithms in Table 4.3.

4.2.2 Estimating the CVB0 Statistics

As we have seen, in order to estimate the CVB0 update based on a subset of the data, the task is to estimate the CVB0 statistics. Following Bottou (1998), these estimates should be unbiased. Specifically, if we use some sampling distribution $f(\mathbf{x})$ to select a subset of the data $\mathbf{x}$, and then estimate a statistic s of the full dataset (such as a CVB0 statistic or the gradient) via an estimator $H(\mathbf{x})$, then we desire that $E_{f(\mathbf{x})}[H(\mathbf{x})] = s$.

K	Number of topics
D	Number of documents
C	Number of words in corpus
W	Size of dictionary
C_d	Number of words in document d
$z_i^{(d)}$	Topic for (i, d), the ith word of the dth document
$w_i^{(d)}$	Dictionary index for word (i, d)
$\theta^{(d)}$	Distribution over topics for document d, $1 \times K$
α	Dirichlet prior parameters for θ, $1 \times K$
$\Phi^{(k)}$	Distribution over words for topic k, $W \times 1$
β	Dirichlet prior parameters for Φ, $W \times 1$
γ_{id}	Variational distribution for word (i, d), $1 \times K$
$\mathbf{N}^{\Theta}$	Expected topic counts per document, $D \times K$
$\mathbf{N}^{\Phi}$	Expected topic counts per word, $W \times K$
$\mathbf{N}^{Z}$	Expected topic counts overall, $1 \times K$
$\mathbf{Y}^{(id)}$	Estimate of $\mathbf{N}^{\Phi}$ based only on word (i, d), $W \times K$
M	Minibatch, a set of documents
$\hat{\mathbf{N}}^{\Phi}$	Estimate of $\mathbf{N}^{\Phi}$ from current minibatch, $W \times K$
$\hat{\mathbf{N}}^{Z}$	Estimate of $\mathbf{N}^{Z}$ from current minibatch, $1 \times K$
ρ_t^{Θ}	Step size for $\mathbf{N}^{\Theta}$ at timestep t
ρ_t^{Φ}	Step size for $\mathbf{N}^{\Phi}$ and $\mathbf{N}^{Z}$ at timestep t
w_{ad}	Dictionary index for ath distinct word of d
m_{ad}	Count of ath distinct word of d
γ_{ad}	Variational distribution for ath distinct word of d, $1 \times K$

Table 4.2: Summary of notation for the SCVB0 algorithm.

Algorithm	**Stochastically Estimated Update Term**
Robbins-Monro SA	Value of the target function
SGD	Gradient of the target function
Stochastic Variational Inference	Natural gradient of the target function
Online EM	E-step sufficient statistics
SCVB0	CVB0 statistics

Table 4.3: The terms of the corresponding deterministic update equations that each stochastic algorithm estimates.

In the context of our algorithm, suppose we have seen a token $w_i^{(d)}$, and its associated γ_{id}. The information this gives us about the statistics depends on how the token was drawn. If the token was drawn uniformly at random from all of the tokens in the corpus, then

$$E[C\gamma_{id}] = \mathbf{N}^Z \ , \tag{4.40}$$

where C is the number of words in the corpus, and the expectation is with respect to the sampling distribution. Therefore, we can use $C\gamma_{id}$ as a stochastic estimate of $\mathbf{N}^Z$. Similarly, for the same sampling procedure, to estimate the word-topic expected counts matrix we have

$$E[C\mathbf{Y}^{(id)}] = \mathbf{N}^\Phi \ , \tag{4.41}$$

where $\mathbf{Y}^{(id)}$ is a $W \times K$ matrix with the $w_i^{(d)}$th row being γ_{id} and with zeros in the other entries. Now if the token was drawn uniformly from the tokens in document d,

$$E[C_d\gamma_{id}] = \mathbf{N}_d^\Theta \ , \tag{4.42}$$

where C_d is the length of document d.[8] So with these sampling procedures, we can use $C\gamma_{id}$, $C\mathbf{Y}^{(id)}$ and $C_d\gamma_{id}$ as stochastic estimates of $\mathbf{N}^Z$, $\mathbf{N}^\Phi$ and $\mathbf{N}_d^\Theta$, respectively.

4.2.3 The SCVB0 Algorithm

Since we may not maintain the γ's, we cannot perform these sampling procedures directly. However, with a current guess at the CVB0 statistics we can *update* a token's variational distribution using the CVB0 update equation, and observe its new value. We can then use this γ_{id} to improve our estimate of the CVB0 statistics. This suggests an iterative procedure,

[8] Other sampling schemes are possible, which would lead to different algorithms. For example, one could sample from the set of tokens with word index w to estimate $\mathbf{N}_w^\Phi$. Our choice leads to an algorithm that is practical in the online setting.

alternating between a "maximization" step, approximately optimizing the evidence lower bound with respect to a particular γ_{id} via CVB0, and an "expectation" step, where we update the expected count statistics to take into account the new γ_{id}. As the algorithm continues, the γ_{id}'s we observe will change, so we cannot simply average them. Instead, we can follow the strategy of Cappé & Moulines (2009) in the online EM algorithm and perform an online average of these statistics as in Equation 4.39.

More specifically, in the proposed SCVB0 algorithm we process the corpus one token at a time, examining the tokens from each document in turn. For each token, we first compute a new γ_{id}. We do not maintain the γ's, but compute (updated versions of) them as needed via CVB0. This means we must make a small additional approximation in that we cannot subtract current values of γ_{id} in Equation 4.28. With large corpora and large documents this difference is negligible. The update becomes

$$\gamma_{idk} :\propto (N^{\Theta}_{dk} + \alpha_k)\frac{N^{\Phi}_{w_i^{(d)}k} + \beta_{w_i^{(d)}}}{N^{Z}_{k} + \sum_w \beta_w} \ . \tag{4.43}$$

We then use this to re-estimate our CVB0 statistics. While we are processing randomly ordered tokens i of document d, we are effectively drawing random tokens from it, so we can stochastically estimate $\mathbf{N}^{\Theta}_d$ by $C_d\gamma_{id}$. We update $\mathbf{N}^{\Theta}_d$ with an online average of the current value and its estimated value,

$$\mathbf{N}^{\Theta}_d := (1-\rho^{\Theta}_t)\mathbf{N}^{\Theta}_d + \rho^{\Theta}_t C_d\gamma_{id} \ , \tag{4.44}$$

where ρ^{Θ}_t is a step size. We use one sequence of step-sizes ρ^{Φ} for $\mathbf{N}^{\Phi}$ and $\mathbf{N}^{Z}$, and another sequence ρ^{Θ} for $\mathbf{N}^{\Theta}$. Although we process one document at a time, we eventually process all of the words in the corpus. So for the purposes of updating $\mathbf{N}^{\Phi}$ and $\mathbf{N}^{Z}$, in the long-run the algorithm is effectively drawing tokens from the entire corpus. As per Section 4.2.2, we can estimate $\mathbf{N}^{\Phi}$ after observing one γ_{id} as $C\mathbf{Y}^{(id)}$, and we can estimate $\mathbf{N}^{Z}$ as $C\gamma_{id}$. Then,

the updates are once again a convex combination of current and estimated values for the statistics,

$$\mathbf{N}^{\Phi} := (1 - \rho_t^{\Phi})\mathbf{N}^{\Phi} + \rho_t^{\Phi} C\mathbf{Y}^{(id)} \tag{4.45}$$

$$\mathbf{N}^{Z} := (1 - \rho_t^{\Phi})\mathbf{N}^{Z} + \rho_t^{\Phi} C\gamma_{id} \ . \tag{4.46}$$

The basic SCVB0 algorithm consists of iterating Equations 4.43 – 4.46 over all of the tokens in all of the documents, with the order of the documents and of the words in each document assumed to be shuffled. The γ's are not maintained, and the $\mathbf{N}^{\Theta}$'s are discarded after a document is processed, so the memory requirements do not depend on the size of the corpus.

4.2.4 Extra Refinements

In practice, it is too expensive to update the entire $\mathbf{N}^{\Phi}$ after every token, as this requires copying a $W \times K$ matrix. This suggests the use of minibatches over tokens, to reduce the frequency at which this update is performed. The estimated $\mathbf{N}^{\Phi}$ after observing a minibatch M is the average of the per-token estimates, and similarly for $\mathbf{N}^{Z}$, leading to the updates:

$$\mathbf{N}^{\Phi} := (1 - \rho_t^{\Phi})\mathbf{N}^{\Phi} + \rho_t^{\Phi} \hat{\mathbf{N}}^{\Phi} \tag{4.47}$$

$$\mathbf{N}^{Z} := (1 - \rho_t^{\Phi})\mathbf{N}^{Z} + \rho_t^{\Phi} \hat{\mathbf{N}}^{Z} \tag{4.48}$$

where $\hat{\mathbf{N}}^{\Phi} = \frac{C}{|M|} \sum_{id \in M} \mathbf{Y}^{(id)}$ and $\hat{\mathbf{N}}^{Z} = \frac{C}{|M|} \sum_{id \in M} \gamma_{id}$. Minibatch updates are also possible for the $\mathbf{N}^{\Theta}$'s, however we chose to avoid this in order to more rapidly update these parameters. Depending on the lengths of the documents and the number of topics, it may also be beneficial to perform a small number of extra passes to learn the document statistics before updating the topic statistics. We found empirically that one such burn-in pass was sufficient in all of the datasets we tried in our experiments. Pseudo-code for the algorithm, which we

Algorithm 7 Stochastic CVB0

- Randomly initialize $\mathbf{N}^{\Phi}$, $\mathbf{N}^{\Theta}$; $\mathbf{N}^{Z} := \sum_w \mathbf{N}^{\Phi}_w$
- For each minibatch M
  - $\hat{\mathbf{N}}^{\Phi} := \mathbf{0}$; $\hat{\mathbf{N}}^{Z} := \mathbf{0}$
  - For each document d in M
    - For zero or more "burn-in" passes
      - For each token i
        - Update γ_{id} (Equation 4.43)
        - Update $\mathbf{N}^{\Theta}_d$ (Equation 4.44)
    - For each token i
      - Update γ_{id} (Equation 4.43)
      - Update $\mathbf{N}^{\Theta}_d$ (Equation 4.44)
      - $\hat{\mathbf{N}}^{\Phi}_{w_i^{(d)}} := \hat{\mathbf{N}}^{\Phi}_{w_i^{(d)}} + \frac{C}{|M|}\gamma_{id}$
      - $\hat{\mathbf{N}}^{Z} := \hat{\mathbf{N}}^{Z} + \frac{C}{|M|}\gamma_{id}$
  - Update $\mathbf{N}^{\Phi}$ (Equation 4.47)
  - Update $\mathbf{N}^{Z}$ (Equation 4.48)

refer to as "Stochastic CVB0" (SCVB0) is given in Algorithm 7. It should be noted that the algorithm is relatively straightforward to implement, as it consists of just a few update rules involving simple arithmetic operations.

An optional additional optimization of the above algorithm is to only perform one update for each distinct token in each document, and scale the update by the number of copies in the document. This process, often called "clumping," is standard practice for fast implementations of all LDA inference algorithms (e.g. see Teh *et al.* (2007a) and Jonathan Chang's R package for LDA)[9], though it is only exact for uncollapsed algorithms, where the $z_i^{(d)}$'s are D-separated by $\theta^{(d)}$.

[9] `http://cran.r-project.org/web/packages/lda/`

Suppose we have observed w_{ad}, which occurs m_{ad} times in document d. Plugging Equation 4.44 into itself m_{ad} times and noticing that all but one of the resulting terms form a geometric series, we find that performing m_{ad} updates for $\mathbf{N}_d^\Theta$ while holding γ_{ad} fixed is equivalent to

$$\mathbf{N}_d^\Theta := (1-\rho_t^\Theta)^{m_{ad}}\mathbf{N}_d^\Theta + C_d\gamma_{ad}(1-(1-\rho_t^\Theta)^{m_{ad}}) \ . \tag{4.49}$$

4.3 Experiments

This section describes an experimental analysis of the proposed SCVB0 algorithm, with direct comparison to the stochastic variational Bayes algorithm of Hoffman et al., hereafter referred to as SVB. As well as performing an analysis on several large-scale problems, we also investigate the effectiveness of the stochastic LDA inference algorithms in terms of learning topics in near real-time on small corpora.

4.3.1 Large-Scale Experiments

We studied the performance of the algorithms on three large data sets. The corpora are:

- *PubMed Central*: A corpus of full-text scientific articles from the open-access PubMed Central database of scientific literature in the biomedical and life sciences.[10] After processing to remove stopwords and words occurring less than 300 times, the corpus contained approximated 320M tokens across 165,000 articles, with a vocabulary size of around 38,500 words.

- *New York Times*: A corpus containing 1.8 million articles from the New York Times, published between 1987 and 2007. After removing stopwords and words occurring less

[10] `http://www.ncbi.nlm.nih.gov/pmc/tools/ftp/`

than 500 times, the corpus had a dictionary of about 50,000 words and contained 475M distinct tokens.

- *Wikipedia*: This collection contains 4.6 million articles from the online encyclopedia Wikipedia. We used the dictionary of 7,700 words extracted by Hoffman et al. for their experiments on an earlier extracted Wikipedia corpus. There were 811M tokens in the corpus.

Comparison to SVB

We explored predictive performance versus wall-clock time for both SCVB0 and SVB. To compare the algorithms fairly, we implemented both of them in the fast high-level language *Julia* (Bezanson *et al.* , 2012). Our implementation of SVB closely follows the python implementation provided by Hoffman, and has several optimizations not mentioned in the original paper including handling the latent topic assignments $\mathbf{z}$ implicitly, "clumping" of like tokens, and sparse updates of the topic matrix. The SCVB0 algorithm was implemented as it is written in Algorithm 7, using the clumping optimization but with no additional algorithmic optimizations. Specifically, neither implementation used the complicated optimizations taking advantage of sparsity that are exploited by the Vowpal Wabbit implementation of SVB[11] and in the variant of SVB proposed by Mimno *et al.* (2012). Instead, our implementations represent a "best-effort" attempt to implement each algorithm efficiently yet following the spirit of the original pseudo-code.

In all experiments, each algorithm was trained with 200 topics, using minibatches of size 100. We used a step-size schedule of $\frac{s}{(\tau+t)^\kappa}$ for document iteration t, with $s = 10$, $\tau = 1000$ and $\kappa = 0.9$. For SCVB0, the document parameters were updated using the same schedule with $s = 1$, $\tau = 10$ and $\kappa = 0.9$, with t referring to the word iteration of the current document. We

[11] `https://github.com/JohnLangford/vowpal_wabbit/wiki`

used LDA hyper-parameters $\alpha = 0.1$ and $\beta = 0.01$ for SCVB0. For SVB, we tried both these same hyperparameter values as well as shifting by 0.5 as recommended by Asuncion *et al.* (2009) to compensate for the implicit bias in how uncollapsed VB treats hyper-parameters. We used a single pass to learn document parameters for SCVB0, and tried both a single pass and five passes for SVB.

For each experiment we held out 10,000 documents and trained on the remaining documents. We split each test document in half, estimated document parameters on one half and computed the log-probability of the remaining half of the document. Figures 4.3 through 4.5 show held-out log-likelihood versus wall-clock time for each algorithm. In the figures, SVB-Bx-Oy corresponds to running SVB with x "burn-in" passes per document and with hyper-parameters offset from $\alpha = 0.1$ and $\beta = 0.01$ by y.

For the PubMed Central data, we found that all algorithms perform similarly after about an hour, but prior to that SCVB0 is better, indicating that SCVB0 makes better use of its time. All algorithms perform similarly per-iteration (see Figure 4.6), but SCVB0 is able to benefit by processing more documents in the same amount of time. The per-iteration plots for the other datasets were similar.

Our results find that SCVB0 shows a more substantial benefit when employed on larger datasets. For both the New York Times and Wikipedia datasets (which are each significantly larger than the PubMed Central dataset in terms of the number of documents), SCVB0 converged to a better solution than SVB for any of its parameter settings. Furthermore, SCVB0 outperforms SVB throughout the run. The superior performance of SCVB0 over the uncollapsed SVB method is consistent with the fact that the variational bound for the collapsed representation is strictly better than the bound for the uncollapsed representation of LDA (Teh *et al.* , 2007a).

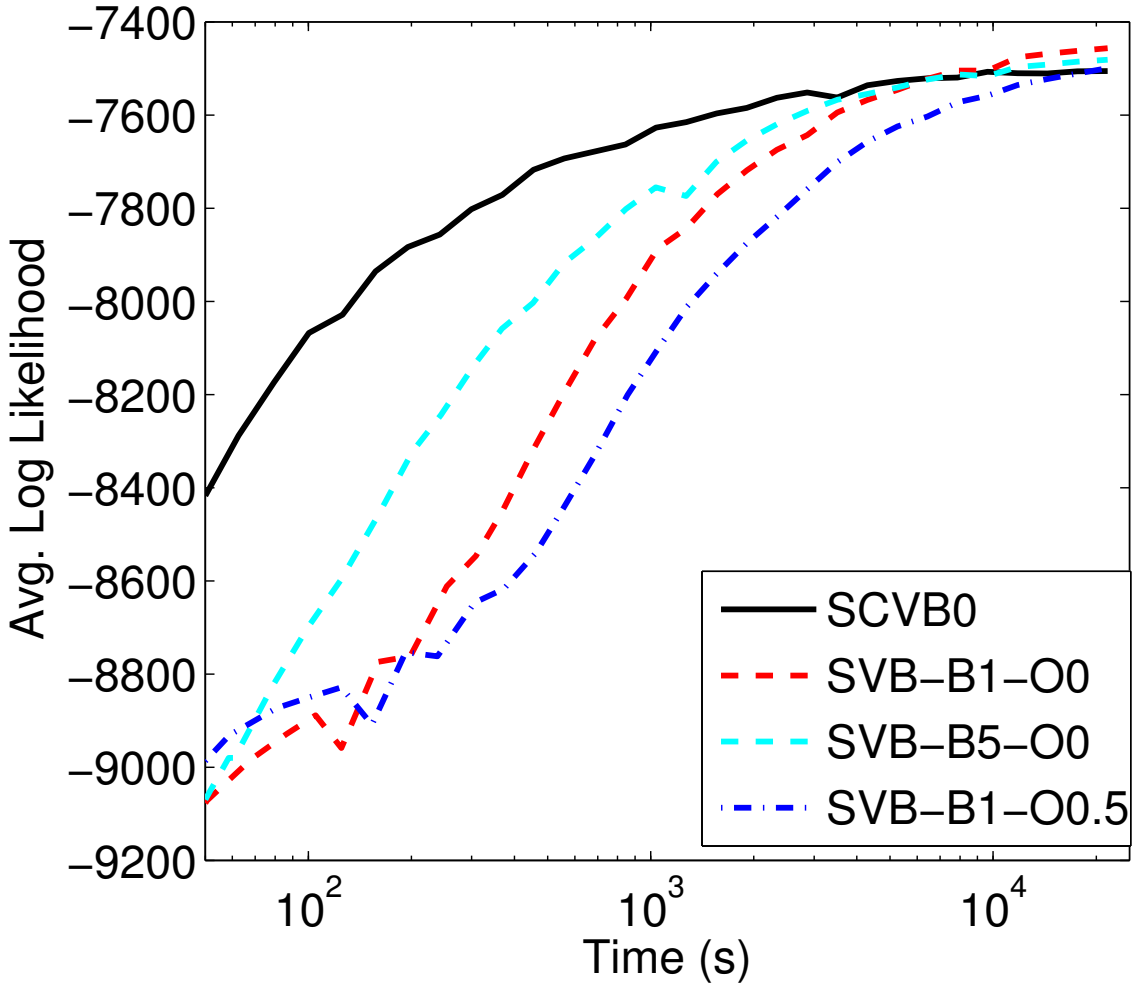


Figure 4.3: Log-likelihood vs time for the PubMed Central experiments. SVB-Bx-Oy corresponds to running SVB with x "burn-in" passes per document and with hyper-parameters offset from $\alpha = 0.1$ and $\beta = 0.01$ by y.

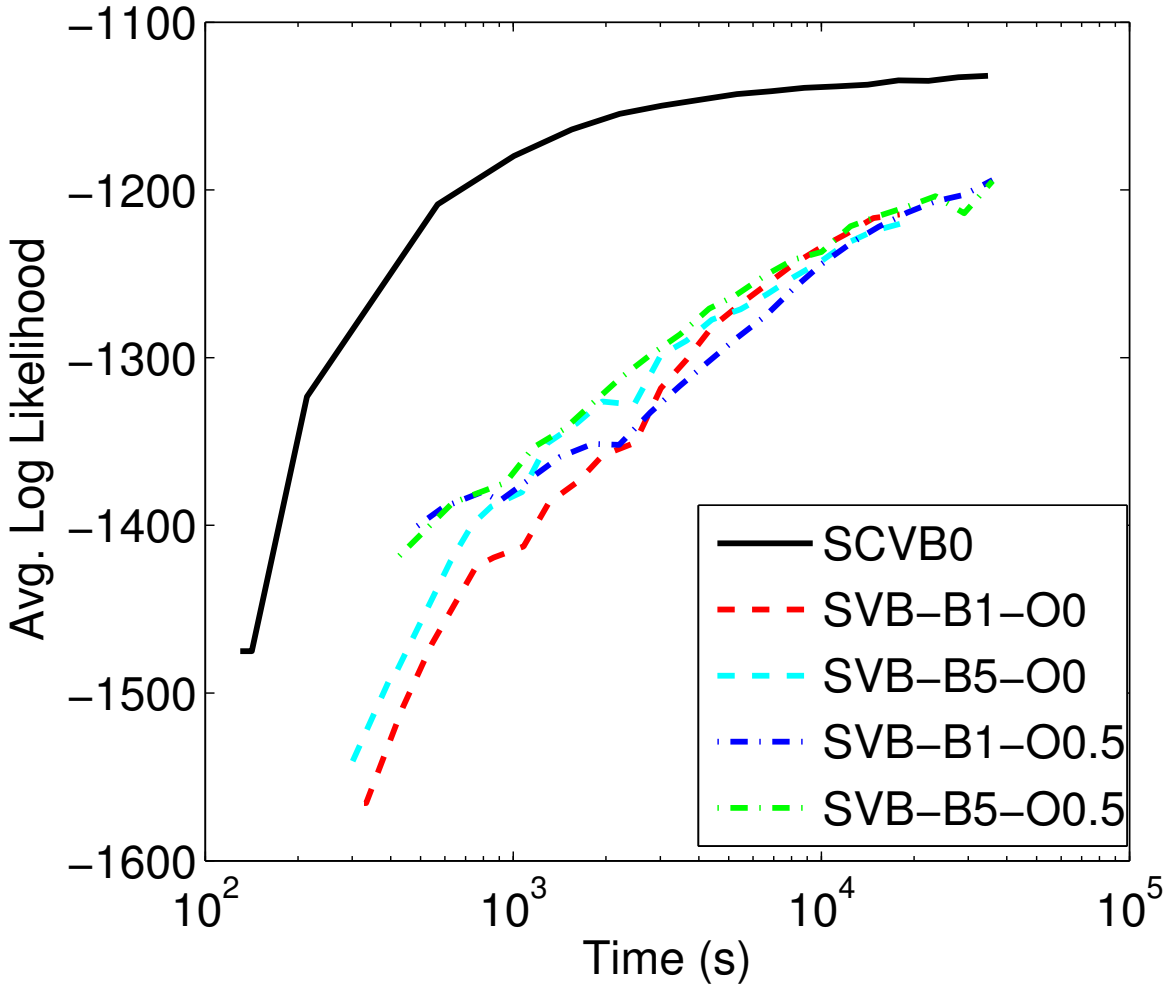


Figure 4.4: Log-likelihood vs time for the New York Times experiments.

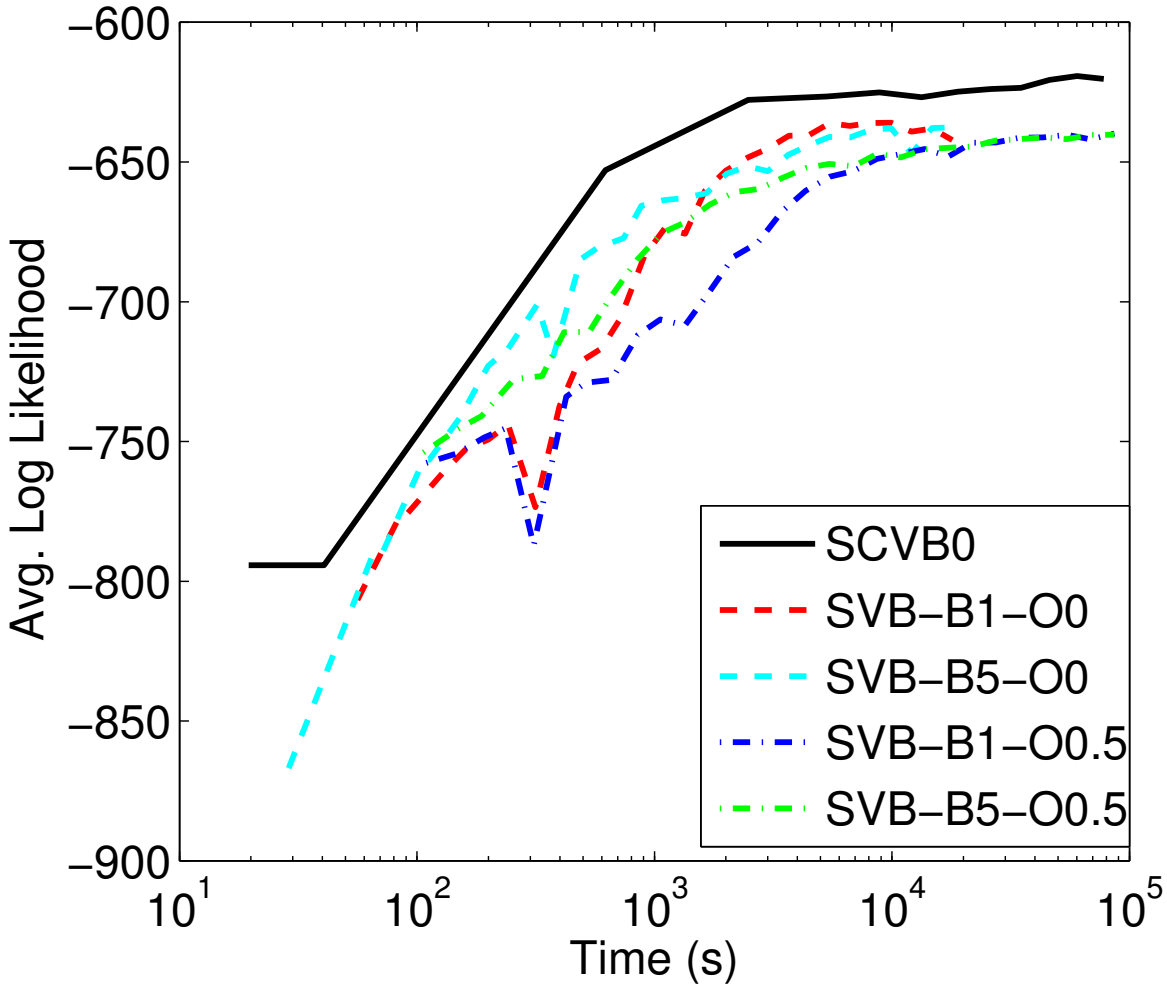


Figure 4.5: Log-likelihood vs time for the Wikipedia experiments.

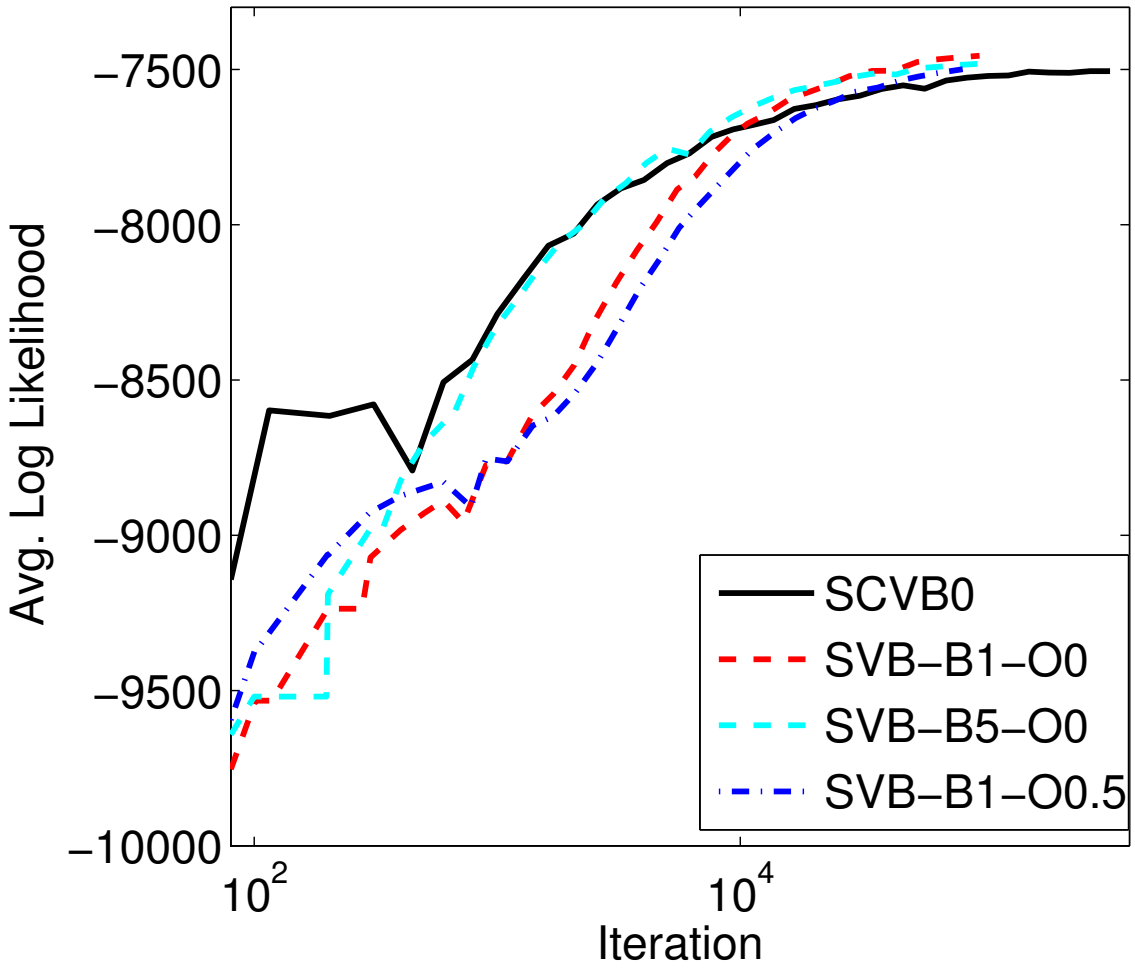


Figure 4.6: Log-likelihood vs iteration for the PubMed Central experiments.

Comparison to Batch VB

We also compared SCVB0 to the batch VB algorithm on the Wikipedia dataset (Figure 4.7); other standard batch algorithms such as Gibbs sampling tend to perform similarly to VB at convergence, particularly if the hyper-parameters are learned for each algorithm (Asuncion *et al.* , 2009). Note that it was not possible to perform even a single iteration of batch VB on the full dataset in the allotted time of twelve hours. Following Hoffman et al., we show instead the performance of the algorithms on subsets of the data. This facilitates faster convergence, but reduces the quality of the final solution as the algorithms are consequently unable to exploit all of the data. In contrast, the stochastic algorithms are able to make use of large datasets while still converging quickly.

4.3.2 Small-Scale Experiments

Stochastic algorithms for LDA have previously only been used on large corpora, however they have the potential to be useful for finding topics very quickly on small corpora as well. The ability to learn interpretable topics in a matter of seconds is very beneficial for exploratory data analysis (EDA) applications, with a human in the loop. Near real-time topic modeling opens the way for the use of topic models in interactive software tools for document analysis.

We investigated the performance of the stochastic algorithms in this small-scale scenario using a corpus of 1740 scientific articles from years 1987 – 1999 of the machine learning conference NIPS. We ran the two stochastic inference algorithms for five seconds each, using the parameter settings from the previous experiments but with 20 topics. Each algorithm was run ten times. In the five seconds of training, SCVB0 was typically able to examine 3300 documents, while SVB was typically able to examine around 600 documents.

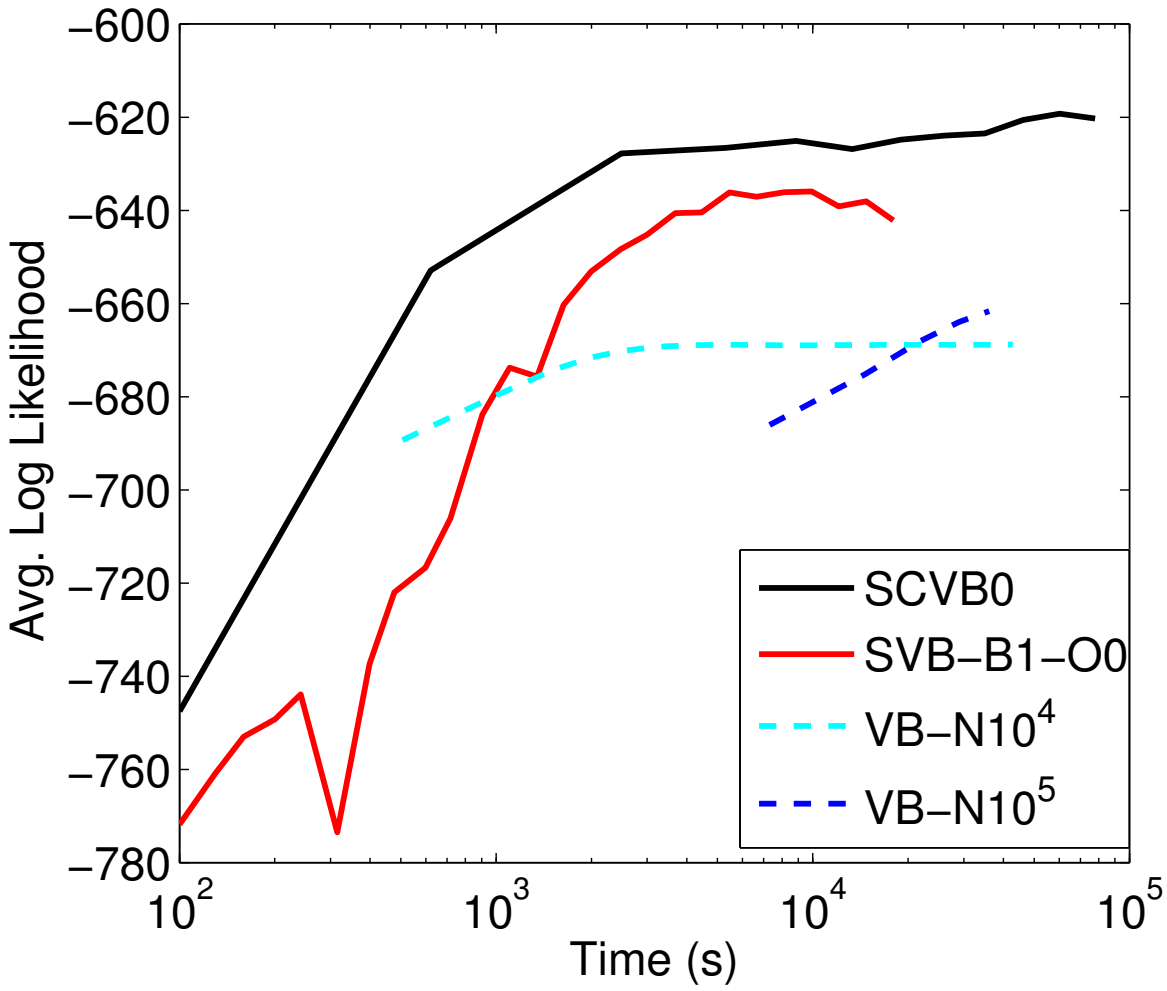


Figure 4.7: Log-likelihood vs time compared to batch VB for Wikipedia, where N is the number of documents used for training in batch VB.

With the EDA application in mind, we performed a human-subject experiment in the vein of the experiments proposed by Chang *et al.* (2009). The sets of topics returned by each run were randomly assigned across seven human subjects. The participants were all machine learning researchers with technical expertise in the subjects of interest to the NIPS community. The subjects did not know which algorithms generated which runs. The top ten words of the topics in each run were shown to the subjects, who were given the following instructions:

> Here are 20 collections of related words. Some words may not seem to "belong" with the other words. Count the total number of words in each collection that don't "belong."

The results provide an estimate of the number of "errors" that a topic model inference algorithm makes, relative to human judgement. It was found that the SCVB0 algorithm had 0.76 errors per topic on average, with a standard deviation of 1.1, while SVB had 1.6 errors per topic on average, with standard deviation 1.2. A one-sided two sample t-test rejected the hypothesis that the means of the errors per topic were equal, with significance level $\alpha = 0.05$. Randomly selected example topics are shown in Table 4.4. As can be seen from the table, both algorithms successfully learned coherent topics in this relatively short time frame.

We also performed a similar experiment on the Amazon Mechanical Turk crowd-sourcing service using the New York Times corpus. We ran the two stochastic inference algorithms for 60 seconds each using the same parameter settings as above but with 50 topics. Each user was presented with 20 random topics from each algorithm. Again, the subjects did not know which algorithms generated each set of topics. We included two easy questions with obvious answers and removed results from users who did not answer them correctly. This step eliminated 4 users, and the analysis was performed with the data from the remaining 52 participants.

SCVB0			SVB		
receptor	data	learning	model	results	visual
protein	classification	function	set	learning	data
secondary	vector	network	data	distribution	activity
proteins	class	neural	training	information	saliency
transducer	classifier	networks	learning	map	noise
binding	set	time	error	activity	similarity
concentration	algorithm	order	parameters	time	model
odor	feature	error	markov	figure	neural
morphology	space	dynamics	estimate	networks	representations
junction	vectors	point	speech	state	functions

Table 4.4: Randomly selected example topics after five seconds running time on the NIPS corpus.

Comparing the number of "errors" for SCVB0 to SVB for each user, we find that SCVB0 had 2.1 errors per topic on average, with standard deviation 1.0, and SVB had 4.4 errors on average with standard deviation 2.4. A paired t-test finds these differences significant for the sampled population at the $\alpha = .05$ level, with p-value $< .001$. Example topics selected uniformly at random from a randomly selected run of each algorithm are shown in Table 4.5, illustrating the relative difference in the coherence of the topics recovered by the two methods in this time period.

SCVB0			SVB		
county	station	league	president	year	mr
district	company	goals	midshipmen	cantatas	company
village	railway	years	open	edward	mep
north	business	club	forrester	computing	husbands
river	services	clubs	archives	main	net
area	market	season	iraq	years	state
east	line	played	left	area	builder
town	industry	cup	back	withdraw	offense
lake	stations	career	times	households	obscure
west	owned	team	saving	brain	advocacy

Table 4.5: Randomly selected example topics after sixty seconds running time on the NYT corpus.

4.4 How Good is the CVB0 Approximation?

Despite the approximations made, CVB0 has been shown to work well empirically (Asuncion *et al.* , 2009), and we have seen in Section 4.2.2 that its stochastic extension SCVB0 is also very effective in practice. The next three sections of this chapter aim to increase our understanding of why both CVB0 and SCVB0 perform well from a theoretical perspective. First, we consider the approximation to the mean field update made by CVB0, and give an argument which suggests that the approximation is reasonable in practice. In Section 4.5, we explore the connection between CVB0 and a MAP estimation algorithm, also due to Asuncion *et al.*, and provide an alternative derivation for SCVB0 as an online EM algorithm for MAP. We then use this alternative interpretation of the algorithm to prove convergence in Section 4.6.

4.4.1 An Explanation for the Success of CVB0

CVB0 makes several approximations to the terms in the collapsed variational Bayes update. Specifically, Teh et al. invoke the central limit theorem (CLT) to justify a Gaussian approximation, which is followed by a second-order Taylor series expansion approximation. Asuncion *et al.* further approximate the terms in the update by reducing the second order Taylor expansion to a first-order expansion. However, the first-order terms are zero, so this is equivalent to performing no expansion at all. Given that three successive approximations have been made, it is not immediately obvious how accurate the CVB0 update approximation should be.

Here, we give an argument suggesting that the CVB0 update is very accurate with sufficiently many words. Instead of using the CLT, a *law of large numbers*-style strategy is used. Roughly speaking, we note that the variance of the count terms involved in the update grows more

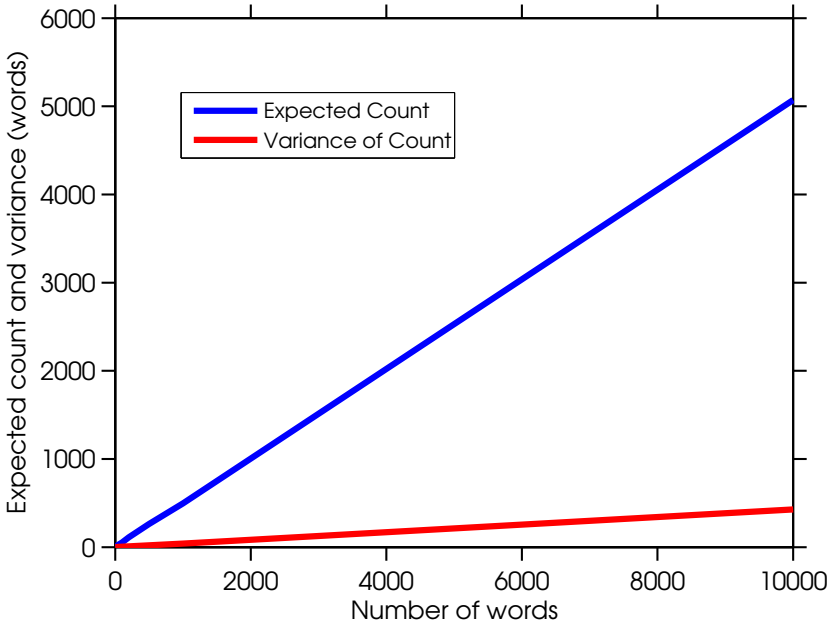


Figure 4.8: Mean and variance of CVB count variables, with respect to the mean field distribution, versus the amount of data. Here, for each word we randomly draw its probability of success $\gamma_i \sim \text{Beta}(0.1, 0.1)$ (the number of which is shown on the X-axis), and calculate the mean and variance of the number of successes (on the Y-axis) analytically. The variance will always grow more slowly than the mean.

slowly than the mean of the counts as we observe more data (see Figure 4.8), leading to a rapid decay of the probability that the count term is different from the mean by any significant fraction. Thus, with enough data, these count variables are well approximated by their mean, which is what is done by CVB0. More formally, we begin with a lemma, which is a slight variant on the weak law of large numbers, and has a similar proof.

Lemma 4.4.1. *Let $z_1, z_2, \ldots$ be an infinite sequence of independent Bernoulli random variables with $\gamma_1, \gamma_2, \ldots$ being their success probabilities. Let the random variable n_i be the number of successes in z up to z_i. Then for any $\epsilon > 0$,*

$$\lim_{i \to \infty} Pr(|n_i - E[n_i]| \geq \epsilon E[n_i]) = 0 \ . \tag{4.50}$$

Proof. First, by linearity of expectation, and by independence, the expected value and the variance of n_i are

$$E[n_i] = \sum_i \gamma_i$$
$$\text{Var}[n_i] = \sum_i \gamma_i(1-\gamma_i) \ .$$

Note that each term in the variance is smaller than the corresponding term in the expectation, so the variance grows more slowly in i than the expectation does. For $\epsilon > 0$, $\epsilon E[n_i]$ may potentially have the reverse behavior and grow more slowly than the variance, for small enough ϵ. Nevertheless, $\epsilon E[n_i]$ will increase with i at most linearly worse than $\text{Var}[n_i]$, and so $(\epsilon E[n_i])^2$ will increase more quickly than $\text{Var}[n_i]$, for sufficiently large n_i. Therefore,

$$\lim_{i\to\infty} \frac{\text{Var}[n_i]}{(\epsilon E[n_i])^2} = 0 \tag{4.51}$$

for any $\epsilon > 0$. By Chebyshev's inequality, we have

$$Pr(|n_i - E[n_i]| \geq \epsilon E[n_i]) \leq \frac{\text{Var}[n_i]}{(\epsilon E[n_i])^2} \ . \tag{4.52}$$

Finally, taking limits on both sides, we have

$$\lim_{i\to\infty} Pr(|n_i - E[n_i]| \geq \epsilon E[n_i]) \leq \lim_{i\to\infty} \frac{\text{Var}[n_i]}{(\epsilon E[n_i])^2} = 0 \ . \tag{4.53}$$

□

In Figure 4.9, we illustrate the Chebyshev bound in Equation 4.52, which is the key to the lemma. The Y-axis bounds the error introduced by assuming that the count variable is equal to its mean, relative to the size of the mean. The lemma states that the error will become

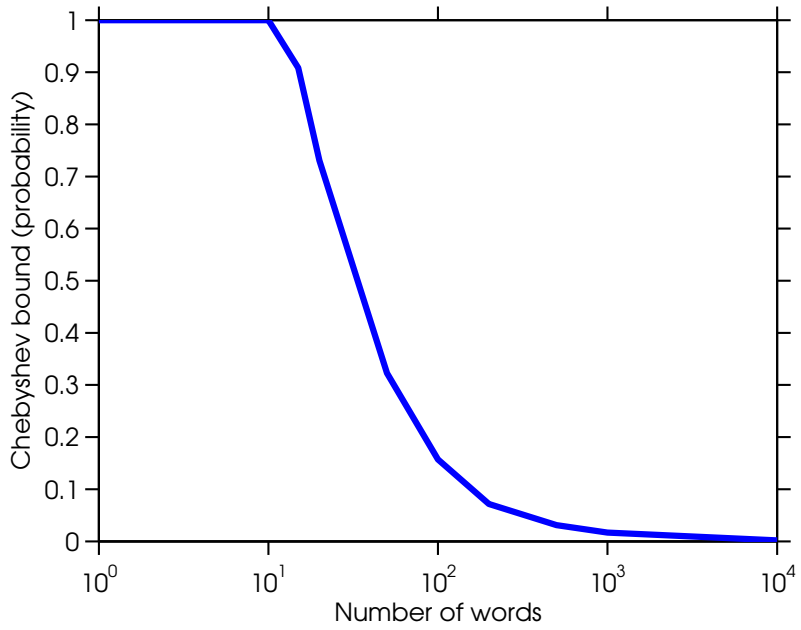


Figure 4.9: A simulation-based exploration of the Chebyshev bound in Equation 4.52. For each word, we draw its probability of being included in the count $\gamma_i \sim \text{Beta}(0.1, 0.1)$. On the Y-axis, we report the bound on the probability of the count being more than 10% different from the expected count, computed analytically.

arbitrarily small (relative to the mean) as the X-axis approaches infinity, which we can see in the figure as the curve approaches the asymptote at zero.

This lemma applies to the count terms in the exact CVB update, $n_k^{(d)-(d,i)}$, $n_k^{(w_i^{(d)})-(d,i)}$ and $n_k^{-(d,i)}$. These count variables are each a sum of independent Bernoulli's, due to the mean field assumption. In the context of CVB0, this lemma says that for sufficiently many words, the variance under the mean field distribution of each of the count variables becomes arbitrarily small, *relative to the magnitude of the count*. E.g., $n_k^{(d)-(d,i)}$ will be within ϵ percent of the CVB0 statistic $N_{dk}^{\Theta\neg id}$, for any $\epsilon > 0$.

Thus, with sufficiently many words, most of the probability mass for the count variables will be very close to the mean. The expectations needed for the update, e.g. $E_{q_{\neg id}}[\log(n_k^{(d)-(d,i)} + \alpha_k)]$, can then be well approximated by taking the expectation with respect to a delta distribution δ_μ at the mean μ, which is equal to the corresponding CVB0 statistic.

For example, in the case of document-level parameters,

$$E_{q_{\neg id}}[\log(n_k^{(d)-(d,i)} + \alpha_k)] \approx E_{\delta_{N_{dk}^{\Theta\neg id}}}[\log(n_k^{(d)-(d,i)} + \alpha_k)] \ . \tag{4.54}$$

Under the delta distribution $\delta_{N_{dk}^{\Theta\neg id}}$, with probability one, $n_k^{(d)-(d,i)} = N_{dk}^{\Theta\neg id}$, and so the CVB0 approximation is exact:

$$E_{\delta_{N_{dk}^{\Theta\neg id}}}[\log(n_k^{(d)-(d,i)} + \alpha_k)] = \log(N_{dk}^{\Theta\neg id} + \alpha_k)] = \log(E_{\delta_{N_{dk}^{\Theta\neg id}}}[n_k^{(d)-(d,i)}] + \alpha_k) \ . \tag{4.55}$$

So for sufficiently large values of the count terms,

$$E_{q_{\neg id}}[\log(n_k^{(d)-(d,i)} + \alpha_k)] \approx \log(N_{dk}^{\Theta\neg id} + \alpha_k)] = \log(E_{\delta_{N_{dk}^{\Theta\neg id}}}[n_k^{(d)-(d,i)}] + \alpha_k) \ , \tag{4.56}$$

which is the approximation made by CVB0. We can therefore expect the CVB0 approximation of the mean field update to be very accurate when there are sufficiently many words involved in the computation of each term, i.e. when we have many documents and the documents contain many words. It should also be noted that the argument also applies for Teh et al.'s second order approximation, because the second order term in Equation 4.25 will drop out in the limit by Equation 4.51.

To corroborate these arguments, we also explored the accuracy of the approximation empirically. We generated a synthetic variational distribution q by drawing a sequence of γ variables Beta$(0.1, 0.1)$. Varying the number of words i, we computed CVB0 and CVB approximations to $E_q[\log(n_i + \alpha)]$, where we set $\alpha = 0.01$ (a reasonable value for LDA). Finally, we computed Monte Carlo estimates of $E_q[\log(n_i + \alpha)]$ by simulating 10,000 draws from q, and recorded the L1 error of the CVB and CVB0 from the Monte Carlo estimates of the true value. The experiment was repeated 1,000 times, and we averaged over the repeats. The results of the experiment are shown in Figure 4.10.

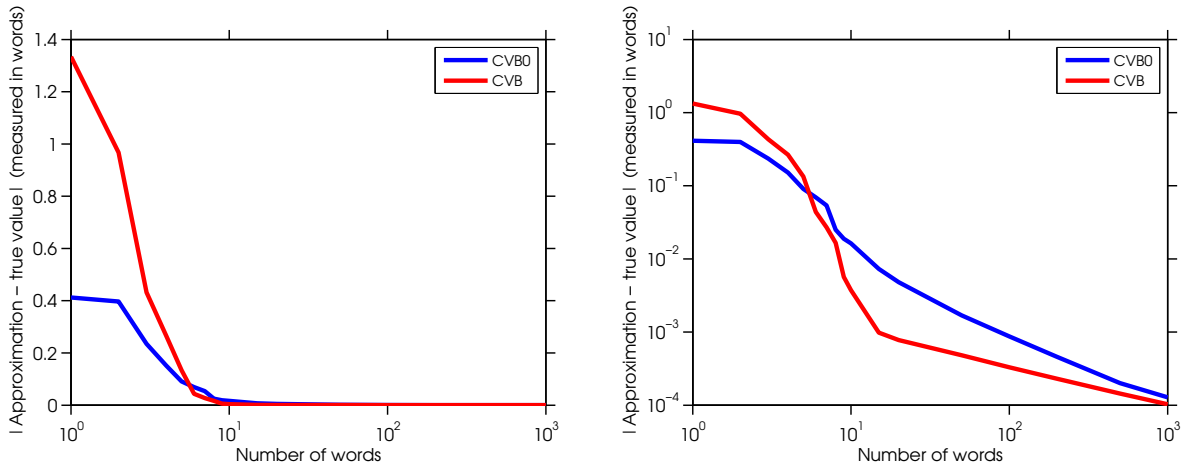


Figure 4.10: Measuring the error of CVB0. **Left**: A semi-log plot, with a logarithmic scale on the X-axis. **Right**: The same plot, on a log-log scale.

We found that both CVB0 and CVB were very accurate when given more than around ten words. When few words were available, CVB0 was surprisingly *more* accurate than CVB. We hypothesize that this is because the central limit theorem does not yet apply, so approximating the distribution by a delta function (CVB0) is more accurate than approximating the distribution by a Gaussian (CVB). When the number of words increases the CLT becomes more applicable, in which case CVB does better than CVB0. Nevertheless, by this time both methods are already very accurate, and we need to use a log-log plot to observe this difference clearly.

4.5 An Alternative Perspective: MAP Estimation

In the SCVB0 algorithm, because the γ's are not maintained we must approximate Equation 4.28 with Equation 4.43, neglecting the subtraction of the previous value of γ_{id} from the CVB0 statistics when updating γ_{id}. This approximation results in an algorithm which is equivalent to an EM algorithm for MAP estimation which operates on an unnormalized parameterization of LDA (Asuncion *et al.* , 2009). Therefore, the approximate collapsed variational updates of SCVB0 can also be understood as MAP estimation updates. Us-

ing this interpretation, we will give an alternative derivation of SCVB0 as a version of Cappé & Moulines (2009)'s online EM algorithm as applied to MAP estimation for LDA, thus providing an alternative perspective on the algorithm.

4.5.1 CVB0 and MAP Estimation

It can be shown that iterating the following batch algorithm update optimizes an EM lower bound on the posterior probability of the parameters (Asuncion *et al.* , 2009):

$$\bar{\gamma}_{idk} :\propto \frac{\bar{N}^{\Phi}_{w_i^{(d)}k} + \beta - 1}{\bar{N}^{Z}_{k} + W(\beta - 1)}(\bar{N}^{\Theta}_{dk} + \alpha - 1) \ , \tag{4.57}$$

where $\bar{\gamma}_{idk} \triangleq Pr(z_i^{(d)} = k | \bar{N}^{\Phi}, \bar{N}^{Z}, \bar{N}^{\Theta}, w_i^{(d)})$ are EM "responsibilities," and the other variables, which we will refer to as *EM statistics*, are aggregate statistics computed from sums of these responsibilities,

$$\bar{N}^{Z}_{k} \triangleq \sum_{id} \bar{\gamma}_{idk} \qquad \bar{N}^{\Theta}_{dk} \triangleq \sum_{i} \bar{\gamma}_{idk} \qquad \bar{N}^{\Phi}_{wk} \triangleq \sum_{id:w_i^{(d)}=w} \bar{\gamma}_{idk} \ . \tag{4.58}$$

Upon completion of the algorithm, MAP estimates of the parameters can be recovered by

$$\hat{\Phi}^{(k)}_{w} = \frac{\bar{N}^{\Phi}_{wk} + \beta - 1}{\bar{N}^{Z}_{k} + W(\beta - 1)} \qquad \hat{\theta}^{(d)}_{k} = \frac{\bar{N}^{\Theta}_{dk} + \alpha - 1}{C_d + K\alpha - K} \ , \tag{4.59}$$

where C_d is the length of document d. We can interpret $\bar{N}^{\Phi}$ and $\bar{N}^{\Theta}$ as unnormalized representations of $\mathbf{\Phi}$ and $\mathbf{\Theta}$, so we will refer to this algorithm as *unnormalized MAP LDA* (MAP_LDA_U). A derivation for this algorithm is given in Appendix B. Note that if we identify the EM statistics and responsibilities with CVB0 statistics and variational distributions, Equation 4.57 is identical to the SCVB0 update in Equation 4.43, but with the hyperparameters adjusted by one. We can think of this as the batch version of SCVB0. After

adjusting the hyper-parameters, the only difference from the batch CVB0 update (Equation 4.28) is that the previous value of the current parameter is not subtracted from the counts.

4.5.2 Online EM for MAP Estimation

We now use a slight generalization of the online EM algorithm (Cappé & Moulines, 2009) which we outlined in Section 4.1.4 to derive a stochastic version of this unnormalized MAP LDA algorithm. Online EM performs maximum likelihood estimation by alternating between updating an online estimate of the expected sufficient statistics of the complete-data log-likelihood, and optimizing parameter estimates via the usual EM M-step. We consider this algorithm as applied to the unnormalized parameterization of LDA above, where the parameters of interest are estimates $\hat{\tilde{\mathbf{N}}}^{\mathbf{\Phi}}$, $\hat{\tilde{\mathbf{N}}}^{\mathbf{\Theta}}$, $\hat{\tilde{\mathbf{N}}}^{\mathbf{Z}}$ of the EM statistics, which are related to $\mathbf{\Theta}$ and $\mathbf{\Phi}$ via Equation 4.59. We also adapt the online EM algorithm to perform MAP estimation instead of finding the MLE, and to operate with stochasticity at the word-level as well as at the document-level. The resulting algorithm is procedurally identical to SCVB0, which gives us an alternative perspective on our algorithm.

Recall that online EM assumes that the complete data likelihood is in the exponential family, and that the stochastic E-step operates on estimates of its expected sufficient statistics. The first step of deriving the online EM algorithm is to write the complete data likelihood of MAP_LDA_U in exponential family form. Making use of the derivations in Appendix B, we find that the complete data likelihood for a word $w_i^{(d)}$ and its topic assignment $z_i^{(d)}$ is

$$\begin{aligned}
&\exp\Big(\sum_{wk}[w_i^{(d)}=w][z_i^{(d)}=k]\log\big(\frac{\hat{\tilde{N}}^{\Phi}_{wk}+\beta-1}{\hat{\tilde{N}}^{Z}_{k}+W(\beta-1)}\big)+\sum_{k}[z_i^{(d)}=k]\log\big(\frac{\hat{\tilde{N}}^{\Theta}_{dk}+\alpha-1}{C_d+K(\alpha-1)}\big)\Big)\\
\propto\exp\Big(&\sum_{wk}[w_i^{(d)}=w][z_i^{(d)}=k]\log(\hat{\tilde{N}}^{\Phi}_{wk}+\beta-1)+\sum_{k}[z_i^{(d)}=k]\log(\hat{\tilde{N}}^{\Theta}_{dk}+\alpha-1)\\
&-\sum_{k}[z_i^{(d)}=k]\log(\hat{\tilde{N}}^{Z}_{k}+W(\beta-1))\Big)\,,
\end{aligned}\tag{4.60}$$

where $[a = b]$ is a Kronecker delta function, equal to one if $a = b$ and zero otherwise, C_d is the length of document d and $\hat{\hat{\mathbf{N}}}$ variables denote current estimates, not necessarily synchronized with $\bar{\gamma}$. There is a direct mapping between the $\hat{\hat{\mathbf{N}}}$'s and the parameters $\mathbf{\Phi}$, $\mathbf{\Theta}$, so we should interpret them here as parameters rather than as the EM statistics themselves – although we will see below that they will soon be assigned to be equal to the EM statistics, which justifies this notation. Now that we have written the equation in exponential family form, we can see that the exponential family sufficient statistics are the delta functions (and products of delta functions),

$$S^{(w)}(w_i^{(d)}, z_i^{(d)}) = ([w_i^{(d)} = 1][z_i^{(d)} = 1], \ldots, [w_i^{(d)} = W][z_i^{(d)} = K], \\ [z_i^{(d)} = 1], \ldots, [z_i^{(d)} = K], [z_i^{(d)} = 1], \ldots, [z_i^{(d)} = K])^\intercal \ , \tag{4.61}$$

and the expected sufficient statistics, given current parameter estimates, are appropriate entries of the EM responsibilities vector $\bar{\gamma}$,

$$\bar{s}^{(w)}(w_i^{(d)}, z_i^{(d)}) = ([w_i^{(d)} = 1]\bar{\gamma}_{id1}, \ldots, [w_i^{(d)} = W]\bar{\gamma}_{idK}, \bar{\gamma}_{id1}, \ldots, \bar{\gamma}_{idK}, \bar{\gamma}_{id1}, \ldots, \bar{\gamma}_{idK})^\intercal \ . \tag{4.62}$$

Cappe and Moulines normalize the likelihood, and the sufficient statistics, by the number of data points n, so that n need not be specified in advance. However, since we are performing MAP estimation, unlike the MLE algorithm described by Cappe and Moulines, we need to modify the algorithm to estimate the unnormalized expected sufficient statistics for the entire corpus in order to maintain the correct scale relative to the prior. This can be achieved by scaling the per-word expected sufficient statistics by appropriate constants to match the size of the corpus (or document, for per-document statistics)

$$\bar{s}'^{(w)}(w_i^{(d)}, z_i^{(d)}) = (C[w_i^{(d)} = 1]\bar{\gamma}_{id1}, \ldots, C[w_i^{(d)} = W]\bar{\gamma}_{idK}, \\ C_d\bar{\gamma}_{id1}, \ldots, C_d\bar{\gamma}_{idK}, C\bar{\gamma}_{id1}, \ldots, C\bar{\gamma}_{idK})^\intercal \ . \tag{4.63}$$

The average of these corpus-wide expected sufficient statistics, computed across all tokens in the corpus, is equal to the EM statistics. Collecting them into appropriate matrices, we can write the the expected sufficient statistics as

$$\bar{\mathbf{N}} = (\bar{\mathbf{N}}^{\mathbf{\Theta}}, \bar{\mathbf{N}}^{\mathbf{\Phi}}, \bar{\mathbf{N}}^{\mathbf{Z}}) \ , \tag{4.64}$$

which are precisely the EM statistics of Equation 4.58. In fact, optimizing the EM objective function with respect to the parameters, we find that the M-step assigns the estimated EM statistics $\hat{\bar{\mathbf{N}}}$ to be consistent with the EM statistics $\bar{\mathbf{N}}$ computed in the E-step (cf. Appendix B),

$$\hat{\bar{\mathbf{N}}}^{\mathbf{\Theta}} := \bar{\mathbf{N}}^{\mathbf{\Theta}} \qquad \hat{\bar{\mathbf{N}}}^{\mathbf{\Phi}} := \bar{\mathbf{N}}^{\mathbf{\Phi}} \qquad \hat{\bar{\mathbf{N}}}^{\mathbf{Z}} := \bar{\mathbf{N}}^{\mathbf{Z}} \ . \tag{4.65}$$

We therefore do not need to store parameter estimates $\hat{\bar{\mathbf{N}}}$ separately from expected sufficient statistics $\bar{\mathbf{N}}$, as M-step updated parameter estimates are always equal to the expected sufficient statistics from the E-step. Inserting Equation 4.63 into the online EM update (Equation 4.39) and using separate step size schedules for document statistics and topic statistics, the online E-step after processing token $w_i^{(d)}$ is given by

$$\bar{\mathbf{N}}^{Z} := (1 - \rho_t^{\Phi})\bar{\mathbf{N}}^{Z} + \rho_t^{\Phi} C \bar{\gamma}_{id} \tag{4.66}$$

$$\bar{\mathbf{N}}_w^{\Phi} := (1 - \rho_t^{\Phi})\bar{\mathbf{N}}^{\Phi} + \rho_t^{\Phi} C \bar{\gamma}_{id} [w_i^{(d)} = w] \ , \forall w \tag{4.67}$$

$$\bar{\mathbf{N}}_d^{\Theta} := (1 - \rho_t^{\Theta})\bar{\mathbf{N}}_d^{\Theta} + \rho_t^{\Theta} C_d \bar{\gamma}_{id} \ , \tag{4.68}$$

with $\bar{\gamma}_{id}$ computed via Equation 4.57. The online EM algorithm we have just derived is procedurally identical to SCVB0 with minibatches of size one, identifying EM responsibilities and statistics with SCVB0 responsibilities and statistics, and with the hyper-parameters adjusted by one. Under this interpretation, an alternative name for SCVB0 might be *stochastic unnormalized MAP LDA* (S_MAP_LDA_U).

4.5.3 Discussion Regarding the MAP and VB Interpretations of SCVB0

We have shown that SCVB0 can be interpreted both as performing collapsed variational Bayes, and as performing MAP estimation. The MAP interpretation of the algorithm implicitly uses adjusted values of the hyperparameters, so this does not contradict the original CVB interpretation, but suggests that there is a close relationship between the optimal solutions of the CVB and MAP estimation problems. Furthermore, the variational Bayes interpretation is particularly useful when performing inference on the $\mathbf{z}$'s of unseen held-out documents, as it allows us to reason over their full posterior.[12]

Interpreting SCVB0 as a MAP estimation algorithm may also help to explain the improvement in predictive performance relative to SVB. The MAP estimate approximates the posterior distribution by a delta function at its mode, while mean field variational Bayes approximates the posterior by a factorized distribution. As the amount of training data increases, the posterior distribution should become more peaked around the mode, i.e. more similar to the delta function at the MAP. The factorized distribution of mean field, on the other hand, may not be able to accurately represent the posterior distribution in the large data regime. We conjecture that in many cases, given enough data it may be preferable to perform MAP estimation instead of variational inference. This observation seems particularly relevant in the case where stochastic algorithms are necessary due to the large amount of data available.

[12]This was pointed out by Dave Blei (personal communication). It is the same insight that motivates the original formulation of LDA with the Dirichlet prior over document-level parameters $\boldsymbol{\Theta}$, and the original variational EM approach, in Blei *et al.* (2003), which performs VB inference over $\boldsymbol{\Theta}$ and maximum likelihood estimation over $\boldsymbol{\Phi}$.

4.6 Convergence Analysis

The MAP estimation interpretation of SCVB0 is the interpretation that is most amenable to convergence analysis, since MAP_LDA_U exactly optimizes a well-defined objective function, while CVB0 has approximate updates. Under the MAP estimation interpretation of SCVB0, it can be shown that the algorithm converges to a stationary point of the MAP objective function, computed as if the prior were modified by increasing the hyper-parameters by one.

The proof strategy broadly follows that of Cappe and Moulines. First, the algorithm is written as a Robbins and Monro stochastic approximation (SA) algorithm. Then, it is shown that there exists a Lyapunov function satisfying the conditions of Andrieu *et al.* (2005), which are sufficient to establish convergence for an SA algorithm. In the context of an SA algorithm, a Lyapunov function can be understood as an "objective function" which, in the absence of stochastic noise, the SA would improve monotonically if small enough steps were taken in the direction of the updates.

We now state the theorem and its proof more formally. In this section, the notation will follow the MAP interpretation of the algorithm (summarized in Table 4.6, along with new notation introduced for the convergence analysis). We have the following theorem:

Theorem 4.6.1. *For an appropriate sequence of step sizes satisfying*

- $0 < \rho_t^{\Phi} \leq 1 \ \forall t$, $0 < \rho_t^{\Theta} \leq 1 \ \forall t$,
- $\sum_{t=1}^{\infty} \rho_t^{\Phi} = \infty$, $\lim_{t\to\infty} \rho_t^{\Phi} = 0$,
- $\sum_{t=1}^{\infty} \rho_t^{\Theta} = \infty$, $\lim_{t\to\infty} \rho_t^{\Theta} = 0$,

in the limit as the number of iterations t approaches infinity SCVB0 converges to a stationary point of the MAP objective function.

$\bar{\gamma}_{id}$	EM responsibility vector for word (i, d), $1 \times K$
$\bar{\mathbf{N}}^{\mathbf{\Theta}}$	EM statistic: responsibility counts per (document, topic) pair, $D \times K$
$\bar{\mathbf{N}}^{\mathbf{\Phi}}$	EM statistic: responsibility counts per (word, topic) pair, $W \times K$
$\bar{\mathbf{N}}^{\mathbf{Z}}$	EM statistic: responsibility counts per topic, $1 \times K$
$\hat{\bar{\mathbf{N}}}^{\mathbf{\Theta}}$	Current parameter estimate of $\bar{\mathbf{N}}^{\mathbf{\Theta}}$, $D \times K$
$\hat{\bar{\mathbf{N}}}^{\mathbf{\Phi}}$	Current parameter estimate of $\bar{\mathbf{N}}^{\mathbf{\Phi}}$, $W \times K$
$\hat{\bar{\mathbf{N}}}^{\mathbf{Z}}$	Current parameter estimate of $\bar{\mathbf{N}}^{\mathbf{Z}}$, $1 \times K$
$\hat{\bar{\mathbf{N}}}$	A parameter estimate of all EM statistics, tuple valued
$\hat{\bar{\mathbf{N}}}^{(t)}$	Current parameter estimate of all EM statistics at step t, tuple valued
$\hat{\bar{\mathbf{N}}}_c$	A parameter estimate of EM statistic c (e.g. $\hat{\bar{\mathbf{N}}}_d^{\mathbf{\Theta}}$), matrix valued
$f_{c,w}(\hat{\bar{\mathbf{N}}})$	Updated value of $\hat{\bar{\mathbf{N}}}_c$ after MAP_LDA_U E and M steps at $\hat{\bar{\mathbf{N}}}$ with data w
$w^{(t+1)}$	Dictionary index for word examined by SCVB0 at step $t+1$
$\bar{\gamma}^{(t+1)}$	EM responsibility vector computed for word $w^{(t+1)}$, $1 \times K$
$\bar{\mathbf{Y}}^{(t+1)}$	Stochastic estimate of $\bar{\mathbf{N}}^{\Phi}$ based on word $w^{(t+1)}$, $W \times K$
$\bar{s}_c(w^{(t+1)}, \hat{\bar{\mathbf{N}}})$	Stochastic estimate of $f_{c,w}(\hat{\bar{\mathbf{N}}})$ based on $w^{(t+1)}$, matrix valued
$\xi^{(t+1)}$	Stochastic error made by $\bar{s}_c(w^{(t+1)}, \hat{\bar{\mathbf{N}}})$, matrix valued

Table 4.6: Summary of notation for the convergence analysis.

Proof. The first stage of the proof is to represent SCVB0 as a Robbins-Monro SA algorithm which aims to find a stationary point of the MAP objective. We derive a deterministic algorithm for MAP estimation which operates entirely on the EM statistics (which are equivalent to the CVB0 statistics). We then show that SCVB0 is a stochastic approximation algorithm for finding the roots of its update moves.

A deterministic algorithm for MAP estimation

Returning to the batch setting for the moment, let us consider the MAP_LDA_U algorithm of Section 4.5.1. As usual for an EM algorithm (Neal & Hinton, 1998), this is a coordinate ascent algorithm on the EM lower bound (see Appendix B). The coordinate ascent update moves available to us are the CVB0-like update of Equation 4.57 to update the $\bar{\gamma}$'s (the E-step), and the synchronization of each entry of the $\hat{\bar{\mathbf{N}}}$ estimated EM statistics matrices with the $\bar{\gamma}$'s (the M-step, Equation 4.65).

As a coordinate ascent method, we are free to select any ordering of the updates. We previously considered an ordering of the updates where the $\hat{\bar{\mathbf{N}}}$'s were synchronized with the $\bar{\gamma}$'s after every word, leading to an algorithm similar to CVB0. Instead, let us choose a more typical EM update schedule which alternates between a full E-step, i.e. updating every $\bar{\gamma}_{id}$ without updating the estimated EM statistics $\hat{\bar{\mathbf{N}}}$, and a full M-step, i.e synchronizing $\hat{\bar{\mathbf{N}}}$ with the $\bar{\gamma}$'s. We do not need to maintain the $\bar{\gamma}$'s between iterations of this procedure because they do not depend on each other given the estimated EM statistics. We can view this version of MAP_LDA_U as operating on just the estimated EM statistics.

The updated value of the EM statistics, then, is a function of the previous value of these statistics. We will now write the algorithm in terms of this function. For each EM statistic $c \in \{\hat{\bar{\mathbf{N}}}^{\boldsymbol{\Theta}}_1, \ldots, \hat{\bar{\mathbf{N}}}^{\boldsymbol{\Theta}}_D, \hat{\bar{\mathbf{N}}}^{\boldsymbol{\Phi}}, \hat{\bar{\mathbf{N}}}^{\mathbf{Z}}\}$, let $f_{c,w}(\hat{\bar{\mathbf{N}}}) : S \rightarrow S_c$ be a mapping from a current value of the EM statistics $\hat{\bar{\mathbf{N}}} = (\hat{\bar{\mathbf{N}}}^{\boldsymbol{\Theta}}_1, \ldots, \hat{\bar{\mathbf{N}}}^{\boldsymbol{\Theta}}_D, \hat{\bar{\mathbf{N}}}^{\boldsymbol{\Phi}}, \hat{\bar{\mathbf{N}}}^{\mathbf{Z}})$ to the updated value of statistic $\hat{\bar{\mathbf{N}}}_c$ after performing an E-step to estimate the $\bar{\gamma}$'s, and then performing an M-step. Here, w is the full corpus, S is the space of possible assignments for the EM statistics, and S_c is the space of possible assignments for EM statistic c. By performing the mapping $f_{c,w}(\hat{\bar{\mathbf{N}}})$ on all of the EM statistics together, we take a complete step in the EM algorithm we have just described.

Casting SCVB0 as a Robbins-Monro SA Algorithm

We now switch to the stochastic case. Here, we will describe SCVB0 in terms of the deterministic algorithm derived above, using the MAP interpretation of the algorithm. Let $\hat{\bar{\mathbf{N}}}^{(t)}$ be the current EM statistics at word iteration t of the algorithm. Furthermore, at iteration $t+1$, let $\bar{\gamma}^{(t+1)}$ be the output of the MAP_LDA_U update (which is equivalent to the SCVB0 update) based on the latest randomly selected word $w^{(t+1)}$ and the current state $\hat{\bar{\mathbf{N}}}^{(t)}$. We define $\bar{s}_c(w^{(t+1)}, \hat{\bar{\mathbf{N}}})$ to be SCVB0's stochastic estimate of the updated statistic c based on $w^{(t+1)}$, as per Equations 4.66 – 4.68, i.e.

$$\bar{s}_{\hat{\tilde{\mathbf{N}}}^{\mathbf{Z}}}(w^{(t+1)}, \hat{\tilde{\mathbf{N}}}) = C\bar{\gamma}^{(t+1)} \tag{4.69}$$

$$\bar{s}_{\hat{\tilde{\mathbf{N}}}^{\mathbf{\Phi}}}(w^{(t+1)}, \hat{\tilde{\mathbf{N}}}) = C\bar{\mathbf{Y}}^{(t+1)} \tag{4.70}$$

$$\bar{s}_{\hat{\tilde{\mathbf{N}}}_d^{\mathbf{\Theta}}}(w^{(t+1)}, \hat{\tilde{\mathbf{N}}}) = C_d\bar{\gamma}^{(t+1)} \ . \tag{4.71}$$

The deterministic algorithm above computes all $\bar{\gamma}$'s and then updates $\hat{\tilde{\mathbf{N}}}^{(t)}$, while SCVB0 computes a single random $\bar{\gamma}^{(t+1)}$, and uses this to noisily estimate the same update to $\hat{\tilde{\mathbf{N}}}^{(t)}$. By Equations 4.40 – 4.42, the estimate is unbiased, with $E[\bar{s}_c(w^{(t+1)}, \hat{\tilde{\mathbf{N}}})] = f_{c,w}(\hat{\tilde{\mathbf{N}}})$, where the expectation is with respect to the sampling of $w^{(t+1)}$. Finally, let

$$\xi^{(t+1)} = \bar{s}_c(w^{(t+1)}, \hat{\tilde{\mathbf{N}}}^{(t)}) - f_{c,w}(\hat{\tilde{\mathbf{N}}}^{(t)}) \tag{4.72}$$

be the stochastic error made at step $t+1$, and observe that $E[\xi^{(t+1)}] = 0$. We can rewrite the SCVB0 updates for each EM statistic c as

$$\begin{aligned}
\hat{\tilde{\mathbf{N}}}_c^{(t+1)} &= (1-\rho_{t+1}^c)\hat{\tilde{\mathbf{N}}}_c^{(t)} + \rho_{t+1}^c \bar{s}_c(w^{(t+1)}, \hat{\tilde{\mathbf{N}}}) \\
&= \hat{\tilde{\mathbf{N}}}_c^{(t)} + \rho_{t+1}^c(-\hat{\tilde{\mathbf{N}}}_c^{(t)} + \bar{s}_c(w^{(t+1)}, \hat{\tilde{\mathbf{N}}})) \\
&= \hat{\tilde{\mathbf{N}}}_c^{(t)} + \rho_{t+1}^c(f_{c,w}(\hat{\tilde{\mathbf{N}}}^{(t)}) - \hat{\tilde{\mathbf{N}}}_c^{(t)} + \bar{s}_c(w^{(t+1)}, \hat{\tilde{\mathbf{N}}}) - f_{c,w}(\hat{\tilde{\mathbf{N}}}^{(t)})) \\
&= \hat{\tilde{\mathbf{N}}}_c^{(t)} + \rho_{t+1}^c(f_{c,w}(\hat{\tilde{\mathbf{N}}}^{(t)}) - \hat{\tilde{\mathbf{N}}}_c^{(t)} + \xi^{(t+1)}) \ .
\end{aligned} \tag{4.73}$$

We have now written the update equations as the Robbins-Monro SA updates of Equation 4.33 (Robbins & Monro, 1951). In this form, we can see that iterating each of the SCVB0 updates corresponds to a stochastic approximation algorithm for finding the zeros of $f_{c,w}(\hat{\tilde{\mathbf{N}}}^{(t)}) - \hat{\tilde{\mathbf{N}}}_c^{(t)}$, which are the steps that MAP_LDA_U takes. The full SCVB0 algorithm, which performs an update on all of the statistics together, is therefore an SA algorithm for finding the locations where the steps that MAP_LDA_U takes are $\mathbf{0}$, i.e. the fixed points of MAP_LDA_U. Since MAP_LDA_U is an EM algorithm, its fixed points are the stationary

points of the posterior probability of the parameters, with the parameters being associated with the EM statistics by Equation 4.59.

A Lyapunov Function to Show that the SA Algorithm Converges

Having shown that SCVB0 is an SA algorithm for finding the stationary points of the MAP objective function, we can now make use of known results for stochastic approximation algorithms to prove convergence. Theorem 2.3 of Andrieu *et al.* (2005) states that under mild conditions, the existence of a Lyapunov function, along with a boundedness condition, implies that such a Robbins-Monro algorithm will converge to a root of the target function with step size schedules such as those in the conditions of the theorem. In the context of an SA algorithm, a Lyapunov function can be understood as an "objective function" which, in the absence of stochastic noise, the SA would improve monotonically if small enough steps were taken in the direction of the updates. In Appendix C, we show that the negative of the Lagrangian of the EM lower bound is a Lyapunov function of the overall SCVB0 algorithm and the set of fixed points of the EM algorithm, and that the required conditions on the function hold. The boundedness condition, namely that the state variables stay within a compact subset of the state space, follows by observing that $0 < \|\bar{s}_c(w^{(t+1)}, \hat{\tilde{\mathbf{N}}})_{(:)}\|_\infty \leq C$ for every EM statistic c, where $\mathbf{A}_{(:)}$ concatentates each entry of $\mathbf{A}$ into a vector. If the initial state also satisfies this, by the convexity of the update, the $\hat{\tilde{\mathbf{N}}}$'s will always have their L_∞ norms similarly bounded. Having demonstrated that the assumptions required by Theorem 2.3 of Andreiu et al. hold, the convergence result follows. □

4.7 Discussion / Related Work

Connections can be drawn between SCVB0 and other methods in the literature. The SCVB0 scheme is reminiscent of the online EM algorithm of Cappé & Moulines (2009), which also alternates between per data-point parameter updates and online estimates of the expected

values of sufficient statistics. Online EM optimizes the EM lower bound on the log-likelihood in the M-step and computes online averages of exponential family sufficient statistics, while SCVB0 (approximately) updates the mean-field evidence lower bound in the M-step and computes online averages of sufficient statistics required for a CVB0 update in the E-step. As discussed in Section 4.5.2, when viewed as a MAP estimation algorithm SCVB0 can also be derived as an extension of online EM, applied to LDA.

The SCVB0 algorithm also has a very similar structure to SVB, alternating between passes through a document (the optional "burn-in" passes) to learn document parameters, and updating variables associated with topics. However, SCVB0 is stochastic at the word-level while SVB is stochastic at the document level. This allows SCVB0 to take the stochastic approach further than SVB, by making stochastic estimates of the document parameters as well as of the topics.

In more detail, the general framework of Hoffman et al. performs inference on "local" parameters specific to a data point, which are used to perform a stochastic update on the "global" parameters. For SVB, variational parameters for $\theta^{(d)}$ are local parameters for document d, and variational parameters for topics are the global parameters. For SCVB0, the γ_{id}'s are local parameters for a word, and both document parameters $\mathbf{N}^{\Theta}$ and topic parameters $\mathbf{N}^{\Phi}$ are global parameters. This means that updates to the parameters can be made *before processing all of the words in the document*, while SVB must wait to complete the processing of a document before performing an update.

The incremental algorithm of Banerjee & Basu (2007), for MAP inference in LDA, is also closely related to the proposed algorithm. They estimate topic probabilities for each word sequentially, and update MAP estimates of $\mathbf{\Phi}$ and $\mathbf{\Theta}$ incrementally, using the expected assignments of words to topics in the current document. SCVB0 can be understood as the collapsed, stochastic variational version of Banerjee and Basu's incremental uncollapsed MAP estimation algorithm. Interpreting SCVB0 as a MAP estimation algorithm, SCVB0 is

the online EM algorithm for MAP estimation operating on the unnormalized representation of LDA, while Banerjee and Basu's algorithm is the incremental EM algorithm operating on the usual normalized representation of LDA. A related algorithm is the sequential Monte Carlo (SMC) approach used by Ahmed *et al.* (2011), which sequentially Gibbs samples the topic assignments of each document for each of F importance-weighted particles. This method updates count statistics for each particle incrementally via sampling, while SCVB0 updates count statistics with online-averaged updates via optimization.

Another stochastic algorithm for LDA, due to Mimno *et al.* (2012), operates in a partially collapsed space, placing it in-between SVB and SCVB0 in terms of representation. Their algorithm collapses out $\boldsymbol{\Theta}$ but does not collapse out $\boldsymbol{\Phi}$. Estimates of online natural gradient update directions are computed by performing Gibbs sampling on the topic assignments of the words in each document, and averaging over the samples. The gradient estimate is non-zero only for word-topic pairs which occurred in the samples. When carefully implemented to take advantage of the sparsity, the updates scale sub-linearly in the number of topics, causing large improvements in high-dimensional regimes. For SCVB0, the minibatch updates are sparse in the rows (words), so some performance enhancements along the lines of those used by Mimno et al. are likely to be possible.

There has been a substantial amount of other work on speeding up LDA inference in the literature. Porteous *et al.* (2008) improved the efficiency of the sampling step for the collapsed Gibbs sampler, and Yao *et al.* (2009) explore a number of alternatives for improving the efficiency of LDA. The Vowpal Wabbit system for fast machine learning,[11] due to John Langford and collaborators, has a version of SVB that has been engineered to be extremely efficient. Parallelization is another approach for improving the efficiency of topic models. Newman *et al.* (2009) introduced an approximate parallel algorithm for LDA where data is distributed across multiple machines, and an exact algorithm for an extension of LDA which takes into account the distributed storage. Smola & Narayanamurthy (2010) developed an

efficient architecture for parallel LDA inference, using a distributed (key, value) storage for synchronizing the state of the sampler between machines. All of these computational improvements are somewhat orthogonal to those proposed in this paper, and it is likely that some of these ideas could be adapted to apply to SCVB0 as well.

4.8 Summary of Contributions

This chapter introduced SCVB0, an algorithm for performing fast stochastic collapsed variational inference in LDA, and showed that it outperforms stochastic VB on several large document corpora, converging faster and often to a better solution. The algorithm is relatively simple to implement, with intuitive update rules consisting only of basic arithmetic operations. We also found that the algorithm was effective at learning good topics from small corpora in seconds, finding topics that were superior than those of stochastic VB according to human judgement.

To summarize, this chapter has made the following contributions:

- We presented a fast, scalable algorithm for training topic models, called SCVB0. The algorithm performs variational Bayesian inference using a stochastic optimization technique, and operates on the collapsed representation of LDA. The algorithm is also simple to implement. Unlike the standard variational algorithm of Blei *et al.* (2003), and its stochastic extension (Hoffman *et al.* , 2010, 2013), it requires no expensive calls to complicated mathematical library functions. Furthermore, the core inner loop (the CVB0 update of Asuncion *et al.* (2009)) is similar to the update for the standard collapsed Gibbs sampling algorithm of Griffiths & Steyvers (2004), but is even simpler because it is deterministic.

- We evaluated the algorithm on three large corpora (New York Times news articles, the free online encyclopedia Wikipedia and scientific articles from PubMed Central), showing the benefit of the algorithm in terms of both predictive performance and wall-clock running time relative to the previous stochastic approach, and to standard batch VB algorithms.

- The algorithm was also evaluated in the context of the very rapid analysis of small-scale data for exploratory data analysis purposes, where a human is in the loop. In these experiments, human participants were asked to count the "mistakes" made by topic models. The test was performed using the New York Times dataset (with participants from the Amazon Mechanical Turk crowdsourcing system) and articles from the NIPS conference (with machine learning researchers as participants). In both cases, the human participants found fewer errors were made by the SCVB0 algorithm than the uncollapsed stochastic VB baseline.

- We suggested a new, simpler explanation for the accuracy of the CVB0 approximation. The explanation uses a law of large numbers argument, as opposed to the original central limit theorem and Taylor expansion approximations by Teh *et al.* (2007a) and Asuncion *et al.* (2009).

- We analyzed the algorithm from another perspective by providing an alternate derivation as an online EM algorithm for performing MAP estimation, with modified hyper-parameters.

- Using this alternative interpretation, we proved the convergence of the algorithm.

This chapter is joint work with Professor Max Welling, Dr Levi Boyles and Dr Christopher DuBois, published in Foulds *et al.* (2013). We thank them and acknowledge their contributions here:

- The original idea of using a stochastic algorithm in the collapsed representation of LDA is due to Professor Max Welling.
- Professor Welling discovered the Lyapunov function, although its derivation as an EM lower bound and the remainder of the convergence proof is due to the author of this thesis.
- Dr Boyles provided expertise in converting matlab code to an efficient implementation in the Julia language, and afforded much logistical support with the experiments. He is also responsible for the geometric series argument for the clumping update.
- Dr DuBois performed the Amazon Mechanical Turk experiment on the New York Times dataset.
- We thank Prof. Welling, Dr Boyles, Dr DuBois, and Dr Arthur Asuncion for many helpful discussions.

Chapter 5

Sampling Algorithms for Evaluating Topic Models

> Expectation is the root of all heartache.
> *Anonymous*

> A speedier course than lingering languishment
> must we pursue, and I have found the path.
> *William Shakespeare, Titus Andronicus*

An important property of topic models is that they can often play a useful role as building blocks for developing richer latent variable models. In Chapters 2 and 3, for example, we made use of the LDA framework to build models for data sets where both network and text information are available. Latent variable model-building based on LDA has become a widespread technique for finding meaningful latent structure, with broad applications to political science (Grimmer, 2010; Zhang & Carin, 2012), the humanities (Mimno, 2012, 2011), sociology (McFarland *et al.* , 2013), conversational dialog (Nguyen *et al.* , 2013), scientific

literature (Dietz *et al.* , 2007; Rosen-Zvi *et al.* , 2004; Chang & Blei, 2009; Foulds & Smyth, 2013) and more. With the growing adoption of these techniques, a number of algorithms have been proposed for fitting them accurately and efficiently on increasingly large data sets, including the algorithm we introduced in Chapter 4. The development of new models and algorithms is likely to continue into the foreseeable future.

Evaluation of Topic Models

As these new ideas continue to be proposed in the literature, it becomes increasingly important to *evaluate* them. Whenever we design a new sophisticated model, we need to ascertain whether its complexity is warranted. Whenever we propose a novel learning algorithm, we need to verify that it performs better than its competitors. To this end, a key measure of performance of any statistical model is its ability to generalize beyond the training data to *predict held-out data* (cf. Gneiting & Raftery (2007)). For example, we would like to be able to compute the likelihood of held-out documents according to the trained model, and compare this to the outputs of competing techniques.

However, in the case of topic models, and many other latent variable models, this presents a considerable computational challenge. The difficulty arises from the latent variables themselves. Each document d is associated with topic assignments $\mathbf{z}^{(d)}$ and a distribution over topics $\theta^{(d)}$. These variables are latent, i.e. they are hidden from us. We therefore must consider every possible assignment to them in order to compute the likelihood of a held-out document:

$$Pr(w^{(d)}|\mathbf{\Phi}, \alpha) = \sum_{\mathbf{z}^{(d)}} Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}, \alpha) \tag{5.1}$$

$$= \int Pr(w^{(d)}, \theta^{(d)}|\mathbf{\Phi}, \alpha)\, d\theta^{(d)} \ . \tag{5.2}$$

The sum in Equation 5.1 is intractable because the number of possible assignments of $\mathbf{z}^{(d)}$ grows exponentially in the number of words in the document. Similarly, the integral in Equation 5.2 is hard to compute because we cannot exploit conjugacy to analytically integrate out $\theta^{(d)}$ when we do not observe $\mathbf{z}^{(d)}$. Moreover, this already difficult computation must be performed for every document in the held-out test set, which frequently contains hundreds to thousands of documents. To make matters even more difficult, when we are evaluating learning algorithms we would also like to repeat the entire process at many points of the training process, in order to show the progress of the learning algorithm over time. The evaluation task for topic models, then, frequently corresponds to the computation (or, more likely, approximation) of possibly *tens of thousands of intractable integrals*.

To put the computational challenge in context, we have seen in the previous chapter that it is now possible to learn topic models on extremely large datasets on a single core processor in a matter of hours. For example, using SCVB0, as implemented in a high-level language, we trained a topic model on Wikipedia to convergence in less than twelve hours. However, to create Figure 4.5, which evaluates the progress of this SCVB0 training run, we needed to use a cluster to parallelize the approximate evaluation process across documents and iterations. Ironically, the time taken to evaluate the model was orders of magnitude greater than the time it took to train it. Furthermore, for computational reasons, that experiment was performed using a simpler evaluation task called *document completion*, where the goal is to predict part of the document, given the remainder. It would be preferable to be able to fully predict the held-out documents as in Equation 5.1, but at this scale, with 10,000 held-out documents and hundreds of training time periods at which to evaluate, this was infeasible with current techniques.

In order to make a practical reality of the evaluation of topic models via $Pr(w^{(d)}|\mathbf{\Phi}, \alpha)$, a wide variety of approximation strategies have previously been proposed in papers such as Wallach *et al.* (2009b), Buntine (2009) and Scott & Baldridge (2013). The techniques

typically involve Monte Carlo simulation schemes designed to address the challenges of high-dimensional integration. Although these methods can lead to significantly more accurate results than naive approaches, the reliable and efficient evaluation of topic models remains a relatively open problem of practical significance.

Contributions of this Chapter

In this chapter, we investigate new methods for the evaluation of topic models. The proposed techniques are based on *annealed importance sampling* (*AIS*) (Neal, 2001), a Monte Carlo integration technique which was previously applied to topic model evaluation by Wallach *et al.* (2009b). Given two probability distributions, AIS produces an estimate of the ratio of their partition functions by annealing between them. Wallach et al. leverage this idea by annealing from the prior over the latent topic assignments $\mathbf{z}^{(d)}$ to the posterior, resulting in an estimate of the probability of the held-out document. AIS can be very accurate given enough computation time, although the amount of time needed may vary greatly between different choices of annealing paths (Grosse *et al.* , 2013).

The first contribution of this chapter is to propose and evaluate an alternative annealing strategy, using two AIS paths which anneal from one topic model to another. This strategy (referred to as ratio-AIS) computes the *ratio* of the likelihoods of two models instead of computing the likelihoods of each model separately. The result is an estimate of the relative performance of the models, with significantly lower empirical variance across runs than previous approaches.[1] This in turn brings computational benefits, as fewer samples or annealing temperatures may be required to achieve reliable results. The reduced variance comes at the cost of potentially increased bias when insufficient iterations are performed to achieve convergence. However, we also show how to detect such bias by annealing between

[1] "Variance" here refers to variance across Monte Carlo estimates of the difference in log-likelihood between models, per document.

the topic models in both directions and comparing the results. The consequence of this bias-variance trade-off is that the proposed method is useful in cases where we would like to perform in-depth analysis at the per-document level and when the two topic models are similar to each other. The high-variance low-bias methods may still be preferred for general full-corpus comparisons of topic models.

Finally, we show how to efficiently evaluate topic model learning algorithms by computing held-out likelihood curves over the iterations of the learning procedure, making use of the proposed ratio-AIS paths. This is achieved by annealing between the topic models at each iteration of the learning algorithm in turn, which allows all previous computation to be reused in each of the likelihood estimates. The proposed method, called *iteration-AIS*, outperforms previous algorithms, in some cases even when it is given an order of magnitude less computation time.

The remainder of the chapter proceeds as follows. In Section 5.1 we discuss in more detail the different methods that are typically used for topic model evaluation. Section 5.2 introduces our new proposed evaluation algorithms. Experimental results are given in in Section 5.3. As an aside, we then consider the relationship between one of our proposed techniques and a learning algorithm due to Asuncion *et al.* (2010), and speculate on potential improvements to that technique (Section 5.4). Finally, we conclude the chapter in Section 5.6.

Note that although we focus on topic models, the ideas presented here could potentially be useful for other latent variable models with intractable likelihoods.

5.1 Background

When proposing a new topic model or learning algorithm, it is important to evaluate its performance. When the model is to be used for a certain task it may be possible to evaluate it

with respect to an extrinsic, task-specific metric. For example one could evaluate the quality of topics being used as features for a classification algorithm by measuring classification accuracy. More generally, however, given that topic models are typically trained in an unsupervised manner (with a few notable exceptions), a ground-truth evaluation metric is usually not available.

Consequently, a number of intrinsic (i.e. task independent) validation strategies for topic models have been developed in the literature. For example, Chang *et al.* (2009) proposed the use of elicitations of judgments from humans to evaluate the quality of topic models. Given that obtaining these judgments can be expensive and difficult, Newman *et al.* (2010), Mimno *et al.* (2011) and Rosner *et al.* (2013) proposed automatic surrogate measures of topic coherence, and showed that these measures, which are typically based on word co-occurrence statistics, are correlated with human judgments.

However, as topic models are statistical models, we also would like to be able evaluate them as such. In the context of unsupervised machine learning, the standard approach for evaluating a statistical model is to compute the probability of held-out data. Regardless of the utility of the aforementioned methods, it is generally useful to demonstrate good predictive performance in addition to any other extrinsic or intrinsic validation results. Intuitively, as our goal is to fit a statistical model to data, we would like to know both how well we are able to fit the model, and how well the model is able to explain unobserved data.

5.1.1 Computing the Likelihood

As in Wallach *et al.* (2009b), we therefore focus on the computation of $Pr(w^{(d)}|\mathbf{\Phi}, \alpha)$, the likelihood of the words $w^{(d)}$ in a held out document d (or equivalently, perplexity), conditioned on point estimates of the topic-word distributions $\mathbf{\Phi}$ and (possibly document specific)

priors α.[2] This quantity can be used to evaluate a point estimate of the topics, or as an inner loop to evaluate Bayesian evaluation metrics such as the posterior predictive probability of held out documents.

Simple Monte Carlo Algorithms

As discussed above, it is in general infeasible to compute $Pr(w^{(d)}|\mathbf{\Phi}, \alpha)$ directly, as it involves an intractable sum $\sum_{\mathbf{z}^{(d)}} Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}, \alpha)$ or an intractable integral $\int_\theta Pr(w^{(d)}, \theta^{(d)}|\mathbf{\Phi}, \alpha)$. Consequently, an approximation strategy must be used. A variety of approximation techniques were considered by Wallach *et al.* (2009b), the number of which alone is a testament to the difficulty of the problem. The simplest strategies are Monte Carlo approaches which simulate from the prior. We can write Equation 5.1 as

$$Pr(w^{(d)}|\mathbf{\Phi}, \alpha) = \sum_{\mathbf{z}^{(d)}} Pr(w^{(d)}|\mathbf{z}^{(d)}, \mathbf{\Phi}) Pr(\mathbf{z}^{(d)}|\alpha) \tag{5.3}$$

$$= E_{Pr(\mathbf{z}^{(d)}|\alpha)}[Pr(w^{(d)}|\mathbf{z}^{(d)}, \mathbf{\Phi})] \tag{5.4}$$

$$\approx \frac{1}{S} \sum_{i=1}^{S} Pr(w^{(d)}|\mathbf{z}^{(d)(i)}, \mathbf{\Phi}) \ , \tag{5.5}$$

where $\mathbf{z}^{(d)(i)} \sim Pr(\mathbf{z}^{(d)}|\alpha)$ is the ith sample of $\mathbf{z}^{(d)}$, and Equation 5.5 follows from the law of large numbers for sufficiently large S. Unfortunately, for any realistically sized document, drawing from the prior is very unlikely to find the $\mathbf{z}^{(d)}$'s for which $Pr(w^{(d)}|\mathbf{z}^{(d)}, \mathbf{\Phi})$ is high. This method consequently works poorly in practice, tending to greatly underestimate the likelihood (Wallach *et al.* , 2009b). A similar Monte Carlo strategy can be applied based on Equation 5.5 by sampling $\theta^{(d)}$, but this encounters similar problems to Equation 5.5.

[2]It is standard practice to learn an asymmetric Dirichlet prior α in LDA models, following Wallach *et al.* (2009a), so we include it as a parameter to evaluate. The prior may also be learned in a document dependent way for models such as DMR (Mimno & McCallum, 2008).

It should also be noted that Equation 5.4 shows why we cannot directly average over draws from the standard collapsed Gibbs sampler for $Pr(\mathbf{z}^{(d)}|w^{(d)}, \mathbf{\Phi}, \alpha)$ to estimate the held-out probability, as the expectation we are trying to compute is with respect to the prior over $\mathbf{z}^{(d)}$, not the posterior. It turns out that taking the *harmonic mean* of the predictions of samples from the posterior is an unbiased estimator,

$$Pr(w^{(d)}|\mathbf{\Phi}, \alpha) \approx \frac{1}{\frac{1}{S}\sum_{i=1}^{S} \frac{1}{Pr(w^{(d)}|\mathbf{z}^{(d)(i)}, \mathbf{\Phi}, \alpha)}} \ , \qquad \mathbf{z}^{(d)(i)} \sim Pr(\mathbf{z}^{(d)}|w^{(d)}, \mathbf{\Phi}, \alpha) \ , \tag{5.6}$$

in a technique due to Newton & Raftery (1994). However, this estimator is very unstable and Wallach *et al.* (2009b) found that it tended to give very different answers when compared to other methods, typically greatly overestimating the likelihood of the held-out documents.

Left-to-Right Sampler

The most widely used of the approaches proposed by Wallach *et al.* (2009b) is the "left-to-right" algorithm, inspired by sequential Monte Carlo techniques. This method uses the product rule to factorize $Pr(w^{(d)}, \theta^{(d)}|\mathbf{\Phi}, \alpha)$ sequentially,

$$\begin{aligned}
Pr(w^{(d)}|\mathbf{\Phi}, \alpha) &= \prod_{n=1}^{N_d} Pr(w_n^{(d)}|w_{<n}^{(d)}, \mathbf{\Phi}, \alpha) && (5.7)\\
&= \prod_{n=1}^{N_d} \sum_{\mathbf{z}_{\leq n}^{(d)}} Pr(w_n^{(d)}, \mathbf{z}_{\leq n}^{(d)}|w_{<n}^{(d)}, \mathbf{\Phi}, \alpha) && (5.8)\\
&= \prod_{n=1}^{N_d} \sum_{\mathbf{z}_{\leq n}^{(d)}} Pr(w_n^{(d)}|\mathbf{z}_n^{(d)}, \mathbf{\Phi}) Pr(\mathbf{z}_{\leq n}^{(d)}|w_{<n}^{(d)}, \mathbf{\Phi}, \alpha) && (5.9)\\
&= \prod_{n=1}^{N_d} E_{Pr(\mathbf{z}_{\leq n}^{(d)}|w_{<n}^{(d)}, \mathbf{\Phi}, \alpha)}\Big[Pr(w_n^{(d)}|\mathbf{z}_n^{(d)}, \mathbf{\Phi})\Big] && (5.10)\\
&\approx \prod_{n=1}^{N_d} \frac{1}{S} \sum_{i=1}^{S} Pr(w_n^{(d)}|\mathbf{z}_n^{(d)(i,n)}, \mathbf{\Phi}) \ , \ \mathbf{z}_{\leq n}^{(d)(i,n)} \sim Pr(\mathbf{z}_{\leq n}^{(d)}|w_{<n}^{(d)}, \mathbf{\Phi}, \alpha) \ , && (5.11)
\end{aligned}$$

where $\mathbf{z}^{(d)(i,n)}$ is the ith sample of $\mathbf{z}^{(d)}$ at iteration n, which represents the $\mathbf{z}$ vector up to the nth word, and the subscripts specify the word indices. This allows sampling to be performed in an incremental "left to right" fashion, estimating each of the terms in 5.10 in turn. At iteration n, which processes the nth word in the document, the algorithm maintains S samples (a.k.a. "*particles*") $\mathbf{z}_{\leq n}^{(d)(i,n)}$. Each iteration performs a collapsed Gibbs sampling sweep conditioned on $w_{<n}^{(d)}$ over each of the samples $\mathbf{z}_{\leq n-1}^{(d)(i,n-1)}$ from the previous iteration, then increments all particles to include the topic assignment for the current word $\mathbf{z}_n^{(d)}$. Averaging over all of the particles at word n approximates one of the expectation terms in the product in Equation 5.10.[3,4] The algorithm was analyzed more closely by Buntine (2009), and a faster, but less accurate, variant of the technique was proposed by Scott & Baldridge (2013). Intuitively, this algorithm is likely to select much better $\mathbf{z}$'s than the naive Monte Carlo method because it is allowed to make use of the previous words instead of drawing blind from the prior. A disadvantage of this approach is that the "*resampling*" Gibbs sweep in each iteration makes the algorithm's run time quadratic in the length of the document.

5.1.2 Importance Sampling

Many of the approaches proposed by Wallach *et al.* (2009b) and Buntine (2009), and also the new methods introduced in this chapter, involve *importance sampling*, an approximate algorithm for computing expectations. Suppose we would like to compute $E_{p(\mathbf{x})}[f(\mathbf{x})]$ for

[3]Wallach *et al.* (2009b) describe the derivation of the left-to-right sampler up to Equation 5.8. We complete the derivation up to Equation 5.11 here, resulting in essentially the same algorithm, but with one small difference. Wallach *et al.* (2009b) and Buntine (2009) draw the topic assignment of the current word $\mathbf{z}_n^{(d)(i,n)} \sim Pr(\mathbf{z}_n^{(d)}|w_n^{(d)}, \mathbf{z}_{<n}^{(d)}, \mathbf{\Phi}, \alpha)$ when estimating the expectation in Equation 5.10. Our derivation here finds that the current word should not be used, and it should instead be drawn $\mathbf{z}_n^{(d)(i,n)} \sim Pr(\mathbf{z}_n^{(d)}|\mathbf{z}_{<n}^{(d)}, w_{<n}^{(d)}, \mathbf{\Phi}, \alpha) = Pr(\mathbf{z}_n^{(d)}|\mathbf{z}_{<n}^{(d)}, \alpha)$, because the expectation is conditioned on *previous* words only. To verify the issue, note that for a document containing a single word, conditioning on the current word corresponds to drawing $\mathbf{z}^{(d)}$ from the posterior, while Equation 5.4 shows that it should be drawn from the prior.

[4]As an aside, one potential improvement to the algorithm may be to sum out the current topic assignment via $Pr(\mathbf{z}_n^{(d)} = k|\mathbf{z}_{<n}^{(d)}, \alpha) \propto n_k^{(d)} + \alpha_k$ instead of sampling it, thus considering all possible values with a very small additional overhead relative to the resampling step.

some function $f(\mathbf{x})$ and some distribution $p(\mathbf{x})$ which we cannot sample from directly. The key idea of importance sampling is to approximate drawing from p by drawing from another, more manageable distribution q, and then re-weight the samples to correct for this process. We can derive this procedure by

$$\begin{aligned} E_{p(\mathbf{x})}[f(\mathbf{x})] &= \sum_{\mathbf{x}} p(\mathbf{x}) f(\mathbf{x}) \\ &= \sum_{\mathbf{x}} q(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} f(\mathbf{x}) \\ &= E_{q(\mathbf{x})}\Big[\frac{p(\mathbf{x})}{q(\mathbf{x})} f(\mathbf{x})\Big] \\ &\approx \frac{1}{S} \sum_{i=1}^{S} \omega_i f(\mathbf{x}^{(i)}) \ , \qquad \mathbf{x}^{(i)} \sim q(\mathbf{x}) \ , \end{aligned} \tag{5.12}$$

where $\omega_i = \frac{p(\mathbf{x})}{q(\mathbf{x})}$ is the *importance weight* of sample i, and it is assumed that $q(\mathbf{x}) > 0$ whenever $p(\mathbf{x}) > 0$. The algorithm consists of simulating Equation 5.12. In some cases we can only compute unnormalized versions $p^*(\mathbf{x})$ and $q^*(\mathbf{x})$ of $p(\mathbf{x})$ and $q(\mathbf{x})$, where $q(\mathbf{x}) = \frac{q^*(\mathbf{x})}{Z_q}$ and $Z_q = \sum_{\mathbf{x}} q^*(\mathbf{x})$. In this scenario, a ratio of two estimates is used,

$$\begin{aligned} E_{p(\mathbf{x})}[f(\mathbf{x})] &= \sum_{\mathbf{x}} \frac{p^*(\mathbf{x})}{Z_p} f(\mathbf{x}) \\ &= \sum_{\mathbf{x}} \Big(q(\mathbf{x}) \frac{Z_q}{q^*(\mathbf{x})}\Big) \frac{p^*(\mathbf{x})}{Z_p} f(\mathbf{x}) \\ &= \frac{Z_q}{Z_p} E_{q(\mathbf{x})}\Big[\frac{p^*(\mathbf{x})}{q^*(\mathbf{x})} f(\mathbf{x})\Big] \\ &= E_{q(\mathbf{x})}\Big[\frac{p^*(\mathbf{x})}{q^*(\mathbf{x})} f(\mathbf{x})\Big] \Big/ \frac{Z_p}{Z_q} \ , \text{ where} \\ \frac{Z_p}{Z_q} &= \sum_{\mathbf{x}} \frac{p^*(\mathbf{x})}{Z_q} = \sum_{\mathbf{x}} \Big(q(\mathbf{x}) \frac{Z_q}{q^*(\mathbf{x})}\Big) \frac{p^*(\mathbf{x})}{Z_q} = E_{q(\mathbf{x})}[\frac{p^*(\mathbf{x})}{q^*(\mathbf{x})}] \ , \text{ so} \end{aligned} \tag{5.13}$$

$$E_{p(\mathbf{x})}[f(\mathbf{x})] \approx \frac{1}{S} \sum_{i=1}^{S} \omega_i f(\mathbf{x}^{(i)}) \Big/ \frac{1}{S} \sum_{i=1}^{S} \omega_i \tag{5.14}$$

$$= \sum_{i=1}^{S} \omega_i f(\mathbf{x}^{(i)}) \Big/ \sum_{i=1}^{S} \omega_i \ , \tag{5.15}$$

with $\mathbf{x}^{(i)} \sim q(\mathbf{x})$ and with importance weights computed using the unnormalized values, $\omega_i = \frac{p^*(\mathbf{x})}{q^*(\mathbf{x})}$. It should be noted that although each of the two Monte Carlo estimates (the numerator and the denominator) are unbiased, their ratio is biased for a finite number of samples. Nevertheless, the strong law of large numbers applies and so asymptotically the algorithm will recover $E_{p(\mathbf{x})}[f(\mathbf{x})]$. Importance sampling can also provide an estimate of the ratio of these normalizing constants ("*partition functions*") as the average of the importance weights,

$$\frac{1}{S}\sum_{i=1}^{S}\omega_i \approx E_{q(\mathbf{x})}\frac{p^*(\mathbf{x})}{q^*(\mathbf{x})} = \sum_{\mathbf{x}} q(\mathbf{x})\frac{p^*(\mathbf{x})}{Z_q q(\mathbf{x})} = \frac{1}{Z_q}\sum_{\mathbf{x}} p^*(\mathbf{x}) = \frac{Z_p}{Z_q} \; . \tag{5.16}$$

We will leverage this fact in the new methods introduced later in this chapter.

In the context of topic model evaluation, Buntine (2009) used mean field variational inference to select a proposal distribution for importance sampling the latent topic assignments. Buntine found that this method did not perform as well as the left-to-right algorithm, although it was considerably faster. More closely related to the present work, an importance sampling scheme called *annealed importance sampling* was one of the more accurate strategies investigated by Wallach *et al.* (2009b). This technique, which we describe below, forms the basis of the methods proposed in this chapter.

5.1.3 Annealed Importance Sampling

When performing importance sampling with high dimensional data, each individual importance sample is unlikely to land in a high probability region unless the proposal distribution is very good. Thus, the variability of the importance weights can be large. In practice, this frequently results in the estimate being determined almost exclusively by a small set of samples with the highest weights. This renders the procedure unreliable. An alternative approach is to use Markov chain Monte Carlo methods. When run to convergence, these

methods will eventually sample correctly from the distribution of interest, even in high dimensions. On the other hand, convergence assessment is difficult, so it is not always possible to know whether the reported results are meaningful. The samples are also dependent, which leads to issues when sampling from distributions with multiple modes.

Annealed importance sampling (*AIS*) (Neal, 2001) is an attempt to find a middle ground, by using MCMC to select a proposal distribution for importance sampling. In the method, a Markov chain technique provides a mechanism for drawing samples from a high dimensional distribution, and importance sampling is used to correct for convergence failures in the Markov chain.

As in traditional importance sampling, suppose we are interested in estimating an expectation of some function of a random variable $\mathbf{x}$ with respect to an intractable distribution of interest p_0. Consider a distribution p_n (which is typically easy to sample from) and a sequence of "intermediate" distributions $p_{n-1}, \ldots, p_1$ leading from p_n to p_0. Using a physical metaphor, the intuition is that p_n is a "high temperature" distribution, i.e. it is easy to move very quickly through the sample space by simulating it with a Markov chain. The target distribution p_0, on the other hand, is assumed difficult to sample from, and so is generally a "low temperature" distribution whose Markov chain mixes poorly.

In order to sample from p_n, the AIS algorithm simulates an *annealing* (i.e. controlled cooling) process by reducing the "temperature" from p_n towards p_0, moving a set of samples from p_n through Markov chain updates from each of the successively cooler intermediate distributions.

One could also imagine a "heating" process, which operates in the reverse direction to the cooling process. This process begins with the low temperature distribution and increases the temperature towards p_n. In this reversed process, the first state is a draw from p_0, which is what we desire. The AIS algorithm simulates from the cooling process, and it then uses

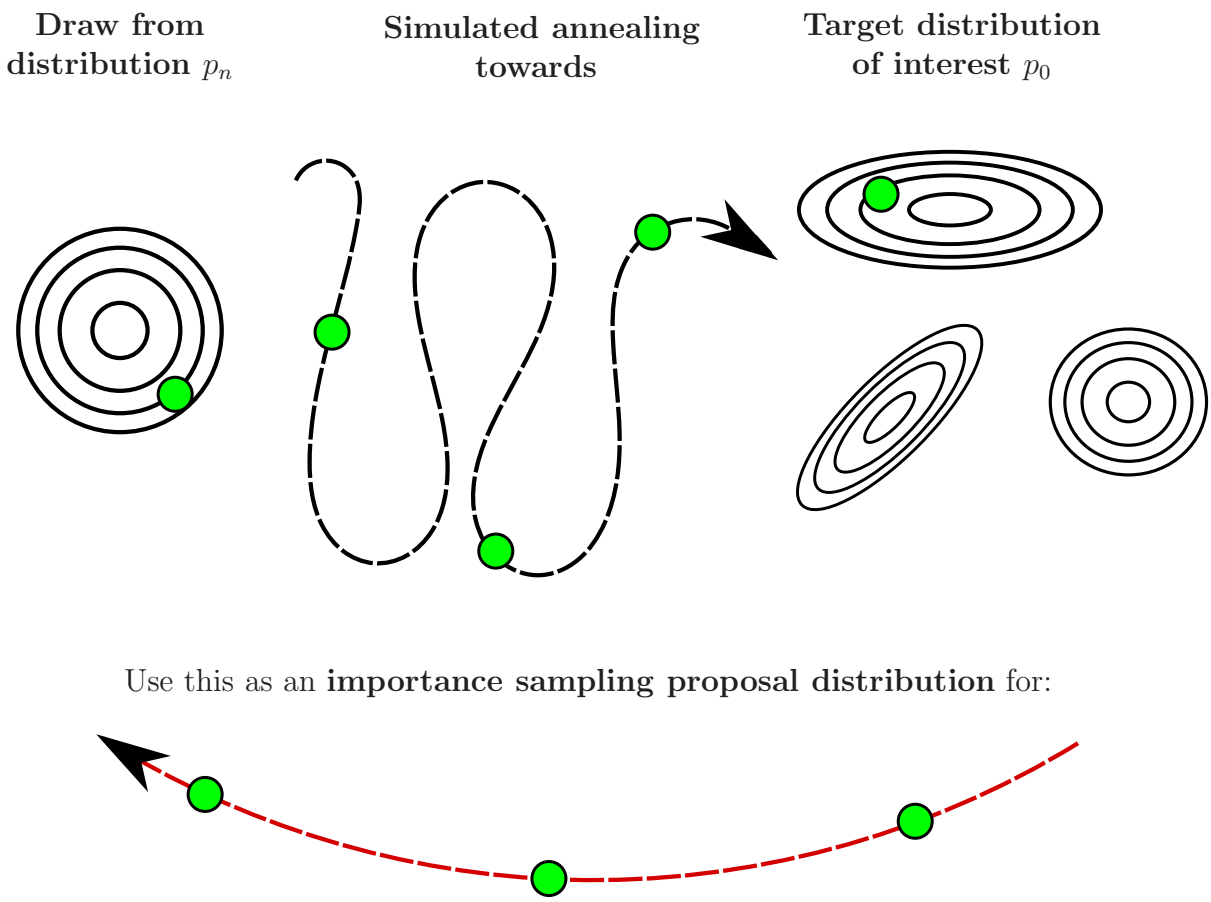


Figure 5.1: Annealed Importance Sampling

the entire cooling simulation as a proposal distribution in an importance sampling scheme where the target distribution is the sequence of states drawn from the "heating" process. The resulting importance weights correct for the fact that the annealing process was used, giving importance weighted samples of p_0. Figure 5.1 illustrates the AIS method.

In more detail, let us assume that for each intermediate distribution p_j we have a Markov chain with transition operator $T_j(\mathbf{x}, \mathbf{x}')$ which is invariant to that distribution. We need to be able to sample from these Markov chains, and for each p_j be to able to evaluate some function f_j which is proportional to it. In a manner similar to that of traditional importance sampling, AIS produces a collection of samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(S)}$ with associated importance

weights $\omega_1, \ldots, \omega_S$. As with importance sampling, the expectation of interest is estimated using the samples, weighted by the importance weights.

The strategy for drawing each sample $\mathbf{x}^{(i)}$ is to begin by drawing a sample $\mathbf{x}_{n-1}$ from p_n, then drawing a sequence of points $\mathbf{x}_{n-2}, \ldots, \mathbf{x}_0$ which "anneal" towards p_0.[5] Each of the remaining $\mathbf{x}_j$'s in the sequence are generated from $\mathbf{x}_{j+1}$ via T_j. Importance weights ω_i are computed by viewing $(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1})$ as an augmented state space, and performing importance sampling on this new state space. The above procedure is used as a proposal distribution Q for importance sampling from another distribution P:

$$Q(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) \propto f_n(\mathbf{x}_{n-1}) \prod_{j=n-1}^{1} T_j(\mathbf{x}_j, \mathbf{x}_{j-1}) \tag{5.17}$$

$$P(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) \propto f_0(\mathbf{x}_0) \prod_{j=1}^{n-1} \tilde{T}_j(\mathbf{x}_{j-1}, \mathbf{x}_j) \ , \tag{5.18}$$

where $\tilde{T}_j(\mathbf{x}, \mathbf{x}') = T_j(\mathbf{x}', \mathbf{x}) \frac{f_j(\mathbf{x}')}{f_j(\mathbf{x})}$ is the reversal of the transition defined by T_j. This leads to importance weights for each of the samples,

$$\omega_i = \frac{P(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1})}{Q(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1})} = \prod_{j=0}^{n-1} \frac{f_j(\mathbf{x}_j)}{f_{j+1}(\mathbf{x}_j)} \ . \tag{5.19}$$

Note that the marginal probability of $\mathbf{x}_0$ under P is $p_0(\mathbf{x}_0)$, so after letting $\mathbf{x}^{(i)} = \mathbf{x}_0$ the procedure correctly carries out importance sampling from p_0. Since it is an instance of importance sampling, AIS also provides an estimate for the ratio of normalizing constants for f_0 and f_n. The normalizing constant for P is the same as the normalizing constant for f_0, and the normalizing constant for Q is the same as the normalizing constant for f_n, and so the average of the importance weights, $S^{-1} \sum_{i=1}^{S} \omega_i$, converges to $\frac{\int f_0(\mathbf{x}) d\mathbf{x}}{\int f_n(\mathbf{x}) d\mathbf{x}}$ by Equation 5.16.

[5] For the remainder of the chapter, subscripts j and n will refer to a temperature, rather than indexing into an array.

5.1.4 AIS for Topic Models

Wallach *et al.* (2009b) showed how to apply the AIS procedure to the problem of calculating LDA likelihoods. They observe that the likelihood is the normalizing constant for the posterior over $\mathbf{z}$, when written as the joint distribution. By the definition of conditional probability,

$$Pr(\mathbf{z}^{(d)}|w^{(d)}, \mathbf{\Phi}, \alpha) = \frac{Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}, \alpha)}{Z(\mathbf{\Phi}, \alpha)} \ , \qquad Z(\mathbf{\Phi}, \alpha) = Pr(w^{(d)}|\mathbf{\Phi}, \alpha) \ . \tag{5.20}$$

In this context, $Pr(w^{(d)}|\mathbf{\Phi}, \alpha)$ is called the *marginal likelihood*, or sometimes, the *evidence* (because we are conditioning on observed evidence $w^{(d)}$ to compute the posterior). The marginal likelihood of a test document for a topic model can be estimated by using AIS to estimate a normalization constant, operating on the latent topic assignments $\mathbf{z}^{(d)}$ for the document.[6] We can set

$$f_0 = Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}, \alpha) \tag{5.21}$$

$$f_n = Pr(\mathbf{z}^{(d)}|\alpha) \ , \tag{5.22}$$

with intermediate distributions

$$f_j = Pr(w^{(d)}|\mathbf{z}^{(d)}, \mathbf{\Phi}, \alpha)^{\beta_j} f_n \tag{5.23}$$

[6]The derivation here differs slightly from that of Wallach *et al.* (2009b). The present derivation suggests that the procedure described in Wallach et al. produces just one importance sample. This may be repeated many times, finally producing as output the average of the resulting importance weights. In practice however, we found that a single sample with a longer annealing run, as in Wallach et al., may still be the best strategy on a budget.

and the transition operators T_j being the Gibbs sampler for f_j. The ratio of normalizing constants is

$$\frac{1}{S}\sum_{i=1}^{S}\omega_i \approx \frac{\sum_{\mathbf{z}^{(d)}} Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}, \alpha)}{\sum_{\mathbf{z}^{(d)}} Pr(\mathbf{z}^{(d)}|\alpha)}$$

$$= \frac{Pr(w^{(d)}|\mathbf{\Phi}, \alpha)}{1} \tag{5.24}$$

$$= Pr(w^{(d)}|\mathbf{\Phi}, \alpha) \ . \tag{5.25}$$

The procedure for producing each importance sample, then, is to draw an initial $\mathbf{z}^{(d)}$ from the prior, and anneal it towards f_0 by performing $r_j \geq 1$ Gibbs iterations at each intermediate distribution. After repeating this procedure for each sample, the likelihood is estimated as the average of the importance weights. Note that in what follows we define a *run* as the full procedure, averaged over importance samples, while a *sample* refers to a single importance sample.

5.1.5 Document Completion

As an alternative to computing $Pr(w^{(d)}|\mathbf{\Phi}, \alpha)$, a strategy for side-stepping some of the computational difficulty is to instead estimate (or sample) $\theta^{(d)}$ on a subset $w^{(d,a)}$ of the document, and predict only the remaining portion of the document $w^{(d,b)}$, thus estimating $Pr(w^{(d,b)}|w^{(d,a)}, \mathbf{\Phi}, \alpha)$. Having recovered an estimate $\hat{\theta}^{(d)}$ of $\theta^{(d)}$ based on $w^{(d,a)}$, the *document completion* probability can then be approximated via

$$Pr(w^{(d,b)}|w^{(d,a)}, \mathbf{\Phi}, \alpha) \approx Pr(w^{(d,b)}|\hat{\theta}^{(d)}, \mathbf{\Phi}) \ . \tag{5.26}$$

This method is frequently used in practice (e.g. Rosen-Zvi *et al.* (2004); Asuncion *et al.* (2009); Wallach *et al.* (2009a), and in Chapter 4 of this thesis). It is computationally and conceptually easier than evaluating the full likelihood, because any algorithm for estimating

θ (or a distribution over θ) such as standard VB or the Rao-Blackwellized Gibbs sampling estimator can be applied to the first half of the document. In many cases it may be "good enough" to determine a difference between methods, as we saw in Chapter 4. However, Wallach *et al.* (2009b) found that it was able to detect less of a difference between ground truth synthetic topics and perturbed copies than left-to-right and AIS (as applied to the document completion task). They also reported that the "estimated θ" strategy performed poorly relative to AIS, by significantly underestimating the likelihood. Intuitively, there is a relatively small amount of data in a single document with which to learn a point estimate of $\theta^{(d)}$, which is one of the key motivations of the original LDA paper (Blei *et al.* , 2003) as a Bayesian extension of PLSA. This problem can be mitigated by drawing multiple samples of $\theta^{(d)}$, but this reduces the computational advantages of the document completion approach.

Another issue with the "document completion" scenario is that it changes the task somewhat, making it not necessarily the gold standard prediction task we would like it to be. It measures the ability of the model to "orient" itself quickly when given partial documents, rather than how likely the overall document is under the model.

The widespread use of the document completion strategy may be largely due to its convenient computational properties (leading in turn to its use as a surrogate for fully held-out prediction), rather being due to any intrinsic benefit of the metric itself. It is unclear how the use of document completion as a surrogate for full-document prediction might affect our conclusions, particularly when using topic models which learn the Dirichlet hyper-parameter α, as recommended by Wallach *et al.* (2009a), and including Dirichlet multinomial regression models (Mimno & McCallum, 2008) and the Topical Influence Regression model of Chapter 3 (which we evaluated using AIS).

Specifically, learning α, especially with a per-document $\alpha^{(d)}$, may help the model to recover $\theta^{(d)}$ better on the training portion of the document, thus increasing the performance of the model for "estimated θ" document completion more than in the fully held-out case. On the

other hand, observing more of the document decreases the relative impact of the prior on the posterior distribution, which could reduce the observed improvement due to learning α. Thus, we suspect that document completion may not always be a good surrogate for the full prediction task. It should be noted that many methods for fully held-out prediction can also be adapted for the document completion task (including those proposed in this chapter).

5.2 Alternative Annealing Paths for the Evaluation of Topic Models

Having reviewed the necessary background material, the remainder of this chapter details our contributions. First, we note that the AIS method described above can be very accurate if given enough computation time, leading Wallach *et al.* (2009b) to use the procedure as a gold standard approach. However, it is subject to several potentially avoidable sources of variability. The method computes the ratio of the desired quantity $Pr(w^{(d)}|\mathbf{\Phi}, \alpha)$ and a quantity which equals one, so stochastic noise is introduced due to the denominator, even though this is a constant. We would also expect that the prior may typically be very different from the posterior, thereby requiring many annealing iterations to prevent the importance weights ω_i from having a large variance. This has consequences for the efficiency of the sampler, which is reduced by a factor of approximately $1 + \text{Var}_q[\omega_i / E_q[\omega_i]]$ relative to direct sampling from the target density (Neal, 2001).[7]

Making matters worse, we typically must perform the AIS procedure many times across all held-out documents, and therefore have a relatively limited computational budget per document, preventing us from compensating for the high variance by collecting many importance samples with a large number of temperatures. In this section, we introduce new

[7]Note that $E_q[\omega_i]$ is equal to the ratio of normalizing constants of the target and proposal densities, which in our case is the quantity of interest, e.g. the likelihood.

AIS annealing paths for the evaluation of topic models which can have lower variance than the standard approach. We first introduce AIS paths which compare two topic models by annealing between them. We then show how to use these paths for evaluating topic model learning algorithms by computing per-iteration predictive performance efficiently, reusing all previous computation.

5.2.1 Comparing Topic Models by Annealing Between Them

The most typical evaluation scenario is model comparison—we want to determine whether a particular model (model 1) performs better at predicting held-out documents than a baseline method (model 2) such as vanilla LDA or a model trained using a previous learning algorithm. Thus, in such situations, the quantity of interest is the *relative* log-likelihood score of the model and the baseline:

$$\begin{aligned} &\log Pr(w^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(1)}) - \log Pr(w^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(2)}) \\ &\quad = \log \frac{Pr(w^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(1)})}{Pr(w^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(2)})} \ . \end{aligned} \tag{5.27}$$

To compute this in the framework proposed by Wallach et al., we must perform the AIS procedure once for each model, incurring the stochastic error twice. To avoid this and the aforementioned sources of variability, and given that the procedure is already designed to compute a ratio, we propose to instead use AIS to compute Equation 5.27 directly. Let $f_0(\mathbf{z}^{(d)}) = Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(1)})$ and $f_n(\mathbf{z}^{(d)}) = Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(2)})$. Then the desired quantity can be estimated via

$$\frac{1}{S}\sum_{i=1}^{S} \omega_i \approx \frac{\sum_{\mathbf{z}^{(d)}} Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(1)})}{\sum_{\mathbf{z}^{(d)}} Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(2)})} = \frac{Pr(w^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(1)})}{Pr(w^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(2)})} \ . \tag{5.28}$$

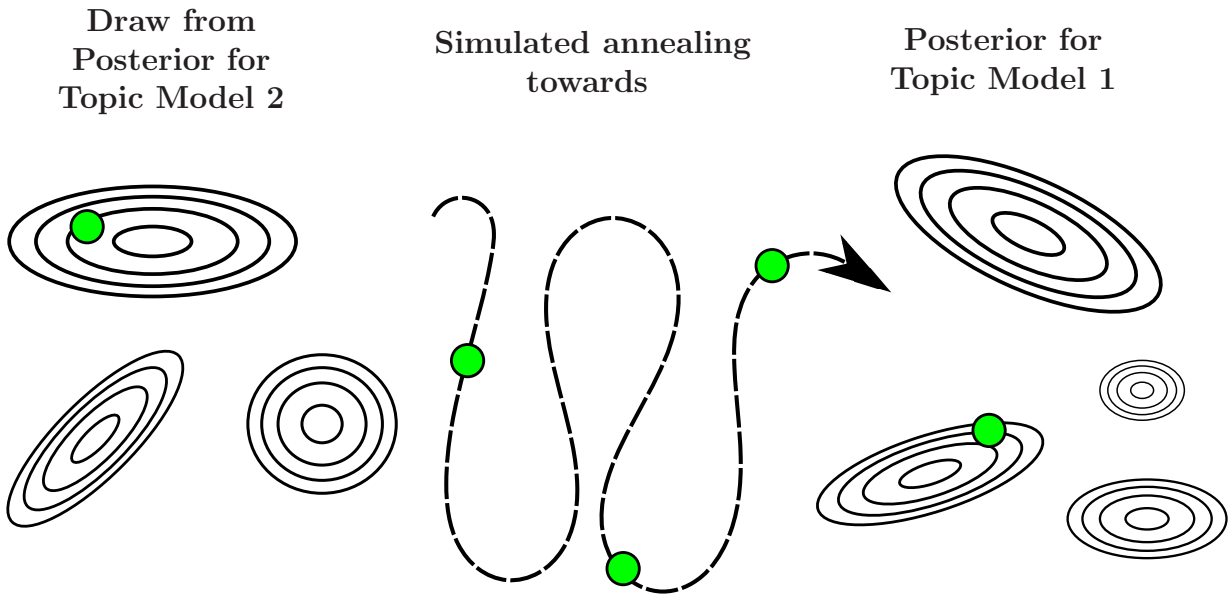


Figure 5.2: Ratio-AIS

We will refer to this strategy as "*ratio-AIS*." An illustration of the method is given in Figure 5.2. Note that although this is a direct application of AIS, we must modify our intuition regarding the physical interpretation of the method. In a standard application of AIS, the target distribution is at a "low temperature," i.e. it is difficult to traverse via a Markov chain. AIS addresses this by simulated annealing from a "high temperature" distribution to the low temperature one. This metaphorically corresponds to the real-world metallurgic technique of annealing, where a material is heated, followed by a controlled cooling process.

In our case, ratio-AIS "anneals" from one topic model towards the other instead of annealing from a high temperature to a low temperature distribution. The collapsed Gibbs sampler (CGS) mixes relatively well, as evidence by the fact that we are typically able to recover a good solution in a reasonable number of Gibbs iterations (Griffiths & Steyvers, 2004). We therefore might describe the Markov chains for topic models as being at "medium temperature." This is no longer strictly an annealing process in the physical sense as we are not necessarily varying the temperature of the system, but instead interpolating between two distributions which may be of similar temperature. Although we are using the AIS algorithm without modification, our scenario is very different to the typical AIS use case, as we are

able to approximately simulate from both the target and the source distributions via Gibbs sampling. Nevertheless, AIS is still very useful to us in this scenario as it gives us an estimate of the ratio of partition functions.

For the simulation to perform well in practice, we need our samples to be able to transition between the distributions of the two models, which makes the AIS proposal distribution accurate. For this to occur, we desire that *(1)* the Markov chains mix reasonably well, and *(2)* the two models are similar enough that the transition between them is not too arduous. As an argument for *(1)*, the CGS algorithm is relatively effective at training topic models from a random initialization. The inference problem in our case is easier than this because we already know the topics, and so we do not have to bootstrap (Rao-Blackwellized estimates of) both $\mathbf{\Phi}$ and $\mathbf{\Theta}$.

For *(2)*, the general efficacy of topic model training algorithms suggests that any two fully trained topic models are likely to be somewhat similar to each other, up to a permutation of the topics (which we address below). As for partially trained models, the topics typically have high entropy in the beginning stages of the algorithm, at least in the case of collapsed Gibbs sampling (cf. Figure 5.11). This means that the greater distance between models may potentially be mitigated by a higher temperature Markov chain, improving *(1)*.

Specifying the Ratio-AIS Path

To implement the ratio-AIS method, it remains to choose the annealing path, i.e. the sequence of intermediate distributions. We first consider a geometric average

$$f_j(\mathbf{z}^{(d)}) = f_0(\mathbf{z}^{(d)})^{\beta_j} f_n(\mathbf{z}^{(d)})^{1-\beta_j} \tag{5.29}$$

of the initial and final distributions, a strategy suggested by Neal (2001) with analogy to simulated annealing, where β_j can be viewed as an "inverse temperature." To choose a transition operator T_j invariant to f_j, we straightforwardly select the Gibbs sampler,

$$Pr(z_i^{(d)} = k|z^{-(d,i)}, \ldots) \propto \left((n_k^{(d)-(d,i)} + \alpha_k^{(1)})\mathbf{\Phi}^{(1)}_{w_i^{(d)},k}\right)^{\beta_j} \left((n_k^{(d)-(d,i)} + \alpha_k^{(2)})\mathbf{\Phi}^{(2)}_{w_i^{(d)},k}\right)^{1-\beta_j} . \quad (5.30)$$

We have importance weights

$$\begin{aligned}
\omega_i &= \prod_{j=0}^{n-1} \frac{Pr(w^{(d)}, \mathbf{z}_j^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(1)})^{\beta_j}}{Pr(w^{(d)}, \mathbf{z}_j^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(1)})^{\beta_{j+1}}} \frac{Pr(w^{(d)}, \mathbf{z}_j^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(2)})^{1-\beta_j}}{Pr(w^{(d)}, \mathbf{z}_j^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(2)})^{1-\beta_{j+1}}} \\
&= \prod_{j=0}^{n-1} \frac{Pr(w^{(d)}, \mathbf{z}_j^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(1)})^{\tau}}{Pr(w^{(d)}, \mathbf{z}_j^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(2)})^{\tau}} \\
\log \omega_i &= \frac{1}{n} \sum_{j=0}^{n-1} \log \frac{Pr(w^{(d)}, \mathbf{z}_j^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(1)})}{Pr(w^{(d)}, \mathbf{z}_j^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(2)})} ,
\end{aligned} \quad (5.31)$$

assuming $\beta_j - \beta_{j+1} = \tau = n^{-1}\ \forall j, 0 \leq j < n-1$. Elegantly, the log importance weights are the average of the log ratios of the probabilities of $w^{(d)}$ and $\mathbf{z}^{(d)}$ according to each model. Observe that the same $\mathbf{z}$ assignments are used for the numerator and denominator in each of the ratios in Equation 5.31, further reducing the variance of the estimate relative to the standard AIS strategy.

Although geometric averages are the standard choice for an annealing path, in many cases there exist annealing paths which perform much better. Grosse *et al.* (2013) introduced an alternative annealing path for exponential families which converges much more quickly, constructed by annealing averages of the moments of the sufficient statistics. The Dirichlet-multinomial distribution $Pr(\mathbf{z}^{(d)}|\alpha)$ is not an exponential family so their method does not directly apply. Nevertheless, we consider an annealing path inspired by their work, where intermediate distributions are constructed by taking convex combinations of the parameters:

$$f_j(\mathbf{z}^{(d)}) = Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}_j = \beta_j\mathbf{\Phi}^{(1)} + (1-\beta_j)\mathbf{\Phi}^{(2)}, \quad \alpha_j = \beta_j\alpha^{(1)} + (1-\beta_j)\alpha^{(2)}) \ . \tag{5.32}$$

The intermediate distributions are topic models, so we set T_j to be the corresponding Gibbs sampler of Equation 1.17. This T_j does not require power operations, providing substantial execution time savings over the geometric path and Equation 5.24. The importance weights are

$$\log \omega_i = \sum_{j=0}^{n-1} \Big(\log Pr(w^{(d)}, \mathbf{z}_j^{(d)}|\mathbf{\Phi}_j, \alpha_j) - \log Pr(w^{(d)}, \mathbf{z}_j^{(d)}|\mathbf{\Phi}_{j+1}, \alpha_{j+1}) \Big) \ . \tag{5.33}$$

To implement this method we need to draw initially from $f_n(\mathbf{z}^{(d)})$, which we accomplish via Gibbs sampling. These initial samples from $f_n(\mathbf{z}^{(d)})$ need not be independent for the procedure to work, although we may choose to run independent chains if the cost of burn-in is deemed to be less than time wasted due to running the annealing on correlated samples. Finally, AIS will be more likely to converge if the initial and target distributions are similar to each other. We therefore align the topics before running the algorithm, using the Hungarian algorithm to minimize the L1 distances between topics. This operation, which is $O(K^3)$, is not a computational bottleneck (relative to performing AIS) and needs only to be performed once per corpus. Pseudocode for ratio-AIS using the path from equation 5.32 is given in Algorithm 8.

Detecting Convergence Failures

AIS can produce poor estimates if the annealing fails to converge to a high-probability state in the target distribution within the given iterations. In general, this may be very difficult to detect. However, in our case we can interchange f_0 and f_n in our AIS strategy to compute

Algorithm 8 Ratio-AIS, using the convex path

for $i = 1 : S$ //importance samples

$\log\_\omega[i] := 0$

$\mathbf{\Phi}^{(next)} := \mathbf{\Phi}^{(2)}$

$\alpha^{(next)} := \alpha^{(2)}$

draw $\mathbf{z}^{(i)} \sim Pr(\mathbf{z}|\alpha^{(2)})$

for $j = n-1, n-2, \ldots, 0$ //temperatures

$\mathbf{\Phi}^{(curr)} := \mathbf{\Phi}^{(next)}$

$\alpha^{(curr)} := \alpha^{(next)}$

$\mathbf{\Phi}^{(next)} := \beta_j \mathbf{\Phi}^{(1)} + (1-\beta_j)\mathbf{\Phi}^{(2)}$

$\alpha^{(next)} := \beta_j \alpha^{(1)} + (1-\beta_j)\alpha^{(2)}$

for $a = 1 : r_j$ //r_{n-1} is large, for burn in

for $l = 1 : \text{length}(w^{(d)})$ //words

draw $z_l^{(i)}$, $Pr(z_l^{(i)} = k|.) \propto (n_k^{(i)} + \alpha_k^{(curr)}) \mathbf{\Phi}^{(curr)}_{w_l^{(d)},k}$

$\log\_\omega[i] := \log\_\omega[i] + \log Pr(w^{(d)}, \mathbf{z}^{(i)}|\mathbf{\Phi}^{(next)}, \alpha^{(next)})$
$- \log Pr(w^{(d)}, \mathbf{z}^{(i)}|\mathbf{\Phi}^{(curr)}, \alpha^{(curr)})$

return $\text{logSumExp}(\log\_\omega) - \log(S)$

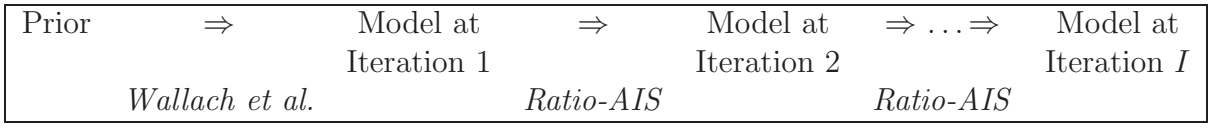


Figure 5.3: Iteration-AIS

the reciprocal of the desired ratio, and compare the reciprocal of this to our estimate. If these two values are wildly different, then we will know that the annealing has failed to converge. This means that we are able to detect convergence failures in many practical cases. In our experiments, we were easily able to catch convergence failures by observing a systematic bias across documents in the results of the different annealing directions (see Section 5.3). Interestingly, such a run in the reverse direction is exactly the target distribution which the AIS importance sampler is attempting to draw from.

5.2.2 Efficiently Evaluating Topic Model Learning Algorithms with Iteration-AIS

When evaluating algorithms for learning topic models (or monitoring their convergence), we would ideally like to compute and plot held-out log-likelihood scores per learning iteration (or unit of computation time) for each algorithm under consideration. This is extremely expensive, requiring $|T| \times I \times M$ Monte Carlo approximations of already intractable high-dimensional integrals, where T is the held-out test set, I is the number of iterations of the learning algorithms to evaluate at, and M is the number of competing learning methods.

To address this computational challenge, a key insight is that we are free to set the annealing path to any convenient sequence of intermediate distributions that we choose. Fortunately, for many learning algorithms such as the collapsed Gibbs sampler, the topics at successive iterations are similar to each other, and the topics typically vary smoothly from "high temperature" high entropy distributions at early iterations to more complicated later distri-

butions. This suggests using a single AIS path to perform the entire evaluation across all of the iterations, with the topic models at each iteration (or a subset of them) as intermediate distributions. Ratio-AIS gives us a smooth path between each of the successive topic models, ensuring that the "gap" between successive distributions is not too large.

The annealing path, then, begins by drawing from the prior $Pr(\mathbf{z}^{(d)}|\alpha^{(1)})$, and then annealing from the prior to the first topic model $Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(1)})$ as in Wallach *et al.* (2009b). The path continues by using ratio-AIS to anneal between successive topic models $Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}^{(k)}, \alpha^{(k)})$. At topic model k, the average $S^{-1}\sum_{i=1}^{S}\omega_{i,k}$ of the importance weights computed up to that point $\omega_{i,k}$ converges to the ratio of normalizing constants,

$$\frac{\sum_{\mathbf{z}^{(d)}} Pr(w^{(d)}, \mathbf{z}^{(d)}|\mathbf{\Phi}^{(k)}, \alpha^{(k)})}{\sum_{\mathbf{z}^{(d)}} Pr(\mathbf{z}^{(d)}|\alpha^{(1)})} = Pr(w^{(d)}|\mathbf{\Phi}^{(k)}, \alpha^{(k)}). \tag{5.34}$$

With n temperatures per learning iteration k, importance weights are given recursively as

$$\log \omega_{i,k} = \sum_{k'=1}^{k}\sum_{j=0}^{n-1} \log \frac{f_{k',j}(\mathbf{z}^{(d)}_{k',j})}{f_{k',j+1}(\mathbf{z}^{(d)}_{k',j})} \tag{5.35}$$

$$= \log \omega_{i,k-1} + \sum_{j=0}^{n-1} \log \frac{f_{k,j}(\mathbf{z}^{(d)}_{k,j})}{f_{k,j+1}(\mathbf{z}^{(d)}_{k,j})} \ . \tag{5.36}$$

This method, which we refer to as *iteration-AIS*, exploits all of the computation for selecting $\mathbf{z}$ assignments and importance weights from the likelihood estimates at previous learning iterations. By concatenating the annealing paths, the estimate at each iteration k gains a successively longer annealing run as k increases, with no extra computation.

Neal (2001) argues extensively that longer annealing runs will reduce the variance of the importance weights, which consequently improves the sampling efficiency of the estimates. Intuitively, a greater number of temperatures means more MCMC iterations, and so a better chance to converge to a high probability region. This also means that both the "cooling" and "heating" processes will be closer to reaching equilibrium, and so more similar to (reversed

versions of) each other, making the cooling process a better proposal distribution for the reversed heating process.

Increasing the number of temperatures also allows the algorithm to explore more of the space, thus taking more of the distribution into account when computing it estimates. In the case of iteration-AIS, we can also think of the procedure as giving AIS a *warm start* at the previous model, instead of starting blind from the prior. An initial sample from the previous model, which is generally similar to the current model, is likely have much higher probability under the current model than an initial sample from the prior. We show a diagram of the method in Figure 5.3.

5.2.3 Application to Document Completion

In another application of ratio-AIS, suppose we would instead like to compare the performance of two topic models on the document completion task described in Section 5.1.5, where we observe some portion of a document $w^{(d,a)}$ and the goal is to predict the remainder of the document $w^{(d,b)}$. The "estimated θ" strategy for document completion uses a point estimate trained on $w^{(d,a)}$ to predict the remainder of the document, but this neglects to account for the uncertainty in the posterior for $\theta^{(d)}$. Instead, we can use ratio-AIS to estimate the document completion likelihood ratio, more fully taking into account this uncertainty by approximately marginalizing over the hidden variables. The relative performance of our two models at this task is

$$\begin{aligned}\frac{Pr(w^{(d,b)}|w^{(d,a)}, \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})}{Pr(w^{(d,b)}|w^{(d,a)}, \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})} &= \frac{Pr(w^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(d,1)})}{Pr(w^{(d,a)}|\mathbf{\Phi}^{(1)}, \alpha^{(d,1)})} \times \frac{Pr(w^{(d,a)}|\mathbf{\Phi}^{(2)}, \alpha^{(d,2)})}{Pr(w^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(d,2)})} \\ &= \frac{Pr(w^{(d)}|\mathbf{\Phi}^{(1)}, \alpha^{(d,1)})}{Pr(w^{(d)}|\mathbf{\Phi}^{(2)}, \alpha^{(d,2)})} \times \frac{Pr(w^{(d,a)}|\mathbf{\Phi}^{(2)}, \alpha^{(d,2)})}{Pr(w^{(d,a)}|\mathbf{\Phi}^{(1)}, \alpha^{(d,1)})} \ . \end{aligned} \tag{5.37}$$

The two terms in the product can each be estimated via ratio-AIS, and then multiplied together to get an estimate of the ratio of the document completion likelihoods of the two models. Note that the first term, corresponding to the usual value computed by ratio-AIS (Equation 5.27), is independent of which portion $w^{(d,a)}$ of the document is observed, which means that it need only be computed once when varying the observed portion. The second term can also be computed using ratio-AIS, annealing from model 1 to model 2 and only executing the sampler on the observed portion $w^{(d,a)}$.

To provide an alternative perspective, we can also derive this procedure based on a single annealing path which estimates the document completion ratio directly. In this case, we let

$$f_0(\mathbf{z}^{(d)}) = Pr(w^{(d,b)}, \mathbf{z}^{(d)} | w^{(d,a)}, \mathbf{\Phi}^{(1)}, \alpha^{(d,1)}) \tag{5.38}$$

$$f_n(\mathbf{z}^{(d)}) = Pr(w^{(d,b)}, \mathbf{z}^{(d)} | w^{(d,a)}, \mathbf{\Phi}^{(2)}, \alpha^{(d,2)}) \ , \tag{5.39}$$

where the state space $\mathbf{z}^{(d)}$ consists of the $\mathbf{z}$'s for the entire document (not just those associated with $w^{(d,b)}$, for reasons which will become clear below). The average of the importance weights will estimate the desired quantity as the ratio of partition functions

$$\frac{1}{S}\sum_{i=1}^{S}\omega_i \approx \frac{\sum_{\mathbf{z}^{(d)}} Pr(w^{(d,b)}, \mathbf{z}^{(d)} | w^{(d,a)}, \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})}{\sum_{\mathbf{z}^{(d)}} Pr(w^{(d,b)}, \mathbf{z}^{(d)} | w^{(d,a)}, \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})} = \frac{Pr(w^{(d,b)} | w^{(d,a)}, \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})}{Pr(w^{(d,b)} | w^{(d,a)}, \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})} \ . \tag{5.40}$$

To compute the importance weights, let us first rewrite

$$Pr(w^{(d,b)}, \mathbf{z}^{(d)} | w^{(d,a)}, \mathbf{\Phi}^{(1)}, \alpha^{(d,1)}) = \frac{Pr(w^{(d)}, \mathbf{z}^{(d)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})}{Pr(w^{(d,a)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})} \tag{5.41}$$

$$Pr(w^{(d,b)}, \mathbf{z}^{(d)} | w^{(d,a)}, \mathbf{\Phi}^{(2)}, \alpha^{(d,2)}) = \frac{Pr(w^{(d)}, \mathbf{z}^{(d)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})}{Pr(w^{(d,a)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})} \ . \tag{5.42}$$

We assume a geometric path with uniform step sizes $\beta_j - \beta_{j+1} = \tau = n^{-1}$. Then the importance weights are

$$
\begin{aligned}
\omega_i &= \prod_{j=0}^{n-1} \frac{f_j(\mathbf{z}_j)}{f_{j+1}(\mathbf{z}_j)} \\
&= \prod_{j=0}^{n-1} \Big(\frac{Pr(w^{(d)}, \mathbf{z}_j^{(d)} | \mathbf{\Phi}^{(1)}, \alpha^{(1)})^{\beta_j}}{Pr(w^{(d)}, \mathbf{z}_j^{(d)} | \mathbf{\Phi}^{(1)}, \alpha^{(1)})^{\beta_{j+1}}} \frac{Pr(w^{(d)}, \mathbf{z}_j^{(d)} | \mathbf{\Phi}^{(2)}, \alpha^{(2)})^{1-\beta_j}}{Pr(w^{(d)}, \mathbf{z}_j^{(d)} | \mathbf{\Phi}^{(2)}, \alpha^{(2)})^{1-\beta_{j+1}}} \\
&\qquad \times \big(\frac{Pr(w^{(d,a)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})^{\beta_j}}{Pr(w^{(d,a)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})^{\beta_{j+1}}} \frac{Pr(w^{(d,a)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})^{1-\beta_j}}{Pr(w^{(d,a)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})^{1-\beta_{j+1}}} \big)^{-1} \Big) \\
&= \prod_{j=0}^{n-1} \Big(\frac{Pr(w^{(d)}, \mathbf{z}_j^{(d)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})^{\tau}}{Pr(w^{(d)}, \mathbf{z}_j^{(d)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})^{\tau}} \Big) \times \frac{Pr(w^{(d,a)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})^{n\tau}}{Pr(w^{(d,a)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})^{n\tau}} \\
&= \prod_{j=0}^{n-1} \Big(\frac{Pr(w^{(d)}, \mathbf{z}_j^{(d)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})^{\tau}}{Pr(w^{(d)}, \mathbf{z}_j^{(d)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})^{\tau}} \Big) \times \frac{Pr(w^{(d,a)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})}{Pr(w^{(d,a)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})} \ . \qquad (5.43)
\end{aligned}
$$

These importance weights consist of two terms: *(1)* the importance weights for ratio-AIS samples of the ratio of likelihoods of the entire document (Equation 5.31), and *(2)* the reciprocal ratio of likelihoods for just the observed portion of the document. We can estimate *(2)* using ratio-AIS on the observed portion $w^{(d,a)}$ in order to estimate Equation 5.43, and then average the resulting weights to estimate the desired quantity in Equation 5.40. By distributivity, this is equivalent to the procedure implied by Equation 5.37.

$$
\begin{aligned}
\frac{1}{S} \sum_{i=1}^{S} \omega_i &= \Big(\frac{1}{S} \sum_{i=1}^{S} \prod_{j=0}^{n-1} \frac{Pr(w^{(d)}, \mathbf{z}_j^{(d)(i)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})^{\tau}}{Pr(w^{(d)}, \mathbf{z}_j^{(d)(i)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})^{\tau}} \Big) \quad \times \quad \frac{Pr(w^{(d,a)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})}{Pr(w^{(d,a)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})} \qquad (5.44) \\
&\approx \frac{Pr(w^{(d)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})}{Pr(w^{(d)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})} \quad \times \quad \frac{Pr(w^{(d,a)} | \mathbf{\Phi}^{(2)}, \alpha^{(d,2)})}{Pr(w^{(d,a)} | \mathbf{\Phi}^{(1)}, \alpha^{(d,1)})} \qquad (5.45)
\end{aligned}
$$

5.3 Experiments

We explored the performance of the proposed techniques using a corpora of scientific articles from the Association of Computational Lingusitics (ACL) conference[8] (Radev *et al.* , 2013),

[8]Available at `http://clair.eecs.umich.edu/aan/index.php` .

and another from the Neural Information Processing Systems (NIPS) conference.[9] The ACL dataset consists of the 3286 articles from the years 1987 to 2011, while the NIPS corpus contains the 1740 articles published between 1987 and 1999. In each experiment, topic models with 50 topics were fit to each corpus by performing 1000 iterations of collapsed Gibbs sampling using the MALLET toolkit (McCallum, 2002). Roughly 10% of the documents in each corpus were withheld for testing (130 NIPS articles, and 300 ACL articles). Although cross-validation would have been preferable to a single hold-out set, the computational expense of the experiments prevented this. For example, across all algorithms and learning iterations, Figures 5.7 – 5.10 required a total of 6.6 million Gibbs iterations for *each one of the test articles*.

When using AIS we must select the number of temperatures n, the number of importance samples S, and the temperature schedule $\beta_0, \beta_1, \ldots, \beta_n$. The variability of an AIS estimator can be reduced by increasing S (due to the law of large numbers) or by increasing n (which reduces the variance of the ω_i). In the experiments, we focused on the case where $S = 1$, as in Wallach *et al.* (2009b). We found in preliminary experiments that $S = 1$ gave essentially exactly the same answer as $S = 100$ importance samples for ratio-AIS with 10,000 temperatures. For simplicity, we used a uniform spacing of the temperatures β_j.

We also compared to the left-to-right (LR) particle filtering algorithm of Wallach *et al.* (2009b), using the implementation provided in MALLET. The left-to-right method requires $N_d(N_d + 1)/2$ word-level Gibbs updates per particle for a document of length N_d. The execution of $p = 2 * n/(N_d + 1)$ particles corresponds to the same number of Gibbs updates as AIS with n temperatures and $S = 1$. We select the number of LR particles by rounding p to the nearest integer greater than zero.

[9]The NIPS dataset, due to Gregor Heinrich, is available at `http://www.arbylon.net/resources.html`.

Ratio-AIS was designed for reliable per-document comparisons. To explore this, we ran each algorithm twice on each document, and reported results comparing the two runs across documents. To remove the effect of document length in the results, instead of reporting the differences in log-likelihood scores for each model we consider instead perplexity scores

$$\text{perplexity}(w^{(d)}; \mathbf{\Phi}, \alpha) = \exp\Big(- \frac{\log Pr(w^{(d)}|\mathbf{\Phi}, \alpha)}{N_d} \Big) \ . \tag{5.46}$$

The ratio of the perplexity of model 1 over the perplexity of model 2 for a document is readily computed from the output of ratio-AIS as

$$\frac{\exp(-\frac{L_1}{N_d})}{\exp(-\frac{L_2}{N_d})} = \exp\Big(\frac{L_2 - L_1}{N_d} \Big) \ , \tag{5.47}$$

where L_j is the log-likelihood for model j. We considered two evaluation scenarios for ratio-AIS: comparing learned topics to perturbed versions of the same topics (Section 5.3.1), and comparing topic models learned with symmetric and asymmetric Dirichlet priors (Section 5.3.2). Finally, we evaluated iteration-AIS for the estimation of per-iteration likelihood values (Section 5.3.3).

5.3.1 Learned Topics versus Perturbed Topics

As the likelihoods we are trying to estimate are intractable, we do not in general have access to ground truth. However, after learning topics $\mathbf{\Phi}$ on a dataset and then creating a noisy copy of them $\mathbf{\Phi}'$, we have good reason to believe that the original topics $\mathbf{\Phi}$ are better than the copy. This style of experiment was previously performed by Wallach *et al.* (2009b). We took the word-topic assignments learned by MALLET, and created $\mathbf{\Phi}'$ by re-assigning 5% of them to new word-topic assignments uniformly at random.[10]

[10]MALLET's left-to-right implementation takes as input a count matrix, so the perturbed topics must be representable as counts.

% Correct	Left to Right	Standard AIS	Ratio-AIS Geometric	Ratio-AIS Geometric (reverse)	Ratio-AIS Convex	Ratio-AIS Convex (reverse)
NIPS (cheap)	63.8	48.8	83.8	**89.2**	84.6	87.7
NIPS (expensive)	84.6	62.3	86.9	**87.7**	**87.7**	**87.7**
ACL (cheap)	80.2	50.8	88.3	92.0	88.3	**92.3**
ACL (expensive)	**90.7**	75.2	90.3	90.3	90.3	90.3

Table 5.1: Percentage of documents where the learned topics $\mathbf{\Phi}$ were estimated to have higher likelihood than perturbed versions of them $\mathbf{\Phi}'$.

Ratios of the perplexities for the two models were computed with both cheap (100 temperatures) and expensive runs (10,000 temperatures). Overall results are given in Tables 5.1 and 5.2, and per-document results are plotted in Figures 5.4 and 5.5. Table 5.1 shows that the two ratio-AIS paths were the most accurate methods by a significant margin in three out of four of the scenarios, and with a similar result to left-to-right in the fourth (the ACL dataset, in the expensive regime).

In the cheap regime, the ratio-AIS points are slightly off-diagonal in Figures 5.4 and 5.5, with one annealing direction giving systematically lower results, representing a detected bias due to convergence failure in at least one annealing direction. Nevertheless, these results had much lower empirical variance (see also Table 5.2, top), and the bias disappeared in the expensive regime. Surprisingly, the standard AIS method performed extremely poorly, with most data points falling outside of the bounds of the figures, which are tight around the results of the other methods (except for Figure 5.5 (top), which is tight around the ratio-AIS results, and also has points for the left-to-right method falling out of the figure). Using many importance samples would very likely mitigate this, at greater computational cost.

Table 5.2 shows the estimated variance of the per-document perplexity ratios (top), and the overall perplexity ratio across the corpus (bottom). The variance of the perplexity ratio estimates was orders of magnitude smaller than those from the left to right and standard AIS approaches. In this case, the geometric path had lower variance than the convex path.

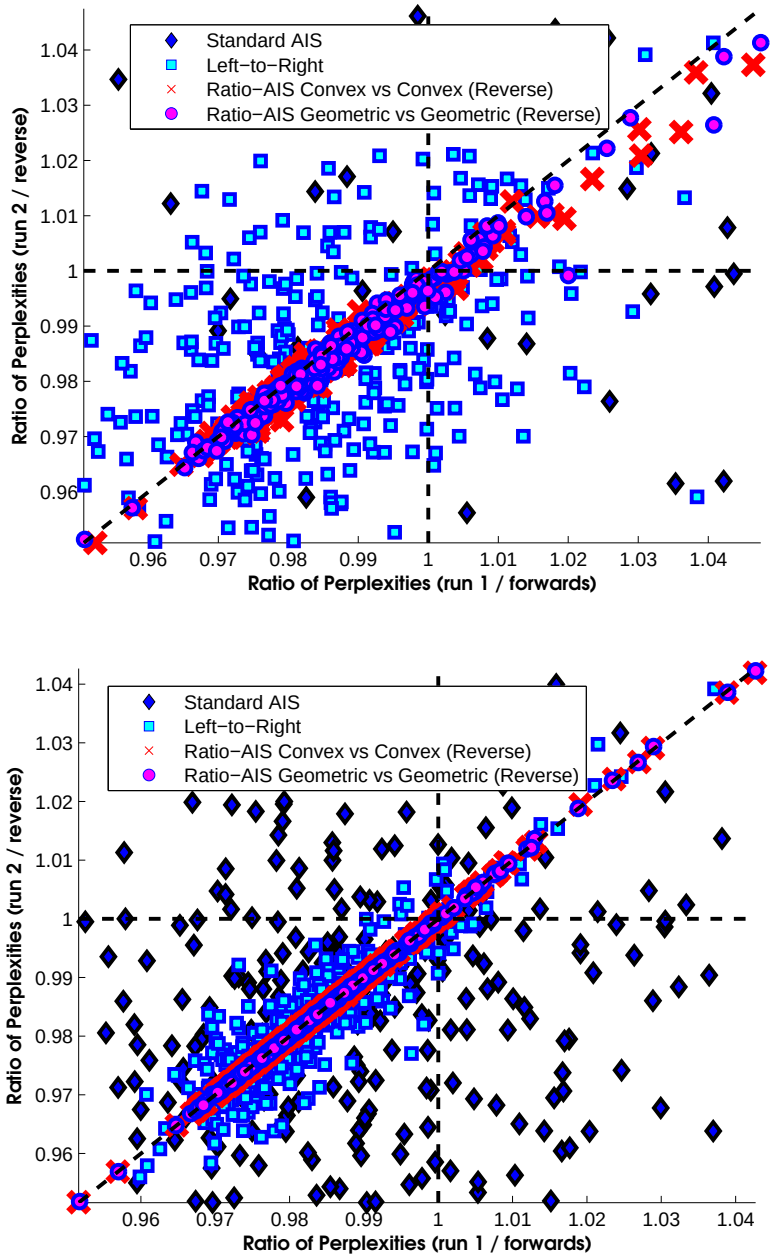


Figure 5.4: Comparing learned topics with perturbed versions of them, on the ACL dataset. In the figures, every point corresponds to a document. Each axis corresponds to estimated $\frac{\text{perp}(\Phi)}{\text{perp}(\Phi')}$ for a repeat of the experiment, with the ratio-AIS repeats being performed in different annealing directions. Points in the lower left quadrant are those which (likely correctly) predict the unperturbed topics as the winner in both trials. Points near the diagonal have consistent results across the two trials. **Top**: 100 temperatures. **Bottom**: 10,000 temperatures. Missing Standard AIS results are outside of the bounds of the figures.

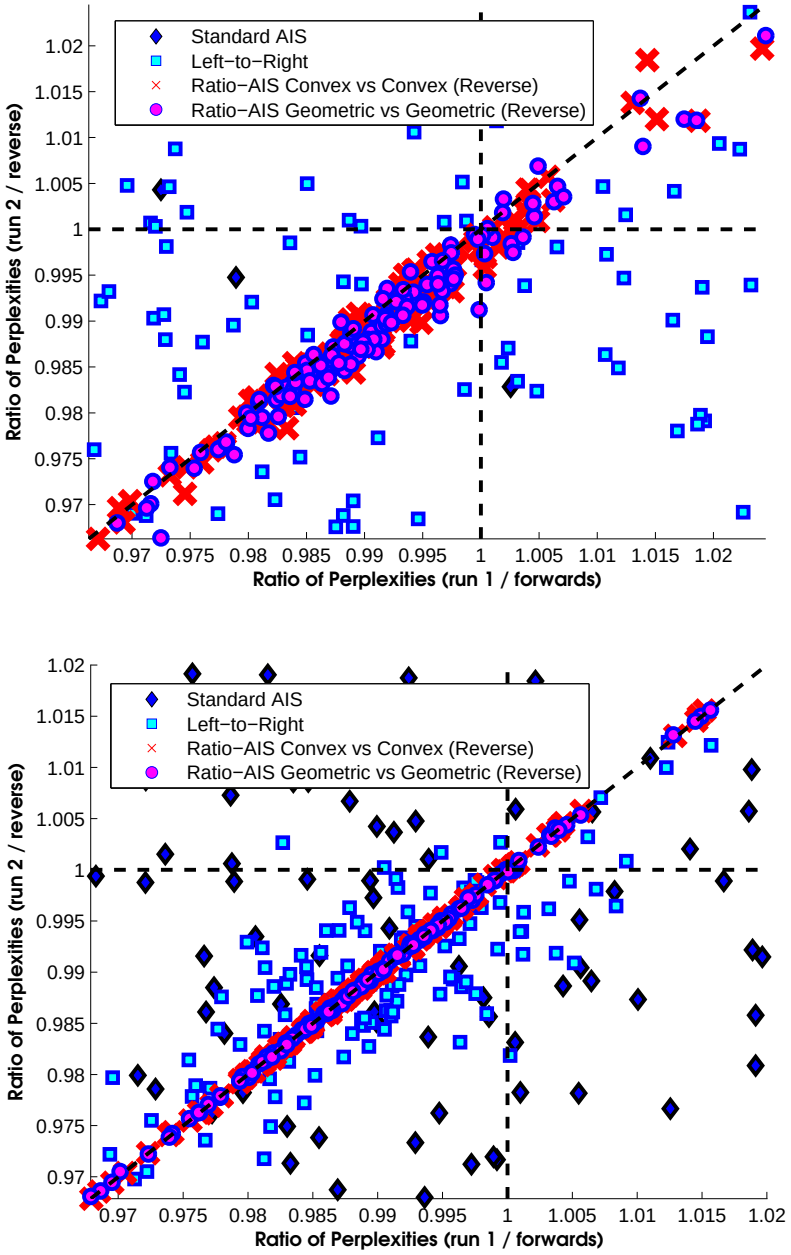


Figure 5.5: Comparing learned topics with perturbed versions of them, on the NIPS dataset. **Top**: 100 temperatures. **Bottom**: 10,000 temperatures. Missing Standard AIS and left-to-right results are outside of the bounds of the figures.

Giving a sense of the role of the number of temperatures in the variance of the estimates, Figure 5.6 plots log-likelihood estimates for a single NIPS article against n, averaged over 100 samples. The error bars in the plot show the standard deviations of the importance samples. It was found that the empirical variance across the ratio-AIS samples was extremely small relative to the standard AIS approach, unless a large number of temperatures was used, in which case the variance of standard AIS eventually decreased. The standard AIS approach also tended to underestimate the difference between the models relative to the other methods, and to the results of very long annealing runs.

However, the number of temperatures did affect the likelihood estimates for both AIS and ratio-AIS. This makes sense, as a greater number of temperatures allows the methods to more fully explore the sample space. With 20,000 temperatures or more, both of the ratio-AIS annealing directions and the standard AIS method converged on the same solution. The figure suggests that a single importance sample can be accurate for both AIS and ratio-AIS as long as a sufficient number of temperatures is used. This is consistent with Neal (2001)'s arguments suggesting that the variance of the importance weights decreases as the number of temperatures grows.

Overall perplexity results, computed across the entire corpus, are shown in Table 5.2 (bottom). Since $\mathbf{\Phi}'$ is a noisy copy of the learned topics $\mathbf{\Phi}$, we expect that $\frac{\text{perp}(\mathbf{\Phi})}{\text{perp}(\mathbf{\Phi}')}$ should be less than one, and lower values correspond to a bigger detected difference. Ratio-AIS estimated lower perplexity ratios than the baselines on NIPS. The left-to-right algorithm was competitive on ACL, and in fact reported a slightly lower perplexity ratio in the expensive regime (10,000 temperatures). However, in this regime ratio-AIS was remarkably consistent across annealing directions and across annealing paths, reporting essentially identical results in all cases, and on both datasets. The consistency of these results, along with the similarity of the predictions to the AIS and left to right methods (e.g. on ACL, the ratio-AIS result was between the ratios predicted by the two baseline methods) provides evidence that these

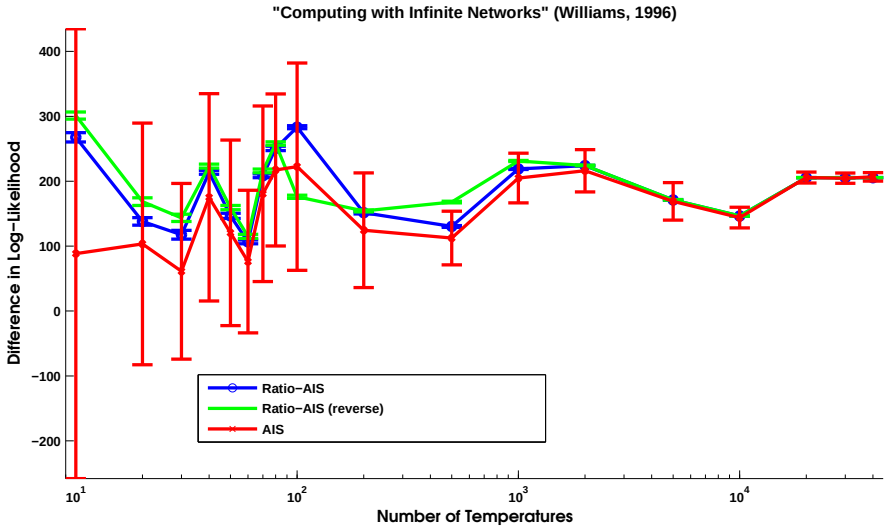


Figure 5.6: Estimated differences in log-likehood for the perturbed topics task, varying the number of temperatures (geometric annealing path). Error bars denote standard deviations across 100 importance samples.

predictions are accurate. It should be noted that the task of comparing two very similar topic models is difficult for standard methods, but is relatively easy for ratio-AIS due to the distance to anneal between the distributions being smaller.

5.3.2 Symmetric versus Asymmetric Dirichlet Priors

Learning asymmetric α hyper-parameters can improve the predictive performance of topic models (e.g., Wallach *et al.* (2009a)). To explore this, on each corpus we learned a topic model with asymmetric α, and a model where α was fixed to be flat but its concentration parameter was learned. The AIS and LR algorithms were used to compare the resulting models, using runs with 1000 temperatures and 10,000 temperatures.

It was found that in the "cheap" 1000 temperature regime, the ratio-AIS estimates were the most closely correlated with left-to-right estimates in the expensive regime, the best available proxy for ground truth (Table 5.3, bottom).[11] In all cases the ratio-AIS paths had one to two

[11]The standard AIS estimate of the perplexity ratios had too high a variance to be used (see Table 5.3).

Variance of Perplexity Ratio	Left to Right	Standard AIS	Ratio-AIS Geometric	Ratio-AIS Geometric (reverse)	Ratio-AIS Convex	Ratio-AIS Convex (reverse)
NIPS (cheap)	3.9×10^{-4}	1.2×10^{-2}	1.8×10^{-6}	$\mathbf{1.1\times10^{-6}}$	1.3×10^{-6}	1.5×10^{-6}
NIPS (expensive)	1.9×10^{-5}	5.1×10^{-4}	$\mathbf{9.8\times10^{-9}}$	1.3×10^{-8}	1.6×10^{-8}	1.3×10^{-8}
ACL (cheap)	2.2×10^{-4}	1.3×10^{-2}	2.1×10^{-6}	$\mathbf{8.2\times10^{-7}}$	1.4×10^{-6}	1.2×10^{-6}
ACL (expensive)	1.6×10^{-5}	5.2×10^{-4}	$\mathbf{1.0\times10^{-8}}$	1.1×10^{-8}	1.2×10^{-8}	1.2×10^{-8}
Corpus-Level Perplexity Ratio	Left to Right	Standard AIS	Ratio-AIS Geometric	Ratio-AIS Geometric (reverse)	Ratio-AIS Convex	Ratio-AIS Convex (reverse)
NIPS (cheap)	0.991	0.996	0.991	**0.989**	0.991	**0.989**
NIPS (expensive)	0.991	0.993	**0.9897**	**0.9897**	**0.9897**	**0.9897**
ACL (cheap)	**0.984**	0.995	0.985	**0.984**	0.986	**0.984**
ACL (expensive)	**0.983**	0.985	0.9845	0.9845	0.9845	0.9845

Table 5.2: Comparing learned topics $\mathbf{\Phi}$ with perturbed versions of them $\mathbf{\Phi}'$. Average empirical variance (evaluated across two runs per document) of the per-document perplexity ratio (**top**), and the overall perplexity ratio for the entire corpus (**bottom**). The "cheap" runs performed 100 temperatures (or equivalent) and the "expensive" performed 10,000 temperatures (or equivalent).

orders of magnitude lower empirical variance in the estimates of per-document perplexity ratios than the previous methods, with the convex path having the least variance (Table 5.3, top). Ratio-AIS therefore achieves the original goal of greatly reducing the variance of per-document comparisons of topic models. This is particularly important if we want to perform detailed analysis at a per-document level, such as exploring the effect of covariates on topic model performance. In such a scenario, the previous methods have unacceptably high variance for a reasonable level of computation (see also Figure 5.1), while the ratio-AIS estimates of relative performance have very small empirical variance, even with an estimate produced using just one importance sample.

Unfortunately, this reduction comes at a price of potentially increased bias in the estimated perplexity ratio when given insufficient computation. Topic models which learn an asymmetric α tend to perform better than those with a symmetric α (Wallach *et al.* , 2009a), and the previous methods detected a larger advantage for the asymmetric approach (Table 5.3, middle). The direction of the ratio-AIS annealing path also made a difference to the out-

Variance of Perplexity Ratio	Left to Right	Standard AIS	Ratio-AIS Geometric	Ratio-AIS Geometric (reverse)	Ratio-AIS Convex	Ratio-AIS Convex (reverse)
NIPS (cheap)	2.6×10^{-4}	2.6×10^{-3}	2.0×10^{-5}	1.5×10^{-5}	$\mathbf{8.2\times10^{-6}}$	9.8×10^{-6}
NIPS (expensive)	1.7×10^{-5}	6.0×10^{-4}	1.4×10^{-6}	1.2×10^{-6}	6.9×10^{-7}	$\mathbf{5.8\times10^{-7}}$
ACL (cheap)	1.7×10^{-4}	3.6×10^{-3}	1.6×10^{-5}	1.3×10^{-5}	7.7×10^{-6}	$\mathbf{6.6\times10^{-6}}$
ACL (expensive)	1.4×10^{-5}	5.6×10^{-4}	1.1×10^{-6}	9.4×10^{-7}	7.4×10^{-7}	$\mathbf{5.1\times10^{-7}}$
Corpus-Level Perplexity Ratio	Left to Right	Standard AIS	Ratio-AIS Geometric	Ratio-AIS Geometric (reverse)	Ratio-AIS Convex	Ratio-AIS Convex (reverse)
NIPS (cheap)	0.984	**0.975**	1.01	0.992	1.01	0.994
NIPS (expensive)	**0.989**	0.990	1.00	0.999	1.00	0.998
ACL (cheap)	0.984	**0.980**	1.00	0.985	1.00	0.988
ACL (expensive)	**0.987**	0.989	0.994	0.992	0.996	0.992
Correlation with Long LR Run	Left to Right	Standard AIS	Ratio-AIS Geometric	Ratio-AIS Geometric (reverse)	Ratio-AIS Convex	Ratio-AIS Convex (reverse)
NIPS (cheap)	0.947	0.619	0.973	0.975	0.976	**0.981**
NIPS (expensive)	**0.993**	0.852	0.981	0.982	0.981	0.982
ACL (cheap)	0.965	0.578	0.984	0.983	**0.987**	0.986
ACL (expensive)	**0.995**	0.892	0.989	0.989	0.990	0.989

Table 5.3: Comparing asymmetric α and symmetric α topic models. Correlation coefficient with the perplexity ratio estimates from a run of left-to-right in the expensive regime (**top**), average empirical variance (evaluated across two runs per document) of the per-document perplexity ratio (**middle**), and the overall perplexity ratio for the entire corpus (**bottom**).

come. In particular, the forward direction of annealing did not detect an overall advantage to the asymmetric hyper-parameter model. On the other hand, the difference per direction allowed us to detect a convergence failure, which is difficult to do in general. Also note that for the perturbed topics task in Section 5.3.1, the overall perplexity ratios were very consistent between annealing directions, and showed a clearer difference between models than the baseline algorithms did.

5.3.3 Evaluating Topic Models per Iteration

The iteration-AIS annealing path evaluates the performance of topic model learning algorithms on a per-iteration basis. We explored its performance using the convex path with 1000 and 10,000 temperatures per learned model, annealing between the models at every 10th learning iteration. At the first learning iteration $\mathbf{\Phi}^{(1)}$, the algorithms were given an extra 1000 temperatures to compensate for the cold-start from the prior.

Results on ACL and NIPS are shown in Figures 5.7 – 5.10. It was found that iteration-AIS estimated higher log-likelihoods than left-to-right and standard AIS in both temperature regimes and data sets (Figures 5.7 and 5.8). The main failure mode of these algorithms is to underestimate the likelihood by failing to find high probability regions, so higher values are likely to be better (Wallach *et al.* , 2009b). Consistent with this observation, the iteration-AIS likelihood curve at 1000 temperatures coincided with the likelihood curves of the baselines when they were given ten times more computation. The proposed method also exhibited much lower variance in the likelihood estimates (Figures 5.9 and 5.10, computed based on two evaluations of the likelihood per document, and averaged across documents). This is expected, as the effective number of annealing temperatures is higher, which is known to reduce the variance of the importance weights (Neal, 2001).

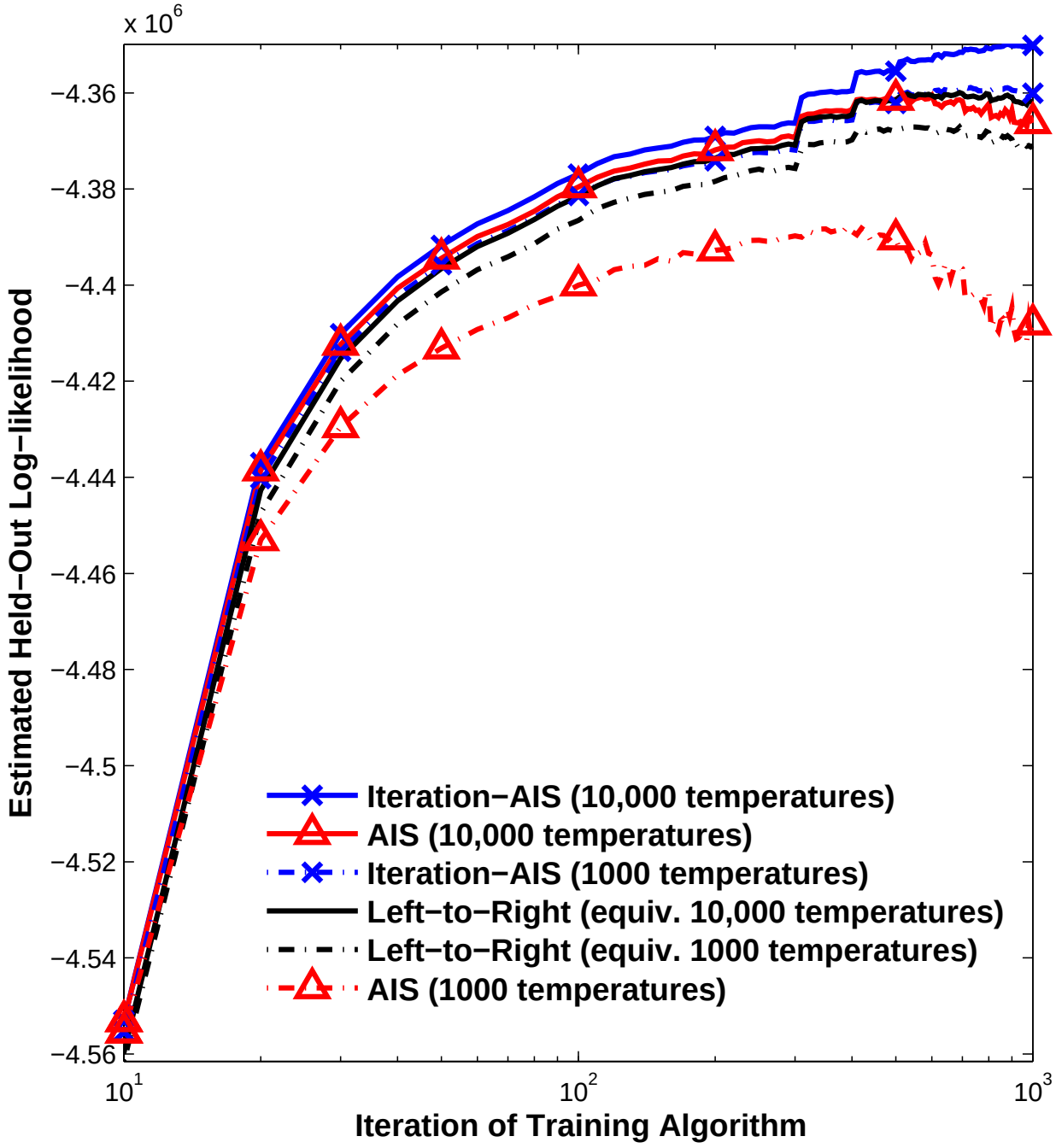


Figure 5.7: Likelihood vs iteration for iteration-AIS on the ACL corpus. Jumps in log-likelihood are due to hyper-parameter optimization.

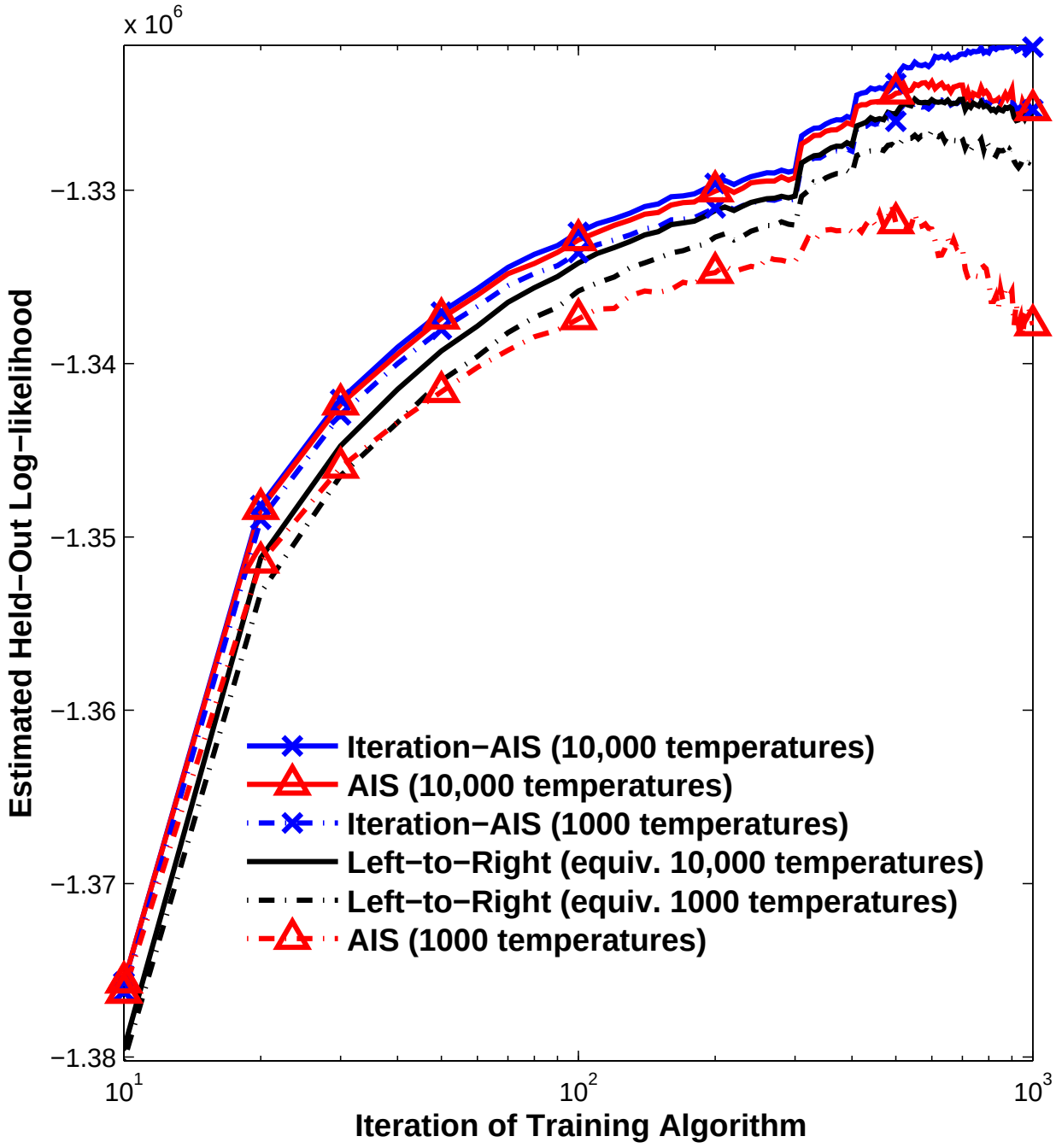


Figure 5.8: Likelihood vs iteration for iteration-AIS on the NIPS corpus. Jumps in log-likelihood are due to hyper-parameter optimization.

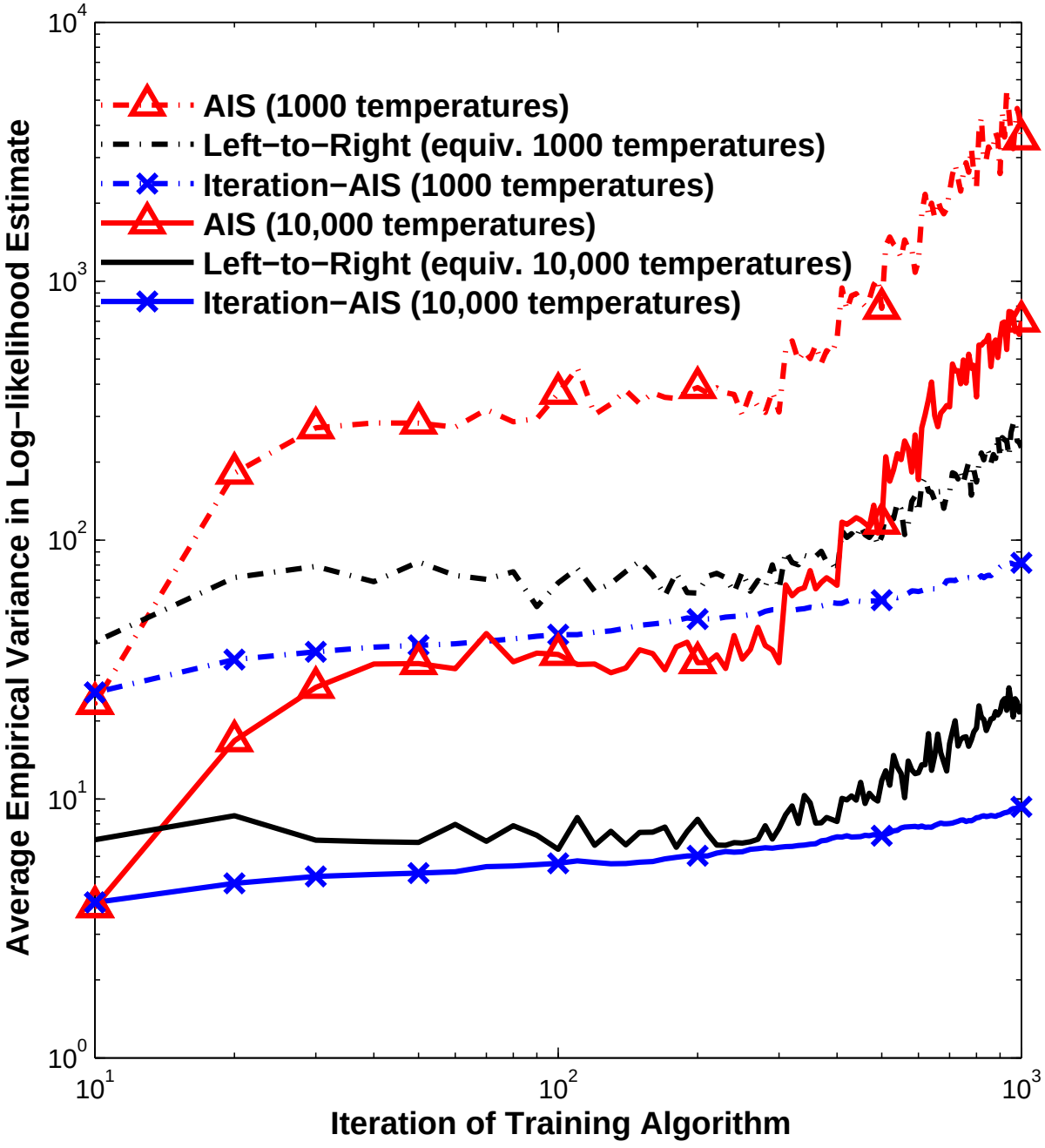


Figure 5.9: Empirical variance vs iteration for iteration-AIS on the ACL corpus.

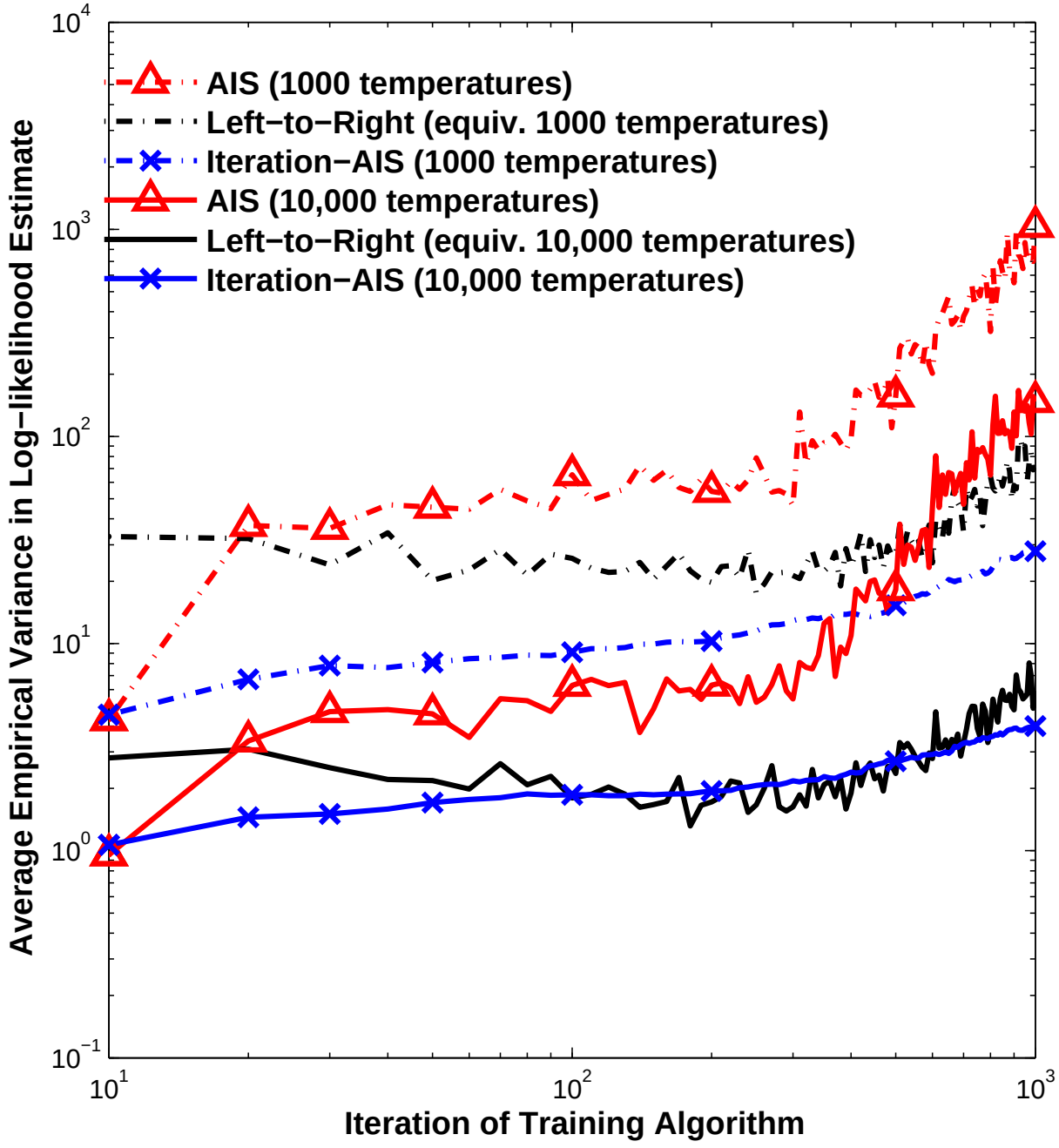


Figure 5.10: Empirical variance vs iteration for iteration-AIS on the NIPS corpus.

Explaining the Behavior of the Baselines

The baselines reported decreasing held-out likelihood in later iterations of the learning algorithm, while iteration-AIS did not. Such a decrease could be due to over-fitting, but is more likely to be caused by convergence failures due to the topics becoming more complex. As evidence for this, the dip in likelihood was smaller with increased computation, and all methods exhibited higher variance in the likelihood estimates for later learning iterations (Figures 5.9 and 5.10).

Figure 5.11 demonstrates this further by showing the entropy and the prior probability of the topics. In early iterations, the topics have low entropy, and so annealing is easy – the Markov chains to sample from them are "high temperature." At iteration 300, after optimizing the hyper-parameters, a phase transition occurs as a mode is found which is far from the prior (Figure 5.11, bottom). This makes sampling from it difficult, and the performance of the baseline likelihood estimation algorithms, which are initialized based on the prior, correspondingly degrades.

5.4 Connections to Particle-Filtered MCMC-MLE

In this section we discuss the relationship between iteration-AIS and another algorithm in the literature due to Asuncion *et al.* (2010). We have seen that iteration-AIS consecutively moves a set of samples through a sequence of distributions, with the sequence containing the models produced at different iterations of a learning algorithm. This is reminiscent of the particle-filtered (PF) MCMC-MLE algorithm of Asuncion *et al.* (2010), which does this *as part of the learning algorithm itself*. Similarly to iteration-AIS, Asuncion *et al.* draw a set of samples based on the parameters at each training iteration by applying MCMC updates to the samples from the previous iteration. Iteration-AIS uses these samples to approximate

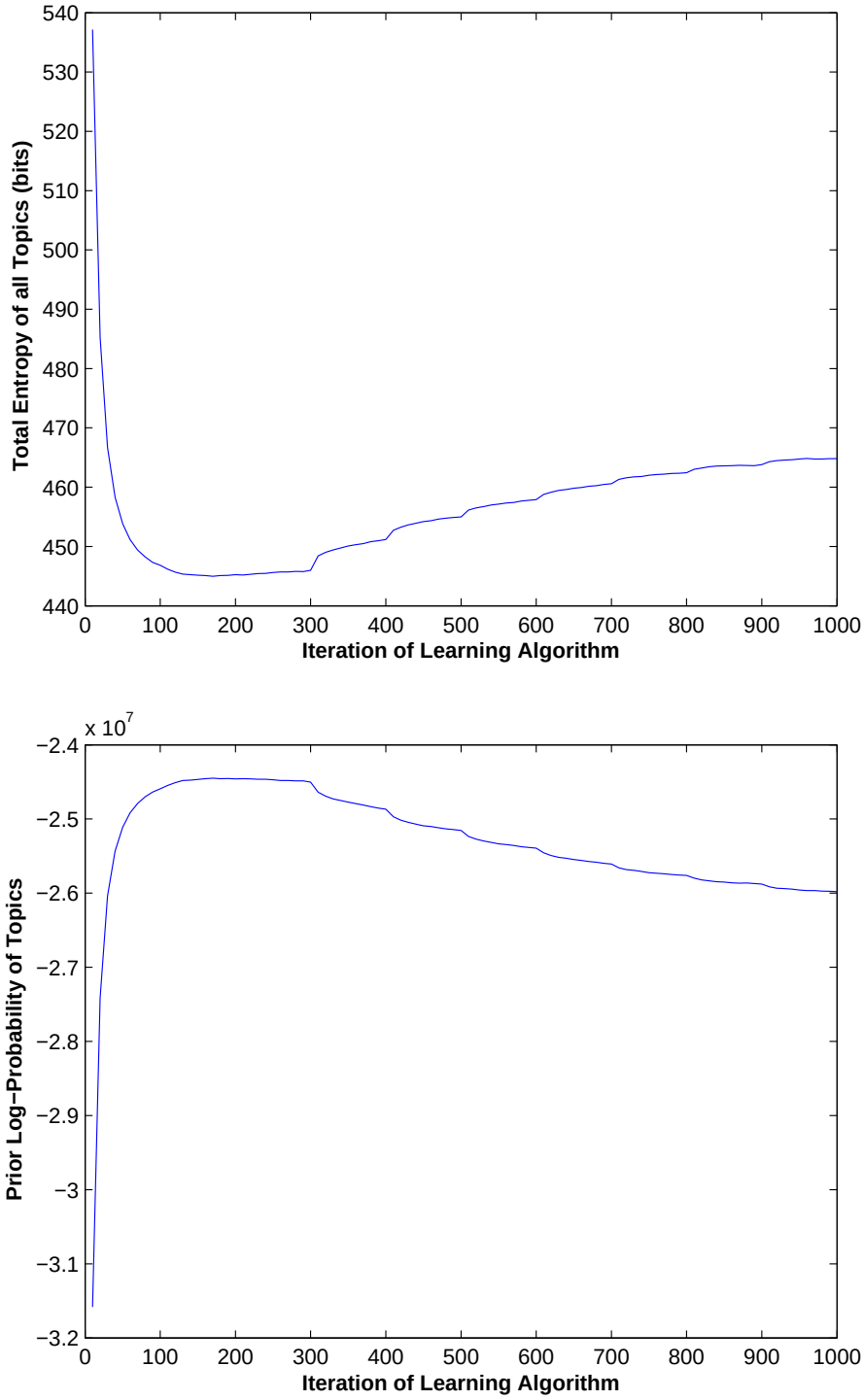


Figure 5.11: Total entropy and prior probability of the topics per iteration of the collapsed Gibbs sampling learning algorithm (ACL corpus). The ridges at 100 iteration intervals correspond to hyper-parameter optimizations.

the probability of observed data after marginalizing hidden variables, while PF MCMC-MLE estimates the *gradient* of the log-likelihood of an undirected model with respect to its parameters based on these samples. Asuncion *et al.* use the resulting Monte Carlo estimate of the gradient to perform a stochastic gradient update, and repeat this process in order to learn the parameters of the model.

Unlike PF MCMC-MLE, iteration-AIS introduces a sequence of AIS *intermediate* distributions between the distributions implied by the models at each training iteration, using e.g. the geometric and convex paths. When the number of AIS temperatures between learned models is set to zero, the path of iteration-AIS reduces to the path of PF MCMC-MLE. Conversely, we can think of the iteration-AIS path as augmenting the PF MCMC-MLE path with intermediate distributions designed to facilitate a smooth transition between the model distributions, and to improve the quality of the resulting importance weights.

Therefore, by applying the iteration-AIS path to the particles (a.k.a. samples) of PF MCMC-MLE, we speculate that the sampling behavior of PF MCMC-MLE could potentially be improved, which may improve the overall performance of the algorithm. The cost of the method is the additional overhead of sampling from the intermediate distributions. The motivation is that spending more effort on the *inference* step may lead to less time needed in the *learning* step. By improving the quality of the particles, it may also lead to a reduction or elimination of the need for a *rejuvination* step, which Asuncion *et al.* use to improve the effective sample size. As a bonus, since the iteration-AIS path is used, the algorithm will also obtain estimates of the partition function if the initial distribution of the weights is normalized. PF MCMC-MLE can also estimate the partition function as it uses a path which is a special case of the iteration-AIS path, although without the intermediate distributions to smooth the transitions the estimate may not be as reliable as the suggested approach. We describe this potential algorithm in more detail in Appendix D.

5.5 Discussion

We have introduced ratio-AIS, a strategy for comparing topic models, and evaluated its properties relative to previous methods on two data sets of scientific articles. Ratio-AIS was found to have low empirical variance, and was able to detect convergence failures, making it useful for document-level analysis. However, importance sampling can suffer from bias with a finite number of samples, e.g. approaches such as those described in Wallach *et al.* (2009b) will typically underestimate the likelihood. For ratio-AIS in particular this results in the potential for a bias that favors a particular model when an insufficient number of samples or temperatures is used, due to the directional nature of the approach. Such a convergence failure of a Monte Carlo algorithm is in general very difficult to detect, but in the proposed method the bias is frequently easily detectable by comparing the results of two Monte Carlo runs. When applied to the evaluation of the per-iteration performance of topic model training algorithms (iteration-AIS), the method outperforms traditional approaches even when given an order of magnitude less computation.

Based on our results, we recommend ratio-AIS for document-level analysis, or in cases where the topics are very similar to each other. Left-to-right is still generally preferred for corpus-level perplexity comparisons, unless per-iteration curves are desired, in which case we recommend iteration-AIS be used. When using ratio-AIS, we recommend running the methods in both annealing directions to gain evidence that a lack of convergence does not bias the results.

If multiple topic models are to be compared, ratio-AIS can be straightforwardly used to compare all models to a single baseline model (such as vanilla LDA), and the results can be reported relative to the baseline. The comparison of multiple models gives further opportunities for convergence checking: for models a, b and c we can use ratio-AIS to compute their relative performances $a - b$ and $b - c$, then compare $(a - b) + (b - c)$ to a ratio-AIS

estimate of $a - c$. If convergence has been achieved, these results should be equal. In a more elaborate approach, an iteration-AIS-style path between any set of models in sequence gives an estimate of all models' likelihoods, although this is only beneficial relative to the standard AIS method if the models can be arranged in a sequence where each consecutive model is similar.

More sophisticated choices of annealing paths within the ratio-AIS and iteration-AIS frameworks have the potential to improve the performance of the methods. For example, Neal (2001) suggests that using a geometric spacing of the temperatures can improve performance, at least for the geometric path. If the techniques are applied to exponential family models instead of topic models, the path of Grosse *et al.* (2013) has been shown to be useful. The results in Section 5.3.2, Figures 5.7 – 5.10 and Figure 5.11 suggest that asymmetric hyper-parameters may impede mixing, which reduced the performance of all evaluation algorithms. One potential strategy to mitigate this is the concatenation of several ratio-AIS annealing paths between copies of the models with "flattened" hyper-parameters. We leave the investigation of these more complex strategies for future work.

We have assumed in this chapter that the competing topic models have the same parametric form, and the same number of topics. This is necessary for an AIS path to be defined between the models, which is a limitation of the approach. It may potentially be possible to circumvent this limitation by using a variable dimension sampling scheme such as reversible jump MCMC.

It should be noted that although we have focused on topic models here, the ideas we have suggested (annealing between models, and annealing along the sequence of models generated during training) can be applied more broadly to other models. Regardless of the form of the model, these general techniques apply whenever we are interested in ratios of the evidence, ratios of partition functions, or the value of the partition function across training iterations.

5.6 Summary of Contributions

In summary, the contributions of this chapter are:

- We developed an algorithm for comparing two topic models. The method, called ratio-AIS, anneals between the two models to compute the ratio of their likelihoods, using a Monte Carlo integration technique called annealed importance sampling (AIS).
- AIS requires as input a sequence of intermediate distributions, known as the annealing path. We showed how to use two such annealing paths for this procedure: geometric averages of the distribution (as suggested by Neal (2001), and a sequence of convex combinations of the parameters.
- We identified a strategy for detecting convergence failures in the proposed method, by performing the annealing in both directions and comparing the results.
- By applying ratio-AIS recursively to the sequence of consecutive models constructed during the training of a topic model, we showed how to efficiently and accurately evaluate the progress of topic model learning algorithms as they are trained, in a technique referred to as iteration-AIS.
- The ratio-AIS method was evaluated on two corpora of scientific articles. Compared to previous approaches, it was found that ratio-AIS had lower empirical variance and was better able to identify the differences between very similar topics. This potentially comes at the cost of an increased potential for a bias in favor of a particular model, however this is frequently detectable using the strategy mentioned above.
- We also evaluated iteration-AIS on the scientific corpora, finding that it performed better than previous approaches, which were more likely to underestimate the likelihood. In some cases, iteration-AIS found a better solution even when given an order of magnitude less computation time than its competitors.

- In more speculative work, connections were shown between iteration-AIS and the particle-filtered MCMC-MLE algorithm for maximum likelihood estimation, suggesting the potential for the method to be applied in the context of learning.

Chapter 6

Conclusions and Future Directions

The road goes ever on and on
Out from the door where it began.
Now far ahead the road has gone,
Let others follow it who can!

Let them a journey new begin,
But I at last with weary feet
Will turn towards the lighted inn,
My evening-rest and sleep to meet.
J.R.R. Tolkien, Return of the King

This dissertation has presented several models and algorithms for the latent variable analysis of network and text data. We conclude by summarizing the contributions of the thesis, suggesting potential avenues for future work, and leaving some parting thoughts.

6.1 Contributions of the Thesis

In Chapter 1, we motivated latent variable modeling as a tool for taming and harnessing digital information overload. We then described a unifying framework for latent variable methods and used it to overview the literature.

Making the concepts introduced in the first chapter more concrete, Chapter 2 by introduced *DRIFT*, a new latent variable model for social networks as they vary over time. The model posits that social interactions are explained by a set of latent features belonging to each individual, and that the individuals can gradually gain or lose features as time passes. As a nonparametric Bayesian model, the number of these latent features is potentially unbounded and can be inferred from the data. We investigated the performance of the model experimentally, finding that it performed better at prediction than previous methods. We also showed how to leverage text when building such latent feature network models, in order to interpret the latent features. The model was applied for exploratory data analysis on Enron email and Twitter data sets.

The theme of *networks and text* was continued in Chapter 3, where we developed *topical influence regression* (*TIR*), a model designed to recover influence relationships in a citation network of scientific articles. The model makes use of the text of the articles in conjunction with the citation graph, positing that influential articles coerce the articles which cite them into having similar topics to them. We evaluated TIR both quantitatively and qualitatively,

finding that the model was able to discover meaningful influence patterns at both the node level and the document level.

As latent variable techniques are increasingly used to analyze internet data, the scalability of the methods becomes crucial. In Chapter 4 we introduced *SCVB0*, an algorithm for learning topic models which is fast, accurate and scalable. The algorithm applies a scalable stochastic method to the collapsed representation of LDA, which allows for simple, efficient and accurate algorithms. We investigated its performance on several large corpora, finding that it typically outperformed the previous stochastic algorithm at predicting held-out documents, while also converging faster. The algorithm was also found to be effective at learning very quickly from small data sets according to human judgment, pointing to potential applications in exploratory data analysis. We further analyzed the algorithm theoretically, by proving its convergence, showing connections to MAP estimation algorithms, and describing an explanation for the accuracy of the approximations used.

When developing latent variable modeling techniques and algorithms, it is important to evaluate their performance. Chapter 5 introduced *ratio-AIS*, a technique for determining the relative predictive performance of two topic models (e.g. a new method and a baseline). Experimental results showed that ratio-AIS has low empirical variance, unlike previous approaches, making it useful for document-level analyses. As a trade-off, the directional nature of the algorithm gave the potential for it to have an increased directional bias in its comparisons if given insufficient computation. However, this was frequently detectable, unlike for most Monte Carlo algorithms. We also showed how to make use of this technique to efficiently evaluate the performance of topic model training algorithms, by evaluating their predictive performance over time as they are trained, in a method called *iteration-AIS*. It was found that iteration-AIS had benefits in terms of computation time, predictive performance and variance relative to approaches which did not take advantage of the sequential nature of the per-iteration evaluation task.

6.2 Future Directions

The models and algorithms we have developed in this thesis each have the potential to seed further endeavors. We will discuss some of the possibilities here.

Firstly, Chapter 2 introduced DRIFT and LFRM_LDA, which model networks over time and text-augmented networks, respectively. These ideas are somewhat orthogonal, and there is the potential to leverage both ideas together in a single model. Scaling up DRIFT and the LFRM to large data sets is an important future direction. Stochastic variational inference algorithms such as those used in Chapter 4 are an obvious choice for attempting this, along the lines of Gopalan *et al.* (2012).

The topical influence regression (TIR) model of Chapter 3 could readily be extended to capture other aspects of scientific influence, such as the effects of authors and journals on topical influence, and to exploit the context in which citations occur. From an exploratory analysis perspective, it would be instructive to compare topical influence trajectories over time for different papers. This could be further facilitated by explicitly modeling the dynamics of each article's topical influence score. The TIR framework could potentially also be applicable to other application domains such as modeling how interpersonal influence affects the spread of memes via social media.

To complement TIR, it would be useful to also have systems for identifying articles which are important for alternative reasons, such as providing methodological tools and/or demonstrating important facts. Ultimately a suite of such tools could feed into a system such as Google Scholar or Citeseer. We envision that this line of work will also be useful for building visualization tools to help researchers explore scientific corpora.

The SCVB0 method could potentially be adapted to Teh *et al.* (2006)'s hierarchical Dirichlet process version of LDA, leveraging the work of Sato *et al.* (2012). An initial attempt at this

has been made in a NIPS workshop paper by Bleier (2013). The speed and scalability of the method (in terms of vocabulary size and the number of topics) could likely be improved by exploiting sparsity, using techniques such as those employed by Mimno *et al.* (2012). Furthermore, the collapsed representation facilitates the use of the parallelization techniques explored by Newman *et al.* (2009) and Smola & Narayanamurthy (2010). It may also be possible to generalize the ideas of SCVB0 to models other than LDA. In this direction, the approximate mean field framework of Asuncion (2010) is one possible starting point. A potentially very useful application of SCVB0 is to incorporate it into an interactive software tool for exploring the topics of document corpora in real-time.

Regarding the annealed importance sampling methods introduced in Chapter 5, these ideas are likely to be useful for other latent variable models such as RBMs. As mentioned in the chapter and in Appendix D, iteration-AIS may be useful in a learning context, e.g. by improving the inference inner loop in the particle-filtered MCMC-MLE algorithm of Asuncion *et al.* (2010). It may also be possible to find other AIS paths with better mixing properties, or different trade-offs between bias and variance.

6.3 Parting Thoughts

Speaking more broadly, there will always be a need to model, interpret and predict high-dimensional data. Latent variable modeling is here to stay. In this thesis, we have developed latent variable models and the algorithms to fit and evaluate them, in order to find meaningful and predictive latent representations. Further advances in model building and model fitting will allow us to continue to find interesting and useful patterns in new kinds of data.

To make these methods easier to develop, refine, and use, probabilistic programming systems have the potential to revolutionize the field. These systems, including WinBUGS

(Lunn *et al.* , 2000), JAGS[1] and Stan (Stan Development Team, 2014), allow the specification of a model in a simple programming language, and then automatically provide an inference algorithm to fit the model. This brings the potential to broaden the audience of these methods within adjacent fields of science by greatly reducing the level of expert machine learning knowledge needed to build and use them. As the projects mature, and automatic general-purpose inference algorithms become more scalable, we are likely to see a renaissance of the field of latent variable modeling, and machine learning in general. Along these lines, a very recent advance due to Ranganath *et al.* (2014) aims to make stochastic variational methods such as those discussed in Chapter 4 more generally and easily applicable, in order to ease the development of new models.

Complementing the special-purpose hand-designed latent variable models we have discussed, there is another trend towards the development of *general-purpose* latent variable models, such as those studied in the deep learning community. Improvements in hardware and recent algorithmic developments such as the dropout algorithm (Hinton *et al.* , 2012) have led to a number of recent successes with these methods, not to mention a string of high-profile hires of deep learning experts by leading technology companies. As with special-purpose latent variable models, these techniques can hope to gain even broader adoption as they mature beyond a reliance on expert knowledge in order to achieve good performance in practice.

This dissertation has taken several small steps towards the overall goal of gaining insight into data in an increasingly data-driven world. We anticipate that latent variable modeling will continue to provide new ways to help us achieve this, and look forward to exciting developments yet to come.

[1] `http://mcmc-jags.sourceforge.net/`

Bibliography

A'Hearn, B. 2004. A restricted maximum likelihood estimator for truncated height samples. *Economics & Human Biology*, **2**(1), 5–19.

Ahmed, Amr, Ho, Qirong, Teo, Choon Hui, Eisenstein, Jacob, Smola, Alex, & Xing, Eric. 2011. Online inference for the infinite topic-cluster model: Storylines from streaming text. *Pages 101–109 of: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*.

Airoldi, E.M., Blei, D.M., Feinberg, S.E., & Xing, E.P. 2008. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, **9**, 1981–2014.

Aldous, David. 1985. Exchangeability and related topics. *Pages 1–198 of: École d'Été de Probabilités de Saint-Flour XIII – 1983*. Lecture Notes in Mathematics, vol. 1117. Springer.

Amari, Shun-Ichi. 1998. Natural gradient works efficiently in learning. *Neural Computation*, **10**(2), 251–276.

Andrieu, Christophe, Moulines, Éric, & Priouret, Pierre. 2005. Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization*, **44**(1), 283–312.

Asuncion, Arthur. 2010. Approximate mean field for Dirichlet-based models. *In: ICML Workshop on Topic Models*.

Asuncion, Arthur, Welling, Max, Smyth, Padhraic, & Teh, Yee Whye. 2009. On smoothing and inference for topic models. *Pages 27–34 of: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Asuncion, Arthur U, Liu, Qiang, Ihler, Alexander T, & Smyth, Padhraic. 2010. Particle filtered MCMC-MLE with connections to contrastive divergence. *Pages 47–54 of: Proceedings of the 27th International Conference on Machine Learning*.

Balasubramanyan, Ramnath, & Cohen, William W. 2011. Block-LDA: Jointly modeling entity-annotated text and entity-entity links. *Pages 450–461 of: Proceedings of the SIAM International Conference on Data Mining*.

Banerjee, Arindam, & Basu, Sugato. 2007. Topic models over text streams: A study of batch and online unsupervised learning. *Pages 437–442 of: Proceedings of the SIAM International Conference on Data Mining*.

Beal, Matthew J, Ghahramani, Zoubin, & Rasmussen, Carl Edward. 2002. The infinite hidden Markov model. *Pages 577–584 of: Advances in Neural Information Processing Systems 14*.

Bell, Robert M, & Koren, Yehuda. 2007. Lessons from the Netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, **9**(2), 75–79.

Bezanson, Jeff, Karpinski, Stefan, Shah, Viral B., & Edelman, Alan. 2012. Julia: A fast dynamic language for technical computing. *Computing Research Repository*, **abs/1209.5145**.

Bishop, Christopher M. 1998. Latent variable models. *Pages 371–403 of:* Jordan, M. (ed), *Learning in Graphical Models*. Springer.

Bishop, Christopher M, *et al.* . 2006. *Pattern Recognition and Machine Learning*. Springer New York.

Blei, D.M., Ng, A.Y., & Jordan, M.I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, **3**, 993–1022.

Bleier, Arnim. 2013. Practical collapsed stochastic variational inference for the HDP. *In: Proceedings of the 2013 NIPS Workshop on Topic Models*.

Bottou, Léon. 1998. Online algorithms and stochastic approximations. *In:* Saad, David (ed), *Online Learning and Neural Networks*. Cambridge University Press. revised, oct 2012.

Bottou, Léon, & LeCun, Yann. 2003. Large scale online learning. *Pages 217–224 of: Advances in Neural Information Processing Systems 16*.

Box, George EP, & Draper, Norman R. 1987. *Empirical Model-Building and Response Surfaces*. John Wiley & Sons.

Brin, S., & Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, **30**(1–7), 107–117.

Buntine, W. 2009. Estimating likelihoods for topic models. *Pages 51–64 of: Advances in Machine Learning*. Springer.

Buntine, W., & Jakulin, A. 2006. Discrete component analysis. *Pages 1–33 of: Statistical and Optimization Perspectives Workshop on Subspace, Latent Structure and Feature Selection, LNCS 3940*. Springer.

Butts, C.T. 2008. A relational event framework for social action. *Sociological Methodology*, **38**(1), 155–200.

Canny, John. 2004. GaP: a factor model for discrete data. *Pages 122–129 of: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Cappé, O., & Moulines, E. 2009. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **71**(3), 593–613.

Carpenter, B. 2010. *Integrating Out Multinomial Parameters in Latent Dirichlet Allocation and Naive Bayes for Collapsed Gibbs Sampling*. Tech. rept. LingPipe.

Chang, J., & Blei, D. 2009. Relational topic models for document networks. *Pages 81–88 of: Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*.

Chang, Jonathan, Boyd-Graber, Jordan, Gerrish, Sean, Wang, Chong, & Blei, David. 2009. Reading tea leaves: How humans interpret topic models. *Pages 288–296 of: Advances in Neural Information Processing Systems 22*.

Cisco Systems. 2014. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018*. Tech. rept.

Cohn, D., & Hofmann, T. 2001. The missing link – A probabilistic model of document content and hypertext connectivity. *Pages 430–436 of: Advances in Neural Information Processing Systems 13*.

Dayan, Peter, Hinton, Geoffrey E, Neal, Radford M, & Zemel, Richard S. 1995. The Helmholtz machine. *Neural Computation*, **7**(5), 889–904.

Deerwester, Scott C., Dumais, Susan T, Landauer, Thomas K., Furnas, George W., & Harshman, Richard A. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, **41**(6), 391–407.

Dempster, Arthur P, Laird, Nan M, Rubin, Donald B, *et al.* . 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1), 1–38.

Dietz, L., Bickel, S., & Scheffer, T. 2007. Unsupervised prediction of citation influences. *Pages 233–240 of: Proceedings of the 24th International Conference on Machine Learning*.

DuBois, Christopher, Foulds, James R, & Smyth, Padhraic. 2011. Latent Set Models for Two-Mode Network Data. *Pages 137–144 of: Fifth International AAAI Conference on Weblogs and Social Media*.

Egghe, Leo. 2006. Theory and practise of the g-index. *Scientometrics*, **69**(1), 131–152.

El-Arini, Khalid, & Guestrin, Carlos. 2011 (August). Beyond keyword search: Discovering relevant scientific literature. *Pages 439–447 of: Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Erosheva, Elena, Fienberg, Stephen, & Lafferty, John. 2004. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, **101**(Suppl 1), 5220–5227.

Fan, Yu, & Shelton, Christian R. 2009. Learning continuous-time social network dynamics. *Pages 161–168 of: Proceedings of the Twenty-Fifth International Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Feld, Scott L. 1981. The focused organization of social ties. *American Journal of Sociology*, **86**(5), 1015.

Ferguson, Thomas S. 1973. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, **1**(2), 209–230.

Fienberg, Steven E., & Wasserman, Stanley. 1981. Categorical data analysis of single sociometric relations. *Sociological Methodology*, **12**, 156–192.

Foulds, J. R., Boyles, L., DuBois, C., Smyth, P., & Welling, M. 2013. Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation. *Pages 446–454 of: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Foulds, James, & Smyth, Padhraic. 2013. Modeling scientific impact with topical influence regression. *Pages 113–123 of: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics.

Foulds, James R, DuBois, Christopher, Asuncion, Arthur U, Butts, Carter T, & Smyth, Padhraic. 2011. A dynamic relational infinite feature model for longitudinal social networks. *Pages 287–295 of: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*.

Fu, Wenjie, Song, Le, & Xing, Eric P. 2009. Dynamic mixed membership blockmodel for evolving networks. *Pages 329–336 of: Proceedings of the 26th International Conference on Machine Learning*. New York, New York, USA: ACM Press.

Gelfand, Alan E, & Smith, Adrian FM. 1990. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, **85**(410), 398–409.

Geman, Stuart, & Geman, Donald. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 721–741.

Gerrish, S., & Blei, D.M. 2010. A language-based approach to measuring scholarly impact. *Pages 375–382 of: Proceedings of the 26th International Conference on Machine Learning*.

Ghahramani, Zoubin, & Jordan, Michael I. 1997. Factorial hidden Markov models. *Machine Learning*, **29**(2-3), 245–273.

Gneiting, Tilmann, & Raftery, Adrian E. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, **102**(477), 359–378.

Gopalan, Prem, Gerrish, Sean, Freedman, Michael, Blei, David M, & Mimno, David M. 2012. Scalable inference of overlapping communities. *Pages 2258–2266 of: Advances in Neural Information Processing Systems 25*.

Griffiths, T., & Ghahramani, Z. 2005. *Infinite Latent Feature Models and the Indian Buffet Process*. Tech. rept. GCNU TR 2005-001. Gatsby Computational Neuroscience Unit, University College London.

Griffiths, T., & Ghahramani, Z. 2006. Infinite latent feature models and the Indian buffet process. *Pages 475–482 of: Advances in Neural Information Processing Systems 18*.

Griffiths, Thomas L, Steyvers, Mark, & Tenenbaum, Joshua B. 2007. Topics in semantic representation. *Psychological Review*, **114**(2), 211.

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

Grimmer, Justin. 2010. A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in Senate press releases. *Political Analysis*, **18**(1), 1–35.

Grosse, Roger, Maddison, Chris, & Salakhutdinov, Ruslan. 2013. Annealing between distributions by averaging moments. *Pages 2769–2777 of: Advances in Neural Information Processing Systems 26*.

Handcock, Mark, Robins, Garry, Snijders, Tom, & Besag, Julian. 2003. Assessing degeneracy in statistical models of social networks. *Journal of the American Statistical Association*, **76**, 33–50.

Hastings, W Keith. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**(1), 97–109.

Haveliwala, Taher H. 2002. Topic-sensitive PageRank. *Pages 517–526 of: Proceedings of the 11th International Conference on World Wide Web*. ACM.

He, Q., Chen, B., Pei, J., Qiu, B., Mitra, P., & Giles, L. 2009. Detecting topic evolution in scientific literature: How can citations help? *Pages 957–966 of: Proceedings of the 18th ACM Conference on Information and Knowledge Management*. ACM.

Heaukulani, Creighton, & Ghahramani, Zoubin. 2013. Dynamic probabilistic models for latent feature propagation in social networks. *Pages 275–283 of: Proceedings of the 30th International Conference on Machine Learning*.

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

Hinton, Geoffrey E, & Van Camp, Drew. 1993. Keeping the neural networks simple by minimizing the description length of the weights. *Pages 5–13 of: Proceedings of the Sixth Annual Conference on Computational Learning Theory*. ACM.

Hinton, Geoffrey E, & Zemel, Richard S. 1994. Autoencoders, minimum description length, and Helmholtz free energy. *Pages 3–10 of: Advances in Neural Information Processing Systems 6*.

Hinton, Geoffrey E, Srivastava, Nitish, Krizhevsky, Alex, Sutskever, Ilya, & Salakhutdinov, Ruslan R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hirsch, Jorge E. 2005. An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences of the United States of America*, **102**(46), 16569–16572.

Hoff, Peter. 2007. Modeling homophily and stochastic equivalence in symmetric relational data. *Pages 657–664 of: Advances in Neural Information Processing Systems 20*.

Hoff, Peter, Raftery, Adrian E, & Handcock, Mark S. 2002. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, **97**(460), 1090–1098.

Hoffman, Matt, Blei, David M, Wang, Chong, & Paisley, John. 2013. Stochastic variational inference. *Journal of Machine Learning Research*, **14**(1), 1303–1347.

Hoffman, Matthew, Bach, Francis R, & Blei, David M. 2010. Online learning for latent Dirichlet allocation. *Pages 856–864 of: Advances in Neural Information Processing Systems 23*.

Hofmann, Thomas. 1999a. Probabilistic latent semantic analysis. *Pages 289–296 of: Proceedings of the Fifteenth International Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc.

Hofmann, Thomas. 1999b. Probabilistic latent semantic indexing. *Pages 50–57 of: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Hofmann, Thomas, Puzicha, Jan, & Jordan, Michael I. 1998. Learning from dyadic data. *Pages 466–472 of: Advances in Neural Information Processing Systems 11*.

Hoover, David N. 1982. Row-column exchangeability and a generalized model for probability. *Exchangeability in Probability and Statistics, North-Holland, Amsterdam*, 81–291.

Hotelling, Harold. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, **24**(6), 417.

Jaakkola, T, & Jordan, M. 1997. A variational approach to Bayesian logistic regression models and their extensions. *In: Sixth International Workshop on Artificial Intelligence and Statistics*.

Jordan, Michael I, Ghahramani, Zoubin, Jaakkola, Tommi S, & Saul, Lawrence K. 1999. An introduction to variational methods for graphical models. *Machine Learning*, **37**(2), 183–233.

Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., & Ueda, N. 2006. Learning systems of concepts with an infinite relational model. *Pages 381–388 of: Proceedings of the Twenty-First National Conference on Artificial Intelligence*.

Kim, Myunghwan, & Leskovec, Jure. 2013. Nonparametric multi-group membership model for dynamic networks. *Pages 1385–1393 of:* Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K.Q. (eds), *Advances in Neural Information Processing Systems 26*.

Kleinberg, Jon M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, **46**(5), 604–632.

Klimt, B., & Yang, Y. 2004. Introducing the Enron corpus. *In: First Conference on Email and Anti-Spam (CEAS)*.

Knowles, David, & Ghahramani, Zoubin. 2007. Infinite sparse factor analysis and infinite independent components analysis. *Pages 381–388 of: Independent Component Analysis and Signal Separation*. Springer.

Krivitsky, Pavel N, Handcock, Mark S, Raftery, Adrian E, & Hoff, Peter D. 2009. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, **31**(3), 204–213.

Kuhn, H.W. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, **2**(1-2), 83–97.

Larsen, Peder Olesen, & von Ins, Markus. 2010. The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics*, **84**(3), 575–603.

Le Cun, B.B., Denker, JS, Henderson, D., Howard, RE, Hubbard, W., & Jackel, LD. 1990. Handwritten digit recognition with a back-propagation network. *Pages 396–404 of: Advances in Neural Information Processing Systems 2*.

Le Roux, Nicolas, Schmidt, Mark W, Bach, Francis, *et al.* . 2012. A stochastic gradient method with an exponential convergence rate for finite training sets. *Pages 2672–2680 of: Advances in Neural Information Processing Systems 25*.

Lin, J. 2008. PageRank without hyperlinks: Reranking with PubMed related article networks for biomedical text retrieval. *BMC Bioinformatics*, **9**(1), 270.

Lunn, David J, Thomas, Andrew, Best, Nicky, & Spiegelhalter, David. 2000. WinBUGS–A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, **10**(4), 325–337.

McCallum, A. 2002. *MALLET: A Machine Learning for Language Toolkit.* http://mallet.cs.umass.edu.

McCallum, A., Wang, X., & Corrada-Emmanuel, A. 2007. Topic and role discovery in social networks with experiments on Enron and academic email. *Journal of Artificial Intelligence Research*, **30**(1), 249–272.

Mccullagh, P, & Nelder, John A. 1989. *Generalized Linear Models.* Monographs on Statistics and Applied Probability, no. 37. Chapman and Hall.

McFarland, Daniel A, Ramage, Daniel, Chuang, Jason, Heer, Jeffrey, Manning, Christopher D, & Jurafsky, Daniel. 2013. Differentiating language usage through topic models. *Poetics*, **41**(6), 607–625.

Meeds, Edward, Ghahramani, Zoubin, Neal, Radford, & Roweis, Sam. 2007. Modeling dyadic data with binary latent factors. *Pages 977–984 of: Advances in Neural Information Processing Systems 19.*

Merton, Robert K. 1968. The Matthew effect in science. *Science*, **159**(3810), 56–63.

Metropolis, Nicholas, Rosenbluth, Arianna W, Rosenbluth, Marshall N, Teller, Augusta H, & Teller, Edward. 2004. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, **21**(6), 1087–1092.

Miller, K.T., Jordan, M.I., & Griffiths, T.L. 2009. Nonparametric latent feature models for link prediction. *Pages 1276–1284 of: Advances in Neural Information Processing Systems 22.*

Mimno, D., & McCallum, A. 2008. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. *Pages 411–418 of: Proceedings of the Twenty-Fourth International Conference on Uncertainty in Artificial Intelligence.*

Mimno, David. 2011. Reconstructing Pompeian households. *Pages 506–513 of: Proceedings of the Twenty-Seventh Annual Conference on Uncertainty in Artificial Intelligence.*

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

Mimno, David, Wallach, Hanna M, Talley, Edmund, Leenders, Miriam, & McCallum, Andrew. 2011. Optimizing semantic coherence in topic models. *Pages 262–272 of: Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics.

Mimno, David, Hoffman, Matt, & Blei, David. 2012. Sparse stochastic inference for latent Dirichlet allocation. *Pages 1599–1606 of:* Langford, John, & Pineau, Joelle (eds), *Proceedings of the 29th International Conference on Machine Learning.* New York, NY, USA: Omnipress.

Minka, Thomas. 2004. *Power EP*. Tech. rept. MSR-TR-2004-149. Microsoft Research, Cambridge, UK.

Minka, Thomas P. 2001. Expectation propagation for approximate Bayesian inference. *Pages 362–369 of: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc.

Minka, T.P. 2000. *Estimating a Dirichlet Distribution*. Tech. rept. Microsoft Research.

Moreno, Jacob Levy. 1934. *Who Shall Survive?: A New Approach to the Problem of Human Interrelations*. Nervous and Mental Disease Monograph Series, no. 58. Nervous and Mental Disease Publishing Co.

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Myers, Seth A, & Leskovec, Jure. 2014. The bursty dynamics of the Twitter information network. *Pages 913–924 of: Proceedings of the 23rd International Conference on the World Wide Web*.

Nallapati, R., McFarland, D., & Manning, C. 2011. Topicflow model: Unsupervised learning of topic-specific influences of hyperlinked documents. *Pages 543–551 of: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*.

Neal, Radford M. 1992. Bayesian mixture modeling. *Pages 197–211 of: Maximum Entropy and Bayesian Methods*. Springer.

Neal, Radford M. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, **9**(2), 249–265.

Neal, Radford M. 2001. Annealed importance sampling. *Statistics and Computing*, **11**(2), 125–139.

Neal, Radford M. 2003. Slice sampling. *The Annals of Statistics*, 705–741.

Neal, Radford M, & Hinton, Geoffrey E. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Pages 355–368 of:* Jordan, M. (ed), *Learning in Graphical Models*. Springer.

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

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

Newton, Michael A, & Raftery, Adrian E. 1994. Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society: Series B (Methodological)*, 3–48.

Nguyen, Viet-An, Boyd-Graber, Jordan, Resnik, Philip, Cai, Deborah A, Midberry, Jennifer E, & Wang, Yuanxin. 2013. Modeling topic control to detect influence in conversations using nonparametric topic models. *Machine Learning*, 1–41.

Nowicki, Krzysztof, & Snijders, Tom A B. 2001. Estimation and prediction of stochastic blockstructures. *Journal of the American Statistical Association*, **96**(455), 1077– 1087.

Orchard, T., & Woodbury, M. A. 1972. A missing information principle: Theory and applications. *Pages 697–715 of: Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics*.

Pearson, Karl. 1901. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **2**(11), 559–572.

Porteous, Ian, Newman, David, Ihler, Alexander, Asuncion, Arthur, Smyth, Padhraic, & Welling, Max. 2008. Fast collapsed Gibbs sampling for latent Dirichlet allocation. *Pages 569–577 of: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Price, Derek J. de S. 1963. *Little Science, Big Science*. New York: Columbia University Press.

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

Radev, Dragomir R., Muthukrishnan, Pradeep, Qazvinian, Vahed, & Abu-Jbara, Amjad. 2013. The ACL anthology network corpus. *Language Resources and Evaluation*, 1–26.

Ranganath, Rajesh, Gerrish, Sean, & Blei, David M. 2014. Black box variational inference. *In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*. In press.

Rasmussen, Carl Edward. 1999. The infinite Gaussian mixture model. *Pages 554–560 of: Advances in Neural Information Processing Systems 12*.

Robbins, Herbert, & Monro, Sutton. 1951. A stochastic approximation method. *The Annals of Mathematical Statistics*, 400–407.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. 2004. The author-topic model for authors and documents. *Pages 487–494 of: Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Rosner, F, Hinneburg, A, Röder, M, Nettling, M, & Both, A. 2013. Evaluating topic coherence measures. *In: Proceedings of the 2013 NIPS Workshop on Topic Models*.

Sarkar, P., & Moore, A.W. 2005. Dynamic social network analysis using latent space models. *ACM SIGKDD Explorations Newsletter: Special Edition on Link Mining*, **7**(2), 31–40.

Sarkar, P., Siddiqi, S.M., & Gordon, G.J. 2007. A latent space approach to dynamic embedding of co-occurrence data. *Pages 420–427 of: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*.

Sato, Issei, & Nakagawa, Hiroshi. 2012. Rethinking collapsed variational Bayes inference for LDA. *Pages 999–1006 of:* Langford, John, & Pineau, Joelle (eds), *Proceedings of the 29th International Conference on Machine Learning*. New York, NY, USA: Omnipress.

Sato, Issei, Kurihara, Kenichi, & Nakagawa, Hiroshi. 2012. Practical collapsed variational Bayes inference for hierarchical Dirichlet process. *Pages 105–113 of: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Scott, J.G., & Baldridge, J. 2013. A recursive estimate for the predictive likelihood in a topic model. *Pages 527–535 of: Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*.

Scott, S.L. 2002. Bayesian hidden Markov models : Recursive computing in the 21st century. *Journal of the American Statistical Association*, **97**(457), 337– 351.

Sethuraman, Jayaram. 1994. A constructive definition of Dirichlet priors. *Statistica Sinica*, **4**, 639–650.

Shaparenko, B., & Joachims, T. 2009. Identifying the original contribution of a document via language modeling. *Pages 350–365 of: Machine Learning and Knowledge Discovery in Databases*. Springer.

Simmel, G. 1955. *Conflict and the Web of Group Affiliations*. The Free Press.

Smola, Alexander, & Narayanamurthy, Shravan. 2010. An architecture for parallel topic models. *Pages 703–710 of: Proceedings of the VLDB Endowment, 36th International Conference on Very Large Data Bases*, vol. 3.

Smolensky, P. 1986. Information processing in dynamical systems: foundations of harmony theory. *Pages 194–281 of: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*. MIT Press.

Snijders, T.A.B. 2006. Statistical methods for network dynamics. *Pages 281–296 of: Proceedings of the XLIII Scientific Meeting, Italian Statistical Society*.

Spearman, Charles. 1904. "General intelligence," objectively determined and measured. *The American Journal of Psychology*, **15**(2), 201–292.

Spiro, Emma S., Fitzhugh, Sean, Sutton, Jeannette, & Butts, Carter T. 2011. *Hazards, Emergency Response, and Online Informal Communication (HEROIC) Project Data Set: Emergency Management Accounts - Messages and Relationships*. Electronic data file.

Stan Development Team. 2014. *Stan: A C++ Library for Probability and Sampling, Version 2.2*.

Sutskever, Ilya, & Tieleman, Tijmen. 2010. On the convergence properties of contrastive divergence. *Pages 789–795 of: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*.

Sutton, Jeannette, Spiro, Emma, Butts, Carter, Fitzhugh, Sean, Johnson, Britta, & Greczek, Matt. 2013. Tweeting the spill: Online informal communications, social networks, and conversational microstructures during the Deepwater Horizon oilspill. *International Journal of Information Systems for Crisis Response and Management*, **5**(1), 58–76.

Teh, Yee Whye, Welling, Max, Osindero, Simon, & Hinton, Geoffrey E. 2003. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, **4**, 1235–1260.

Teh, Yee Whye, Jordan, Michael I, Beal, Matthew J, & Blei, David M. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, **101**(476).

Teh, Y.W., Newman, D., & Welling, M. 2007a. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. *Pages 1353–1360 of: Advances in Neural Information Processing Systems 19*. MIT; 1998.

Teh, Y.W., Görür, D., & Ghahramani, Z. 2007b. Stick-breaking construction for the Indian buffet process. *Pages 556–563 of: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*.

Teufel, S., Siddharthan, A., & Tidhar, D. 2006. Automatic classification of citation function. *Pages 103–110 of: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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

Tieleman, Tijmen. 2008. Training restricted Boltzmann machines using approximations to the likelihood gradient. *Pages 1064–1071 of: Proceedings of the 25th International Conference on Machine Learning*. ACM.

Tipping, Michael E, & Bishop, Christopher M. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **61**(3), 611–622.

Titsias, Michalis K. 2008. The infinite gamma-Poisson feature model. *Pages 1513–1520 of: Advances in Neural Information Processing Systems 20*.

Van Gael, J., Teh, Y.W., & Ghahramani, Z. 2009. The infinite factorial hidden Markov model. *Pages 1697 – 1704 of: Advances in Neural Information Processing Systems 21*.

Van Gael, Jurgen. 2011. *Bayesian Nonparametric Hidden Markov Models*. Ph.D. thesis, University of Cambridge.

Wallach, Hanna M, Mimno, David M, & McCallum, Andrew. 2009a. Rethinking LDA: Why priors matter. *Pages 1973–1981 of: Advances in Neural Information Processing Systems 22*.

Wallach, H.M. 2006. Topic modeling: Beyond bag-of-words. *Pages 977–984 of: Proceedings of the 23rd International Conference on Machine Learning*. ACM.

Wallach, H.M., Murray, I., Salakhutdinov, R., & Mimno, D. 2009b. Evaluation methods for topic models. *Pages 1105–1112 of: Proceedings of the 26th International Conference on Machine Learning*. ACM.

Wang, C., & Blei, D. 2011. Collaborative topic modeling for recommending scientific articles. *Pages 448–456 of: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Wang, Chong, & Blei, David M. 2009. Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process. *Pages 1982–1989 of: Advances in Neural Information Processing Systems 22*.

Wasserman, Stanley. 1994. *Social Network Analysis: Methods and Applications*. Cambridge university press.

Wasserman, Stanley, & Pattison, Philippa. 1996. Logit models and logistic regressions for social networks: I. An introduction to Markov graphs and p^*. *Psychometrika*, **61**(3), 401–425.

Williamson, S., Wang, C., Heller, K., & Blei, D. 2010. The IBP compound Dirichlet process and its application to focused topic modeling. *Pages 1151–1158 of: Proceedings of the 27th International Conference on Machine Learning*.

Yao, Limin, Mimno, David, & McCallum, Andrew. 2009. Efficient methods for topic model inference on streaming document collections. *Pages 937–946 of: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Zhang, XianXing, & Carin, Lawrence. 2012. Joint modeling of a matrix with associated text via latent binary features. *Pages 1565–1573 of: Advances in Neural Information Processing Systems 25*.

Zhu, Xiaodan, Turney, Peter, Lemire, Daniel, & Vellino, André. 2014. Measuring academic influence: Not all citations are equal. *Journal of the Association for Information Science and Technology*. In press.

Ziman, J.M. 1968. *Public Knowledge: An Essay Concerning the Social Dimension of Science*. Cambridge University Press.

Appendices

Appendix A

Details of LFRM_LDA

To recover latent features and their semantics, we propose to jointly model a network sociomatrix $\mathbf{Y}$ and a collection of text documents ω comprising the communications within that network. Within ω, it is assumed that there are *edge documents* $\omega^{(ij)}$, each associated with an edge (i, j) in the network, and *node documents* $\omega^{(i)}$, each associated with a node i in the network. The model builds upon the (finite) LFRM and latent Dirichlet allocation. Extensions to infinite dimensional latent features and time-varying data are also possible, but we focus on the simple case here.

A.1 Generative Model

The generative process of the model is assumed to be as follows. First, the finite LFRM generative process is performed, generating binary vector representations in the latent $N \times K$ matrix $\mathbf{Z}$ and an $N \times N$ observed network $\mathbf{Y}$, by way of Equations 2.3 – 2.10.

Next, an LDA topic $\Phi^{(k)}$ is generated for each of the latent features k. Text documents ω are then generated for one or more of the edges and the nodes of $\mathbf{Y}$.[1] The text documents $\omega^{(ij)}$, $\omega^{(i)}$ are generated via LDA with a unique Dirichlet prior $\alpha^{(ij)}$, $\alpha^{(i)}$ on the distribution over topics for each document. The Dirichlet parameters are chosen such that the topics corresponding to the latent features belonging to the actors associated with the document get the most weight in the prior. This modeling strategy is similar to the Dirichlet-multinomial regression (DMR) model of Mimno & McCallum (2008), except that the Dirichlet parameters are selected based on latent features instead of observed features.

We assume that all documents have the same total Dirichlet prior concentration α^+, and that a proportion γ of the prior weight comes from the topics of the latent features of the entities associated with the document, with the remaining weight coming from a flat distribution over all the topics. For documents $\omega^{(ij)}$ on edges, the γ proportion of the prior weight is divided beween i's features and j's features with proportion λ going to i's features. This leads to K-dimensional Dirichlet priors over the K topics with parameters

$$\alpha_k^{(ij)} = \alpha^+\Big(\frac{\gamma\lambda}{\sum_{k'} z_{ik'}} z_{ik} + \frac{\gamma(1-\lambda)}{\sum_{k'} z_{jk'}} z_{jk} + \frac{(1-\gamma)}{K}\Big) \tag{A.1}$$

$$\alpha_k^{(i)} = \alpha^+\Big(\frac{\gamma}{\sum_{k'} z_{ik'}} z_{ik} + \frac{(1-\gamma)}{K}\Big)\ . \tag{A.2}$$

The documents are then sampled according to the LDA generative process.

$\theta^{(ij)} \sim \text{Dirichlet}(\alpha^{(ij)})$

For each word $\omega_l^{(ij)}$

Sample a topic $t_l^{(ij)} \sim \text{Discrete}(\theta^{(ij)})$

Sample the word $\omega_l^{(ij)} \sim \text{Discrete}(\Phi_{t_l^{(ij)}})$,

and similarly for documents on the nodes. We call this model *LFRM_LDA*.

[1] We do not model which documents are generated or their lengths, but it would be straightforward to extend the model to include this.

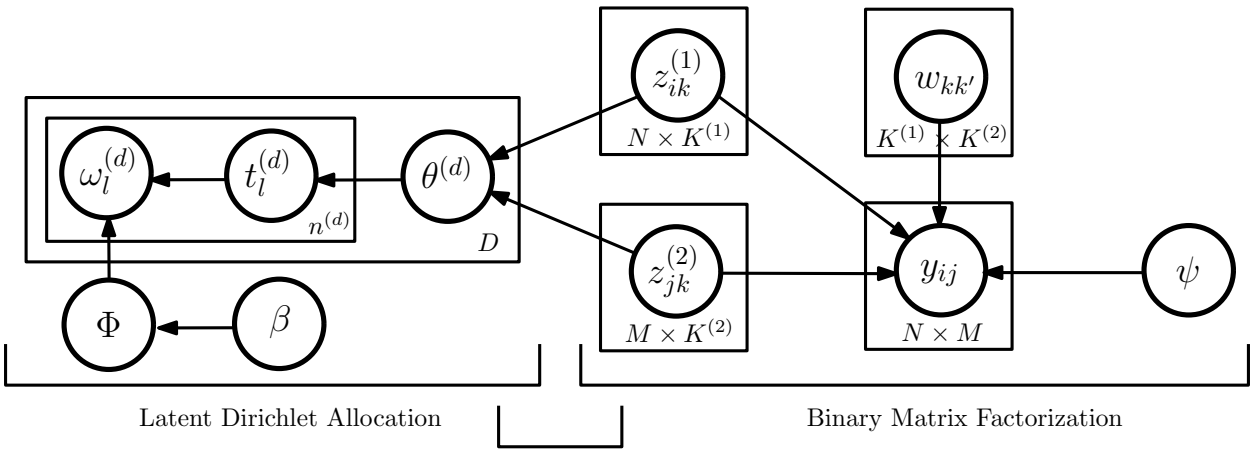


Figure A.1: Graphical model for BMF_LDA.

A.1.1 Extension to Rectangular Matrices

We can extend this model to operate on rectangular $N \times M$ data matrices $\mathbf{Y}$, using the binary matrix factorization (BMF) model of Meeds *et al.* (2007). The BMF model represents the N "row" entities and M "column" entities by latent vectors of binary features $\mathbf{Z}_i^{(1)}$ and $\mathbf{Z}_j^{(2)}$ of dimension $N \times K^{(1)}$ and $M \times K^{(2)}$, respectively. After adding effects terms as in Miller *et al.* (2009) and writing the model in the framework of Section 1.2, we can write the BMF generative process for the matrix $\mathbf{Y}$ as

$$\eta_{ij} = \mathbf{z}_i^{(1)} \mathbf{W} \mathbf{z}_j^{(2)\intercal} + \rho_i + \xi_j + \epsilon \tag{A.3}$$

$$E[y_{ij}] \triangleq \mu_{ij} = g^{-1}(\eta_{ij}) \tag{A.4}$$

$$Pr(y_{ij} | \mathbf{z}^{(1)}, \mathbf{z}_j^{(2)}, \mathbf{W}, \psi) = f_{\theta^{(f)}}(\mu_{ij}) \ , \tag{A.5}$$

where g^{-1} is an inverse link function, ρ, ξ and ϵ are optional effects and intercept terms, and $\theta^{(f)}$ contains any extra parameters relating to the likelihood such as variances. In BMF, $\mathbf{Z}_i^{(1)}$ and $\mathbf{Z}_j^{(2)}$ are given IBP priors. The same generative process for the text, conditioned on the network and latent features, can be used as for the square matrix case, but with $K = K^{(1)} + K^{(2)}$ topics and using instead the following Dirichlet priors:

$$\alpha_k^{(ij)} = \alpha^+\Big(\frac{\gamma\lambda}{\sum_{k'}\bar{z}_{ik'}^{(1)}}\bar{z}_{ik}^{(1)} + \frac{\gamma(1-\lambda)}{\sum_{k'}\bar{z}_{jk'}^{(2)}}\bar{z}_{jk}^{(2)} + \frac{(1-\gamma)}{K}\Big) \tag{A.6}$$

$$\alpha_k^{(i)} = \alpha^+\Big(\frac{\gamma}{\sum_{k'}\bar{z}_{ik'}^{(1)}}\bar{z}_{ik}^{(1)} + \frac{(1-\gamma)}{K}\Big) \tag{A.7}$$

$$\alpha_k^{(j)} = \alpha^+\Big(\frac{\gamma}{\sum_{k'}\bar{z}_{jk'}^{(2)}}\bar{z}_{ik}^{(2)} + \frac{(1-\gamma)}{K}\Big)\ , \tag{A.8}$$

where $\bar{\mathbf{Z}}^{(1)} = [\mathbf{Z}^{(1)}|\mathbf{0}_{N,K^{(2)}}]$, $\bar{\mathbf{Z}}^{(2)} = [\mathbf{0}_{M,K^{(1)}}|\mathbf{Z}^{(2)}]$ are the $\mathbf{Z}$ matrices extended to have K columns. These Dirichlet priors are equivalent to Equations 2.30 and 2.31 if $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ are equal, and "row topics" are equal to their corresponding "column topics". Thus, this version of the model is more general than LFRM_LDA, the network version described above. We call this final model BMF_LDA. The graphical model of BMF_LDA is shown in Figure A.1. To make this concrete and to demonstrate the flexibility of the framework, let us consider three examples:

1. $\mathbf{Y}$ is an $N \times N$ matrix of the non-negative integer counts of the number of emails sent between N people within a corporation within a fixed time period, and $\omega^{(ij)}$ is the sequence of words of all of the emails sent from actor i to actor j. We model $\mathbf{Y}$ using the LFRM (Miller *et al.* , 2009), with $y_{ij} \sim \text{Poisson}(\exp(\mathbf{z}_i\mathbf{W}\mathbf{z}_j^\intercal))$.

2. $\mathbf{Y}$ is an $N \times M$ matrix representing the two-mode network of N users of a recommendation system and M products (such as movies or restaurants), with y_{ij} being a real-valued rating that user i gives to product j. Text document ω_{ij} corresponds to the review that user i wrote for product j. Ratings are assumed distributed via $y_{ij} \sim \text{Gaussian}(\mathbf{z}_i^{(1)}\mathbf{W}\mathbf{z}_j^{(2)\intercal}, \sigma_g)$.

3. $\mathbf{Y}$ is an $N \times N$ binary matrix representing friendship ties in an online social media network, and ω_i is the total word counts of all microblogging status updates (e.g.

"tweets" on the Twitter social media website) by user i. A binary friendship tie y_{ij} between actors i and j exists with probability $\sigma(\mathbf{z}_i \mathbf{W} \mathbf{z}_j^\intercal)$.

A.2 Inference

We perform inference using a Gibbs sampling Markov chain Monte Carlo technique. For the LDA portion of the model we use the collapsed Gibbs sampling approach of Griffiths & Steyvers (2004), integrating out Θ and Φ. The update equations are almost identical to those of Griffiths and Steyvers, except that each document d now has its own unique topic prior parameter vector $\alpha^{(d)}$, which arises from the latent features of the entities associated with the document (d indexes over both node and edge documents):

$$Pr(t_l^{(d)} = k | \ldots) \propto (n_k^{(d)} + \alpha_k^{(d)}) \frac{n_k^{(\omega_l^d)} + \beta_k}{n_k + \sum_k \beta_k} \ , \tag{A.9}$$

where the n_k's are the counts of the occurrences of topic k over all of the entries determined by the superscript, excluding the current assignment for $t_l^{(d)}$. From the topic assignments, we can recover estimates of Φ and Θ as in Equations 6 and 7 from (Griffiths & Steyvers, 2004). The full conditionals for the latent features $z_{ik}^{(a)}$ are given by

$$\begin{aligned} Pr(z_{ik}^{(a)} = z | \ldots) \propto & \ Pr(\mathbf{Y} | z_{ik}^{(a)} = z, \mathbf{Z}_{\neg ik}^{(a)}, \mathbf{Z}^{\neg(a)}, \mathbf{W}, \psi) {\pi_k^{(a)}}^z (1 - \pi_k^{(a)})^{1-z} \\ & \prod_d Pr(t^{(d)} | z_{ik}^{(a)} = z, \mathbf{Z}_{\neg ik}^{(a)}, \mathbf{Z}^{\neg(a)}, \gamma, \lambda) \ , \end{aligned} \tag{A.10}$$

where $Pr(t^{(d)} | z_{ik}^{(a)} = z, \mathbf{Z}_{\neg ik}^{(a)}, \mathbf{Z}^{\neg(a)}, \gamma, \lambda)$ is a multivariate Polya distribution with parameter vector $\alpha^{(d)}$. This is similar to the update equation in (Meeds *et al.* , 2007), except that the conditionals are now weighted by the multivariate Polya terms, that specify the effect that each value of $z_{ik}^{(a)}$ has on the likelihood of the current topic assignments.

As in BMF, closed form updates for the entries of $\mathbf{W}$ are not available, so we resort to Metropolis-Hastings updates, with the full conditional being

$$Pr(w_{kk'}|\ldots) \propto Pr(y_{ij}|\mathbf{Z}^{(1)}, \mathbf{Z}_j^{(2)}, w_{kk'}, \mathbf{W}_{\neg kk'}, \psi) Pr(w_{kk'}|\sigma_W) \ . \tag{A.11}$$

We use a Gaussian proposal distribution, Gaussian$(w_{kk'}^{(\text{new})}; w_{kk'}^{(\text{old})}, \sigma_W)$. Updates for additional parameters such as intercept terms and row or column effects are performed similarly.

The hyper-parameters γ, λ, and α^+ for the Dirichlet priors on the topic distributions for each document are of great importance to the model (Wallach *et al.* , 2009a), but it is not clear how to choose them apriori. Instead, we follow Wallach *et al.* and take an empirical Bayes approach, optimizing them in each iteration of the MCMC scheme instead of sampling or hand-tuning them. We use gradient ascent to maximize the multivariate Polya log-likelihood of the topic assignments with respect to $[\gamma, \lambda]$, conditioned on the other parameters/variables. The topic distribution concentration parameter α^+ is optimized using an iterative procedure that maximizes a lower bound on the multivariate Polya log-likelihood (Minka, 2000):

$$\alpha^+ \leftarrow \frac{\alpha^+ \sum_d \sum_k m_k^{(d)} \big(\Psi(n_k^{(d)} + \alpha^+ m_k^{(d)}) - \Psi(\alpha^+ m_k^{(d)})\big)}{\sum_d \big(\Psi(n^{(d)} + \alpha^+) - \Psi(\alpha^+)\big)} \ , \tag{A.12}$$

where $\Psi(x) = \frac{d \log \Gamma(x)}{dx}$ is the digamma function, and $\mathbf{m}^{(d)}$ is the mean of the Dirichlet prior on document d. Similar strategies could be used to optimize β; in our experiments we simply use a flat prior with a fixed value, $\beta_k = 0.1$.

Finally, after each iteration we re-align topics with features, choosing the assignment that maximizes the multivariate Polya log-likelihood of the topic counts. Let $h : \{1, 2, \ldots K\} \rightarrow \{1, 2, \ldots K\}$ be a bijection from topics to features. Discarding terms of the log-likelihood objective function that do not depend on h,

$$
\begin{aligned}
\sum_d \log Pr(t^{(d)}|h,\ldots) &= \sum_d \Big(\sum_k \log\Gamma(n_k^{(d)} + \alpha^+ m_{h(k)}^{(d)}) - \sum_k \log\Gamma(\alpha^+ m_{h(k)}^{(d)}) \\
&\quad + \log\Gamma(\alpha^+) - \log\Gamma(n^{(d)} + \alpha^+) \Big) \\
&= \sum_d \sum_k \log\Gamma(n_k^{(d)} + \alpha^+ m_{h(k)}^{(d)}) - \sum_d \sum_k \log\Gamma(\alpha^+ m_{h(k)}^{(d)}) \\
&\quad + \text{const} \\
&= \sum_d \sum_k \log\Gamma(n_k^{(d)} + \alpha^+ m_{h(k)}^{(d)}) - \sum_d \sum_k \log\Gamma(\alpha^+ m_k^{(d)}) \\
&\quad + \text{const} \\
&= \sum_k \sum_d \log\Gamma(n_k^{(d)} + \alpha^+ m_{h(k)}^{(d)}) + \text{const} \\
&= \sum_k v_{k,h(k)} + \text{const} \ ,
\end{aligned}
$$

where $v_{k,k'} \equiv \sum_d \log\Gamma(n_k^{(d)} + \alpha^+ m_{k'}^{(d)})$ is the value of assigning topic k to feature k'. The task of finding the bijection h to maximize $\sum_k v_{k,h(k)}$ is known as the assignment problem in the combinatorial optimization literature. We use the Hungarian algorithm (Kuhn, 1955) to solve this.

Appendix B

Derivation of the Unnormalized MAP Algorithm

In this appendix we derive an EM algorithm for MAP estimation, where the parameters are represented by unnormalized count matrices. The algorithm, which we refer to as MAP_LDA_U, is due to Asuncion *et al.* (2009). Here, we give a more complete derivation of the algorithm than in Asuncion *et al.*, and show that by using a certain ordering of the EM updates, the result is an algorithm which is very similar to CVB0.

B.1 An EM Algorithm

As a warm-up, we first consider an algorithm for MAP estimation in the usual parameterization of LDA. MAP estimation aims to maximize the log posterior probability of the parameters,

$$\log Pr(\mathbf{\Theta}, \mathbf{\Phi}|w, \beta, \alpha) = \sum_{d=1}^{D}\sum_{i=1}^{N_d} \log\Big(\sum_{k=1}^{K} Pr(w_i^{(d)}, z_i^{(d)}|\theta^{(d)}, \mathbf{\Phi})\Big) + \sum_{d=1}^{D}\sum_{k=1}^{K}(\alpha-1)\log(\theta_k^{(d)}) + \sum_{w=1}^{W}\sum_{k=1}^{K}(\beta-1)\log(\Phi_w^{(k)}) + \text{ const.} \tag{B.1}$$

This objective function cannot easily be optimized directly via, e.g., a gradient update, since the log-likelihood term and its gradient require a sum over $\mathbf{z}$ inside the logarithm. Instead, EM may be performed. A standard Jensen's inequality argument gives the EM objective function as described by Neal & Hinton (1998), which, when applied to the MAP estimation problem, is a lower bound $\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}, \bar{\gamma})$ on the posterior probability (cf. Bishop *et al.* (2006)),

$$\log Pr(\mathbf{\Theta}, \mathbf{\Phi}|X) \geq \mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}, \bar{\gamma}) \triangleq R(\mathbf{\Theta}, \mathbf{\Phi}, \bar{\gamma}) - \sum_{idk} \bar{\gamma}_{idk} \log \bar{\gamma}_{idk} \ , \tag{B.2}$$

where

$$R(\mathbf{\Theta}, \mathbf{\Phi}; \Theta^{(t)}, \Phi^{(t)}) = \sum_{wk}\Big(\sum_{id:w_i^{(d)}=w} \bar{\gamma}_{idk} + \beta - 1\Big)\log \Phi_w^{(k)} + \sum_{dk}\Big(\sum_{i}\bar{\gamma}_{idk} + \alpha - 1\Big)\log\theta_k^{(d)} + \text{const} \tag{B.3}$$

is the expected complete data log-likelihood, plus terms arising from the prior, and the $\bar{\gamma}_{idk}$'s are E-step "responsibilities",

$$\bar{\gamma}_{idk} \triangleq Pr(z_i^{(d)} = k|\mathbf{\Theta}, \mathbf{\Phi}, w_i^{(d)}) \propto Pr(w_i^{(d)}|z_i^{(d)} = k, \mathbf{\Theta}, \mathbf{\Phi}) Pr(z_i^{(d)} = k|\mathbf{\Theta}, \mathbf{\Phi}) = \Phi^{(k)}_{w_i^{(d)}} \theta_k^{(d)} \ . \tag{B.4}$$

The E-step update computes these responsibility values. After adding Lagrange terms $-\sum_k \lambda_k^{\Phi}(\sum_w \Phi_w^{(k)} - 1)$ and $-\sum_d \lambda_d^{\Theta}(\sum_k \theta_k^{(d)} - 1)$ to constrain the parameter vectors to sum to one, taking derivatives and setting to zero, we obtain the following M-step updates:

$$\Phi_w^{(k)} :\propto \sum_{id:w_i^{(d)}=w} \bar{\gamma}_{idk} + \beta - 1 \qquad\qquad \theta_k^{(d)} :\propto \sum_i \bar{\gamma}_{idk} + \alpha - 1 \ . \tag{B.5}$$

B.2 An EM Algorithm with an Unnormalized Parameterization

It is possible to reparameterize the above EM algorithm for LDA in terms of unnormalized counts of the EM "responsibilities" instead of $\mathbf{\Theta}$ and $\mathbf{\Phi}$ (Asuncion *et al.* , 2009), which we refer to as the *EM statistics*. Their definitions are given in Equation 4.58, which we reproduce here:

$$\bar{N}_k^Z \triangleq \sum_{id} \bar{\gamma}_{idk} \qquad\qquad \bar{N}_{dk}^{\Theta} \triangleq \sum_i \bar{\gamma}_{idk} \qquad\qquad \bar{N}_{wk}^{\Phi} \triangleq \sum_{id:w_i^{(d)}=w} \bar{\gamma}_{idk} \ . \tag{B.6}$$

For given values of the parameters $\hat{\mathbf{\Theta}}$ and $\hat{\mathbf{\Phi}}$ obtained after an M-step update, we can rewrite these parameters in terms of the EM statistics by plugging the definitions in Equation B.6 into the M-step updated values of the parameters in Equation B.5,

$$\hat{\Phi}_w^{(k)} = \frac{\bar{N}_{wk}^{\Phi} + \beta - 1}{\bar{N}_k^Z + W(\beta - 1)} \qquad\qquad \hat{\theta}_k^{(d)} = \frac{\bar{N}_{dk}^{\Theta} + \alpha - 1}{C_d + K\alpha - K} \; . \tag{B.7}$$

If we let these EM statistics vary to some other values $\hat{\bar{\mathbf{N}}}^{\mathbf{\Phi}}$, $\hat{\bar{\mathbf{N}}}^{\mathbf{\Theta}}$, $\hat{\bar{\mathbf{N}}}^{\mathbf{Z}}$, not necessarily synchronized with the counts but having entries which sum to the number of words in the corpus C, Equation B.7 will give us back some other (suboptimal) $\hat{\mathbf{\Theta}}$ and $\hat{\mathbf{\Phi}}$. The EM bound will still hold for these other suboptimal values, which we will refer to as *estimated EM statistics*. So we can substitute Equation B.7 into the EM bound of Equation B.2 to obtain a reparameterization in terms of the (estimated) EM statistics:

$$\begin{aligned}
\log Pr(\mathbf{\Theta}, \mathbf{\Phi}|X) \geq \sum_{wk} (& \sum_{id: w_i^{(d)} = w} \bar{\gamma}_{idk} + \beta - 1) \log(\hat{\bar{N}}_{wk}^{\Phi} + \beta - 1) \\
& + \sum_{dk} (\sum_i \bar{\gamma}_{idk} + \alpha - 1) \log(\hat{\bar{N}}_{dk}^{\Theta} + \alpha - 1) \\
& - \sum_k (\sum_{id} \bar{\gamma}_{idk} + W(\beta - 1)) \log(\hat{\bar{N}}_k^Z + W(\beta - 1)) \\
& - \sum_{idk} \bar{\gamma}_{idk} \log \bar{\gamma}_{idk} + \text{const}
\end{aligned} \tag{B.8}$$

where $\hat{\bar{\mathbf{N}}}^{\mathbf{\Phi}}$, $\hat{\bar{\mathbf{N}}}^{\mathbf{\Theta}}$ and $\hat{\bar{\mathbf{N}}}^{\mathbf{Z}}$ are current estimates of the EM statistics, not necessarily synchronized with the $\bar{\gamma}$'s. These variables correspond to equivalent parameter estimates $\hat{\mathbf{\Theta}}$ and $\hat{\mathbf{\Phi}}$, so they should be understood as parameters rather than as statistics derived from $\bar{\gamma}$. We will now derive an EM algorithm which operates on this parameterization.

To derive M-step updates, we first add Lagrangian terms to enforce the constraints that each of the estimated EM statistics sums to the number of words in the corpus C, $-\lambda_\Phi(\sum_{wk} \hat{\bar{N}}_{wk}^{\Phi} - C)$, $-\lambda_\Phi(\sum_{dk} \hat{\bar{N}}_{dk}^{\Theta} - C)$, $\lambda_Z(\sum_k \hat{\bar{N}}_k^Z - C)$. In the following, we derive the update for $\hat{\bar{N}}_{wk}^{\Phi}$; the derivation is similar for the other parameters.

We take derivatives of this Lagrangian with respect to each parameter and set them to zero,

$$\frac{\sum_{id:w_i^{(d)}=w} \bar{\gamma}_{idk} + \beta - 1}{\hat{\bar{N}}^{\Phi}_{wk} + \beta - 1} - \lambda_{\Phi} = 0$$

$$\sum_{id:w_i^{(d)}=w} \bar{\gamma}_{idk} + \beta - 1 = \lambda_{\Phi}(\hat{\bar{N}}^{\Phi}_{wk} + \beta - 1)$$

$$\hat{\bar{N}}^{\Phi}_{wk} = \frac{\sum_{id:w_i^{(d)}=w} \bar{\gamma}_{idk} + \beta - 1}{\lambda_{\Phi}} - (\beta - 1) \ . \tag{B.9}$$

Plugging Equation B.9 into the constraint which the Lagrangian enforces, we have

$$C = \sum_{wk} \hat{\bar{N}}^{\Phi}_{wk} = \frac{1}{\lambda_{\Phi}} \sum_{wk} \Big(\sum_{id:w_i^{(d)}=w} \bar{\gamma}_{idk} + \beta - 1 \Big) - KW(\beta - 1) \ .$$

Solving for the Lagrange multipliers, they turn out to be one:

$$\lambda_{\Phi} = \frac{\sum_{wk} \sum_{id:w_i^{(d)}=w} \bar{\gamma}_{idk} + KW(\beta - 1)}{C + KW(\beta - 1)} = \frac{C + KW(\beta - 1)}{C + KW(\beta - 1)} = 1 \ .$$

Plugging this back into Equation B.9 (and similarly for the other estimated EM statistics), we obtain M-step updates which synchronize the *estimated* EM statistics $\hat{\bar{\mathbf{N}}}$ with the current values of the *actual* EM statistics $\bar{\mathbf{N}}$, following definitions in Equation 4.58 (a.k.a. Equation B.6),

$$\hat{\bar{\mathbf{N}}}^{\boldsymbol{\Theta}} := \bar{\mathbf{N}}^{\boldsymbol{\Theta}} \qquad \hat{\bar{\mathbf{N}}}^{\boldsymbol{\Phi}} := \bar{\mathbf{N}}^{\boldsymbol{\Phi}} \qquad \hat{\bar{\mathbf{N}}}^{\mathbf{Z}} := \bar{\mathbf{N}}^{\mathbf{Z}} \ . \tag{B.10}$$

Note that after the M-step, $\sum_w \hat{\bar{N}}^{\Phi}_{wk} = \hat{\bar{N}}^{Z}_{k}$, $\forall k$, and we did not need to enforce this explicitly in the algorithm. The E-step finds the expected value of the complete-data log-likelihood, as encoded by the responsibilities $\bar{\gamma}_{id}$. Plugging in the estimates of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ from Equation B.7 into Equation B.4 gives us the update

$$\bar{\gamma}_{idk} :\propto \frac{\hat{\tilde{N}}^{\Phi}_{w_i^{(d)}k} + \beta - 1}{\hat{\tilde{N}}^{Z}_{k} + W(\beta - 1)}(\hat{\tilde{N}}^{\Theta}_{dk} + \alpha - 1) \ . \tag{B.11}$$

Alternatively, adding Lagrange terms $\sum_{id} \lambda_{id}(\sum_k \bar{\gamma}_{idk} - 1)$ to the bound to enforce the constraint that the $\bar{\gamma}$'s sum to one, setting the derivatives to zero then solving for $\bar{\gamma}_{id}$ also gives us Equation B.11.

We have shown that both updating the $\bar{\gamma}$'s (the E-step) and subsequently synchronizing the EM statistics with the $\bar{\gamma}$'s (the M-step) each optimizes the EM lower bound. The standard EM algorithm alternates between complete E and M-steps, i.e. updating all of the $\bar{\gamma}_{id}$'s, followed by synchronizing the EM statistics with the responsibilities. When the algorithm has converged, we can recover parameter estimates from the estimated EM statistics using Equation B.7.

However, the EM algorithm can be viewed as a coordinate ascent algorithm on the lower bound objective function, and partial E and M-steps also improve this bound (Neal & Hinton, 1998). In our case, both updating a single $\bar{\gamma}_{id}$, and subsequently synchronizing the EM statistics to reflect the new value (partial E and M-steps, respectively) are coordinate ascent updates which improve the EM lower bound in Equation B.8. So an algorithm that iteratively performs the update in Equation B.11 for each token (a partial E-step), while continuously keeping the EM statistics in synch with the $\bar{\gamma}_{id}$'s as in Equation B.6 (a partial M-step), is equivalent to the above EM algorithm but merely performing the coordinate ascent updates in a different order. This algorithm is very similar to CVB0, but using Equation B.11 (referred to as Equation 4.57 in the main body of this dissertation) instead of Equation 4.28. Such a strategy is likely to be more effective in practice than alternating full E-steps and M-steps, as it will propagate the results of the updates sooner, allowing up-to-date EM statistics to be used when updating each $\bar{\gamma}_{id}$.

Appendix C

Lyapunov Function for SCVB0

A Lyapunov function can be understood as an "objective function" which a stochastic algorithm would monotonically improve, if sufficiently small steps were taken and stochastic noise were absent. The existence of such a function is a standard argument for the stability and convergence of a stochastic algorithm. Theorem 2.3 of Andrieu *et al.* (2005) states that convergence is assured for a Robbins-Monro SA algorithm endowed with a Lyapunov function with certain properties, along with a boundedness condition and an appropriate sequence of step sizes. Andreiu *et al.* consider an SA with state space $\breve{\mathbf{\Theta}}$ for finding $h(\theta) = \mathbf{0}$, where $\breve{\mathbf{\Theta}}$ is an open subset of $\mathbb{R}^n$, and $h : \breve{\mathbf{\Theta}} \to \mathbb{R}^n$. They require the existence of a continuously differentiable function $w : \breve{\mathbf{\Theta}} \to [0, \infty)$, the Lyapunov function, where:

- *(i)* There exists $M_0 > 0$ such that
  $\mathcal{L} \triangleq \{\theta \in \breve{\mathbf{\Theta}}, \langle \nabla w(\theta), h(\theta) \rangle = 0\} \subset \{\theta \in \breve{\mathbf{\Theta}}, w(\theta) < M_0\}$,
- *(ii)* There exists $M_1 \in (M_0, \infty]$ such that $\{\theta \in \breve{\mathbf{\Theta}}, w(\theta) \leq M_1\}$ is a compact set,
- *(iii)* For any $\theta \in \breve{\mathbf{\Theta}} \setminus \mathcal{L}, \langle \nabla w(\theta), h(\theta) \rangle < 0$,
- *(iv)* $w(\mathcal{L})$ has an empty interior.

To establish convergence, a boundedness condition must also hold, namely that θ remains within a compact set $\mathcal{K} \subset \breve{\mathbf{\Theta}}$.

In our case, recall that in Section 4.6 we showed that the SCVB0 updates for each of the EM statistics c corresponds to a Robbins-Monro SA for finding the zeros of $f_c(X, \hat{s}^{(t)}) - \hat{s}_c^{(t)}$, i.e. the fixed points of MAP_LDA_U for $\hat{s}_c$. In the overall algorithm, $\theta = (\hat{\bar{N}}^{\Theta}_{(:)}, \hat{\bar{N}}^{\Phi}_{(:)}, \hat{\bar{N}}^{Z}_{(:)})^{\intercal} \in \mathbb{R}^n$, where we have concatenated the entries of the estimated EM statistics matrices so that θ is a vector of length n, and $h(\theta)$ is the direction of the M-step update that we would take if we were to first perform a full E-step. Finding $h(\theta) = \mathbf{0}$, as the SA algorithm is designed to do, corresponds to finding the fixed points of the MAP_LDA_U EM algorithm, which are at the stationary points of the posterior distribution of the parameters, i.e. the objective function for MAP estimation.

We will now show that the negative of the EM lower bound, augmented with Lagrange terms, is a Lyapunov function of the overall algorithm which satisfies the above conditions. As we found in Appendix B, if we include Lagrange constraints in the EM bound to ensure that the EM statistics sum to C, set the gradient to zero and solve for the Lagrange multipliers, the Lagrange multipliers turn out to equal one. Substituting this value into the Lagrangian and dropping constant terms, we have our candidate function

$$
\begin{aligned}
-w(\hat{\bar{\mathbf{N}}}^{\mathbf{\Theta}}, \hat{\bar{\mathbf{N}}}^{\mathbf{\Phi}}, \hat{\bar{\mathbf{N}}}^{\mathbf{Z}}) \triangleq \sum_{wk} \Big[(\sum_{id : w_i^{(d)} = w} & \bar{\gamma}_{idk} + \beta - 1) \log(\hat{\bar{N}}^{\Phi}_{wk} + \beta - 1) - \hat{\bar{N}}^{\Phi}_{wk} \Big] \\
& + \sum_{dk} \Big[(\sum_{i} \bar{\gamma}_{idk} + \alpha - 1) \log(\hat{\bar{N}}^{\Theta}_{dk} + \alpha - 1) - \hat{\bar{N}}^{\Theta}_{dk} \Big] \\
& - \sum_{k} \Big[(\sum_{id} \bar{\gamma}_{idk} + W(\beta - 1)) \log(\hat{\bar{N}}^{Z}_{k} + W(\beta - 1)) - \hat{\bar{N}}^{Z}_{k} \Big] \\
& - \sum_{idk} \bar{\gamma}_{idk} \log \bar{\gamma}_{idk} \ , \qquad \text{(C.1)}
\end{aligned}
$$

where $\bar{\gamma}$ are E-step estimates computed from the current EM statistics – note that $w(\theta)$ is not a function of them. We want to show that conditions *(i)* – through *(iv)*, along with the boundedness condition, hold for $w(\theta)$.

Regarding boundedness, we select the compact set $\mathcal{K}$ to be the non-negative hemisphere of the closed L_∞ ball with radius C, $\mathcal{K} = \{\theta \in \mathbb{R}^n | \forall j \theta_j \geq 0, \|\theta\|_\infty \leq C\}$. The state θ is always within the L_∞ ball because each of the EM statistics matrices are constrained to sum to C (or C_d), which is enforced because the updates take the form of convex combinations of matrices which satisfy the constraint. It is worth noting that when the hyper-parameters satisfy $\alpha - 1 > 0$ and $\beta - 1 > 0$ (which we assume as we are performing MAP estimation rather than maximum likelihood estimation), θ is furthermore always within the *interior* of the ball, because every $\bar{\gamma}_{idk}$ is non-zero, so every entry of the EM statistics count matrices is non-zero and less than C. Andreiu *et al.* require $\mathcal{K} \subset \breve{\mathbf{\Theta}}$ for some larger open set $\breve{\mathbf{\Theta}}$. We can choose $\breve{\mathbf{\Theta}}$ to be a slightly larger set of finite radius, say $\breve{\mathbf{\Theta}} = \{\theta \in \mathbb{R}^n | \forall j \theta_j \geq -\min(\alpha - 1, \beta - 1)/2, \|\theta\|_\infty < C + 1\}$, noting that the objective function is defined over slightly negative values greater than $-\min(\alpha - 1, \beta - 1)$ as the values inside the logarithm will be positive. The algorithm will never reach such values by the argument above, but we needed to show that the set $\mathcal{K}$ is strictly a subset of an *open* set $\breve{\mathbf{\Theta}}$ to satisfy the requirements of Andrieu *et al.* (2005).

We now consider the requirements for $w(\theta)$. Condition *(iv)* holds by Sard's theorem. The key conditions are *(i)* and *(iii)*, which involve the directional derivative of $w(\theta)$ at θ along $h(\theta)$, $\langle \nabla w(\theta), h(\theta) \rangle$. This is the instantaneous change in $w(\theta)$ in the direction of the EM update.[1] Note that a step with a step-size multiplier of one in the direction $h(\theta)$ is guaranteed by the monotonicity of EM to improve the (Lagrangian of the) lower bound, and thereby lower $w(\theta)$. However, for *(iii)*, we have to check that an infinitesimal step in that direction also improves this function.

[1]The directional derivative of w at θ along v is defined to be $\lim_{\lambda \to 0} \frac{w(\theta + \lambda v) - w(\theta)}{\lambda}$. If w is differentiable at θ, the directional derivative equals $\langle \nabla w(\theta), v \rangle$.

Suppose $\theta \in \breve{\mathbf{\Theta}} \setminus \mathcal{L}$. If $h(\theta) = \mathbf{0}$ this would contradict the assumption, so we are not at a fixed point of EM. Fixing $\bar{\gamma}$ to E-step-updated values based on θ, we know from the derivation of the M-step update, and from the concavity of the bound, that the Lagrangian of the EM lower bound has a unique maximum at the M-step updated value, located at $\theta + h(\theta)$. Since this maximum is unique and there are no other stationary points, each point in the direction $h(\theta)$ of the maximum has an increasingly large value of the Lagrangian of the EM bound, holding $\bar{\gamma}$ fixed. These values computed with $\bar{\gamma}$ fixed to its current value are a lower bound on the Lagrangian $-w(\theta)$ at those points: $w(\theta)$ is computed using E-step updated $\bar{\gamma}$'s which must strictly improve the EM lower bound relative to the current (or any other) $\bar{\gamma}$. More formally, let $w_{\bar{\gamma}_\theta}(\theta')$ be the negative of the Lagrangian at θ' with $\bar{\gamma}$ equal to the E-step updated values based on EM statistics θ. Then, making use of the concavity of the bound with $\bar{\gamma}$ fixed, for λ where $0 < \lambda \leq 1$,

$$\begin{aligned} -w\big(\theta\lambda + (\theta + h(\theta))(1-\lambda)\big) &> -w_{\bar{\gamma}_\theta}\big(\theta\lambda + (\theta + h(\theta))(1-\lambda)\big) \\ &\geq -w_{\bar{\gamma}_\theta}\big(\theta\big)\lambda + -w_{\bar{\gamma}_\theta}\big(\theta + h(\theta)\big)(1-\lambda) \\ &> -w_{\bar{\gamma}_\theta}\big(\theta\big) \\ &= -w\big(\theta\big)\ . \end{aligned} \tag{C.2}$$

So every point on the line segment between θ and $\theta + h(\theta)$ has a strictly higher value of the Lagrangian $-w(\theta)$ than at θ, i.e. $w(\theta + \lambda h(\theta)) - w(\theta) < 0$, $\forall \lambda \in (0, 1]$. Together with the assumption that $\theta \notin \mathcal{L}$, this implies that $\langle \nabla w(\theta), h(\theta) \rangle = \lim_{\lambda \to 0} \frac{w(\theta + \lambda h(\theta)) - w(\theta)}{\lambda} < 0$, and *(iii)* holds.

To show *(i)*, suppose $\theta \in \mathcal{L}$, i.e. the directional derivative $\langle \nabla w(\theta), h(\theta) \rangle = 0$. If θ is not a fixed point of the EM algorithm, then the directional derivative is negative by the above argument and we have a contradiction. So θ is a fixed point of MAP_LDA_U, and due to the properties of EM, is a stationary point of the MAP objective function. It follows that $w(\theta) < M_0$ for any $-M_0$ which is lower than the worst stationary point of the MAP. There

always exists such a strictly lower $-M_0$, as the Lagrangian can always be decreased by violating the constraints, e.g. multiplying the estimated EM statistics by a positive constant can arbitrarily decrease the bound, so we have that *(i)* holds. The set $\breve{\mathbf{\Theta}}$ has finite radius, so we can pick $M_1 > M_0$, where $\{\theta \in \breve{\mathbf{\Theta}}, w(\theta) \leq M_1\}$ is a compact set, and *(ii)* holds.

Having shown that the necessary conditions hold, Theorem 2.3 of Andrieu *et al.* (2005) now gives us that with an appropriate sequence of step sizes, in the limit as the number of iterations approaches infinity the distance from $\mathcal{L}$ is zero.

Appendix D

AIS-SG for Fast Learning in Undirected Graphical Models

It is difficult to perform maximum likelihood estimation for undirected graphical models such as restricted Boltzmann machines, due to an intractable sum in the gradient of the log-likelihood corresponding to sampling from the model's distribution. Consequently, a standard way to train such models is with contrastive divergence (CD) (Hinton, 2002), an approximate algorithm which works well in practice but whose convergence properties are not well understood. For example, it is known that the CD update does not correspond to the gradient of any function (Sutskever & Tieleman, 2010).

In this appendix, we propose a method for efficiently performing stochastic gradient ascent, using the iteration-AIS annealing path to compute ever-improving approximations to the gradient. The algorithm is similar to a variant of CD called persistent constrastive divergence (Tieleman, 2008), but has the advantage of importance weighting to correctly sample from the model distribution. It begins with the particle-filtered MCMC-MLE algorithm of

Asuncion *et al.* (2010), but extends it to use AIS annealing paths to improve the inference step of that algorithm.

Before describing the proposed method, we first provide some background on particle-filtered MCMC-MLE.

D.1 Particle-Filtered MCMC-MLE

Suppose we have a model in exponential family form,

$$Pr(\mathbf{x}|\theta) = \frac{\exp(\theta^\intercal \mathbf{x})}{Z(\theta)} \tag{D.1}$$

where $Z(\theta) = \sum_{\mathbf{x}'} \exp(\theta^\intercal \mathbf{x}')$ is the partition function, and $\mathbf{x}$ is a vector of sufficient statistics. Let us further assume that $Z(\theta)$ cannot easily be computed. For example, $Pr(\mathbf{x}|\theta)$ may be an undirected graphical model. We would like to be able to learn the parameters θ based on a data set $\mathbf{X}$ via maximum likelihood estimation. After normalizing by the number of data points N (a constant), we can write the log-likelihood as

$$\log Pr(\mathbf{X}|\theta) \propto \frac{1}{N} \sum_{i=1}^{N} \theta^\intercal \mathbf{x}^{(i)} - \log Z(\theta) \tag{D.2}$$

To maximize the log-likelihood, let us compute the gradient with respect to θ,

$$\frac{d \log Pr(\mathbf{X}|\theta)}{d\theta} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)} - \frac{1}{Z(\theta)} \Big(\sum_{\mathbf{x}'} \exp(\theta^\intercal \mathbf{x}') \mathbf{x}' \Big) \tag{D.3}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)} - \sum_{\mathbf{x}'} Pr(\mathbf{x}'|\theta) \mathbf{x}' \tag{D.4}$$

$$= E_{D(\mathbf{x})}[\mathbf{x}] - E_{Pr(\mathbf{x}|\theta)}[\mathbf{x}] \ , \tag{D.5}$$

where $D(\mathbf{x})$ is the data distribution, i.e. mixture of delta functions at each of the data points with uniform mixture weights. Interestingly, the gradient turns out to be the difference between the expectation of the data under the observed data distribution, and the expected value of what Hinton (2002) refers to as "fantasy data" under the model's distribution at θ. We cannot compute this directly as the right hand term is intractable. If we could simulate from $Pr(\mathbf{x}|\theta)$ we could perform a stochastic gradient update. However, this is still typically very difficult to do. MCMC algorithms can be used to sample from it, but they need to burn-in first, which is very expensive to perform for every gradient update. The *contrastive divergence* algorithm of Hinton (2002) avoids this problem by drawing from the data distribution $D(\mathbf{x})$ and taking just a *single* MCMC step (or perhaps a small number of steps). This will lead to an incorrect estimate of the gradient, but in practice the algorithm can work well as the direction of the approximate gradient is good enough to be useful. However, since the CD update is approximate, it does not exactly optimize its stated objective function (a difference between two KL-divergences), and in fact it is not the gradient of any function. We would prefer to have an algorithm with the efficiency of CD which optimizes the likelihood or another reasonable objective function.

Alternatively, the particle-filtered (PF) MCMC-MLE algorithm of Asuncion *et al.* (2010) achieves an estimate of the gradient by using *importance samples* of the model distribution, drawn via a particle filter. The algorithm defines a sequence of distributions from the model with parameters $\theta_1, \theta_2, \ldots$, where θ_j is the model at iteration j of the learning algorithm. It maintains a set of particles (samples) $\{\mathbf{x}^{(s)}\}$, which it updates by performing MCMC steps at each iteration j. These particles are used to find a Monte Carlo estimate of Equation D.5, with which a stochastic gradient update of θ is made. The algorithm iterates this procedure until convergence.[1]

[1]The full algorithm of Asuncion *et al.* also uses a *resampling* step, where the particles are resampled by drawing with replacement according to their current distribution (i.e. a mixture of delta functions at each sample, weighted by their importance weights). This step, as well as the MCMC updates ("*rejuvination*") are only performed if the effective sample size is small enough.

D.2 AIS-SG

We propose to modify PF MCMC-MLE to leverage annealed importance sampling. This is accomplished by changing the algorithm to use the iteration-AIS path to estimate the intractable expectations in Equation D.5. The key difference from PF MCMC-MLE is that AIS provides a sequence of *intermediate distributions* between the models at θ_{j-1} and θ_j, which give the algorithm a better chance of reaching θ_j. The intermediate distributions may also improve the importance weights, which are updated at each intermediate distribution.

To initialize the SG algorithm (referred to as *Annealed Importance Sampled Stochastic Gradient* (*AIS-SG*), the samples $\{\mathbf{x}^{(s)}\}$ are first drawn from some initial distribution θ_0, which is chosen to be normalized. In each iteration j, the samples are annealed towards the current model distribution θ_j using AIS with some path containing T_j temperatures (e.g. convex combinations of the parameters of the previous and current distributions). This gives importance weighted samples of $Pr(\mathbf{x}|\theta_j)$, which can be used to form a Monte Carlo estimate of the gradient in Equation D.5,

$$\Delta_{AIS-SG} = \frac{1}{B}\sum_b \mathbf{x}_b - \sum_{s=1}^{S} \omega_s \mathbf{x}^{(s)} \Big/ \sum_{s=1}^{S} \omega_s \ , \tag{D.6}$$

where $\{\mathbf{x}_b\}$ is a minibatch of data points drawn from the observed data distribution (which could potentially be a single data point, or the entire data set). Since the annealing path is an iteration-AIS path, the importance weights can be computed recursively by Equation 5.36. We can then take a stochastic gradient update,

$$\theta_j = \theta_{j-1} + \rho_j \Delta_{AIS-SG} \ . \tag{D.7}$$

The algorithm is detailed in Algorithm 9. Note that the models at each iteration are expected to be reasonably similar to each other, making them sensible candidates to anneal between.

With sufficiently many intermediate distributions, the importance samples are expected to improve as the algorithm proceeds.

Note the relationship that AIS-SG has to persistent constrastive divergence (Tieleman, 2008). It is essentially the same algorithm, but with importance weights. Persistent contrastive divergence was heuristically motivated – we can now justify that algorithm as approximately performing AIS-SG. Note that when AIS is run for long enough the variance of the importance weights decreases, in which case AIS-SG will become progressively more similar to persistent CD.

Also note that if the initial distribution is normalizable, we also obtain for free an estimate of the partition function at each iteration, since for AIS we have that $S^{-1}\sum_{s=1}^{S} w^{(s)}$ converges to the ratio of partition functions of the final and initial distributions (Equation 5.16):

$$Z(\theta_j) \approx \frac{\sum_{s=1}^{S} \omega_s}{S} \ . \tag{D.8}$$

D.3 Restricted Boltzmann Machines

Restricted Boltzmann machines (*RBMs*) are a key motivating example for the work presented in this appendix. For completeness, we describe these models here, as well as the standard contrastive divergence training procedure. An RBM (Smolensky, 1986; Hinton, 2002) is a Markov random field defining a probability distribution over binary vectors $V \in \{0,1\}^d$ and $H \in \{0,1\}^m$, where V are observed data and H are hidden (latent), with probability distribution

$$Pr(V,H) = \frac{1}{Z(\theta)} \exp(-E(V,H)) \ , \tag{D.9}$$

$$E(V,H) = -V^\intercal W H - a^\intercal V - b^\intercal H \ , \tag{D.10}$$

Algorithm 9 AIS-SG

$\theta_0 :=$ a normalized distribution which we can sample from

For each importance sample s

$\mathbf{x}^{(s)} \sim Pr(\mathbf{x}|\theta_0)$

$log\_\omega[s] = 0$

for each iteration $j = 1, 2, \ldots$

//Stochastically estimate the "positive gradient"

For each minibatch data point b

$\mathbf{x}_b \sim D(\mathbf{x})$

//Stochastically estimate the "negative gradient"

For each importance sample s

$(\mathbf{x}^{(s)}, \log\_\omega[s]) :=$ iteration-AIS($\mathbf{x}^{(s)}, \log\_\omega[s]$, θ_{j-1}, θ_j, T_j)

$\Delta_{AIS-SG} = \frac{1}{B}\sum_b \mathbf{x}_b - \sum_{s=1}^{S} \exp(log\_\omega[s])\mathbf{x}^{(s)} \Big/ \sum_{s=1}^{S} \exp(log\_\omega[s])$

$\theta_{j+1} = \theta_j + \rho_{j+1}\Delta_{AIS-SG}$

Output:

θ_j	An estimate of the MLE
$\text{logSumExp}(log\_\omega) - \log(S)$	An estimate of the log partition function $\log Z(\theta_j)$

and where the partition function $Z(\theta)$ is

$$Z(\theta) = \sum_V \sum_H \exp(-W(V, H)). \tag{D.11}$$

To train an RBM via maximum likelihood with gradient descent, from Equation D.5 the gradient updates are

$$\frac{dL}{da} = E_{D(V,H)}[V] - E_{Pr(V,H)}[V] \tag{D.12}$$

$$\frac{dL}{db} = E_{D(V,H)}[H] - E_{Pr(V,H)}[H] \tag{D.13}$$

$$\frac{dL}{dW} = E_{D(V,H)}[VH^\intercal] - E_{Pr(V,H)}[VH^\intercal] \tag{D.14}$$

where $D(V,H) = D(V)Pr(H|V)$ and $D(V)$ is the observed data distribution, i.e. a mixture of delta functions at each of the data points with uniform mixture weights. The left-hand expectations are simple to estimate via sampling, since $D(V)$ is trivial to sample from and the H's are conditionally independent given V. The right hand expectations, over $Pr(V,H)$, are difficult to approximate, leading to the use of alternative approximate methods such as the contrastive divergence (CD) algorithm (Hinton, 2002), which takes "gradient-like" steps

$$\Delta^a_{CD} = E_{D(V,H)}[V] - E_{R(V,H)}[V] \tag{D.15}$$

$$\Delta^b_{CD} = E_{D(V,H)}[H] - E_{R(V,H)}[H] \tag{D.16}$$

$$\Delta^W_{CD} = E_{D(V,H)}[VH^\intercal] - E_{R(V,H)}[VH^\intercal] \ , \tag{D.17}$$

where $R(V,H)$ is the distribution obtained by a single Gibbs update invariant to $Pr(V,H)$ on H and then V, starting from the observed V. In practice, the algorithm makes stochastic estimates of these updates, replacing the expectations with estimates based on one or more samples.