Abstract of "Scalable Bayesian Nonparametric Models for Networks and Documents" by Daeil Kim, Ph.D., Brown University, May 2017.

We develop Bayesian nonparametric statistical models of document collections and social networks. Extending classic parametric topic models of documents, and stochastic block models of networks, we first formulate flexible Bayesian nonparametric models based on the logistic stick-breaking process. This prior allows our model to automatically learn the dimension of the latent structure, use observed metadata to influence this structure, and discover correlations that exist between them. We call this model the Doubly Correlated Nonparametric Model (DCNM), and derive efficient MCMC learning algorithms.

We then focus on the problem of scaling inference to large networks. We propose a hierarchical Dirichlet Process (HDP) relational model and derive a structured variational inference algorithm. For the practically important case of communities with assortative structure, we derive new updates where inference scales linearly in time and memory with the number of active clusters. From this, we develop a stochastic variational approach that allows us to scale inference to networks that contain tens of thousands of nodes. Finally, we develop pruning techniques that allow us to dynamically shrink the number of communities, and effective strategies for specifying learning rate parameters.

After developing scalable inference models for relational data, we develop a memoized variational inference algorithm for the HDP topic model. This approach provides a more scalable framework for comparing models of varying complexity, by caching sufficient statistics of small batches of a very large dataset. Elegant delete-merge moves are then derived to optimize rigorous lower bounds on the marginal likelihood of the data, avoiding approximations required by previous stochastic inference algorithms. We use our memoized variational inference algorithms to develop Refinery, an open-source web platform for topic modeling that allows non-technical experts to leverage the power of topic models.

Scalable Bayesian Nonparametric Models for Networks and Documents

by
Daeil Kim
B. A., Sarah Lawrence College, 2004
Sc. M., Brown University, 2011

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2017

This dissertation by Daeil Kim is accepted in its present form by
the Department of Computer Science as satisfying the dissertation requirement
for the degree of Doctor of Philosophy.

Date ______________ ________________________________

Erik B. Sudderth, Director

Recommended to the Graduate Council

Date ______________ ________________________________

Eugene Charniak, Reader
Dept. of Computer Science, Brown University

Date ______________ ________________________________

David M. Blei, Reader
Dept. of Statistics, Columbia University

Approved by the Graduate Council

Date ______________ ________________________________

Andrew G. Campbell
Dean of the Graduate School

Vitae

Daeil Kim was born on May 14, 1982, in Seoul, South Korea and moved to New York City during the summer of 1984. He attended Stuyvesant High School and later Sarah Lawrence College, graduating with a B.A in Liberal Arts with a concentration in 19th Century Literature. Afterwards, he worked at a newly minted wine shop in Harlem and volunteered at a social psychology lab at Columbia University. This led to several years studying schizophrenia through fMRI analysis in Hartford Connecticut and Albuquerque New Mexico. In 2009, he applied and successfully entered Brown University's Department of Computer Science as a doctoral candidate along with an NSF graduate research fellowship. He earned his Master's degree from Brown in 2011 and his Ph.D. in 2016.

Publications related to this thesis

Doubly Correlated Nonparametric Models
Daeil Kim, Michael C. Hughes, and Erik B. Sudderth.
Journal of Machine Learning Research (JMLR), 2016. (In Submission)

Reliable and Scalable Variational Inference for the Hierarchical Dirichlet Process.
Michael C. Hughes, Daeil Kim, and Erik B. Sudderth.
Artificial Intelligence & Statistics (AISTATS), 2015.

Efficient Online Inference for Bayesian Nonparametric Relational Models
Daeil Kim, Prem Gopalan, David M. Blei, Erik B. Sudderth.
Neural Information Processing Systems (NIPS), 2013

The Nonparametric Metadata Dependent Relational Model.
Daeil Kim, Michael C. Hughes, and Erik B. Sudderth.
International Conference on Machine Learning (ICML), 2012.

The Doubly Correlated Nonparametric Topic Model
Daeil Kim, Erik B. Sudderth.
Neural Information Processing Systems (NIPS), 2011

Dedicated to my mother, Soon Sang

She had sacrificed everything to raise me the best way she knew. How difficult my life would have been otherwise. This is for her.

Acknowledgements

It is sometimes only time which allows us to appreciate the significance of another person's mind. As the years have gone by and my knowledge of machine learning deepened, it became clear how brilliant my advisor Erik was. It is really now I understand how difficult it must have been for him during my early years. A great mind can raise everyone else and he has passed down to me skills that are priceless and have become an endless source of intellectual fulfillment. For that I will always be thankful.

My time at Brown was wonderful. The community of researchers who I spent close to five years were the bulk of the reason for this. These include in no particular order, Ben Swanson who was my office mate and whose adventurous spirit livened up any dull moment. Soumya Ghosh who was the first colleague I met when we both decided to drive together cross-country and live together our first year. To this day it is always a joy to talk about machine learning with him. There is of course Michael Hughes whose skills I've admired and tried to emulate. If imitation is a form of flattery, then he certainly deserves a great deal of that. Jason Pacheco, my colleague who shared my passion for the humanities as well as machine learning. There was nothing we couldn't chat about and this had a great deal to do with his authenticity and kindness.

The Brown University computer science is a wonderful place to do research. It is truly a luxury to focus deeply on solving one very difficult problem. I'd like to thank Eugene Charniak for willing to be part of this committee and occasionally listening to my whacky ideas for NLP modeling. Much thanks goes to David Blei from Columbia University for the great conversations and insights he generously provided to me and anyone who asks. Working at the New York Times however has opened me up to a new community of applied researchers that have added richness to my life as a machine learning researcher. In particular, I need to thank Chris Wiggins who has been a great mentor the past two years on aspects both personal and career related.

Finally, this thesis is devoted to my family. My father would have been proud to see what I have accomplished given his unbelievably difficult and relatively short life. My mother was a martyr in this sense as well, working long endless nights to provide for us the best she can. I believe in this kind of happiness. To give back to those who have fought so hard for the priviliges we enjoy now.

Contents

List of Figures

List of Algorithms

Scalable Bayesian Nonparametric Models for Networks and Documents

Daeil Kim

September 14, 2016

Chapter 1

Introduction

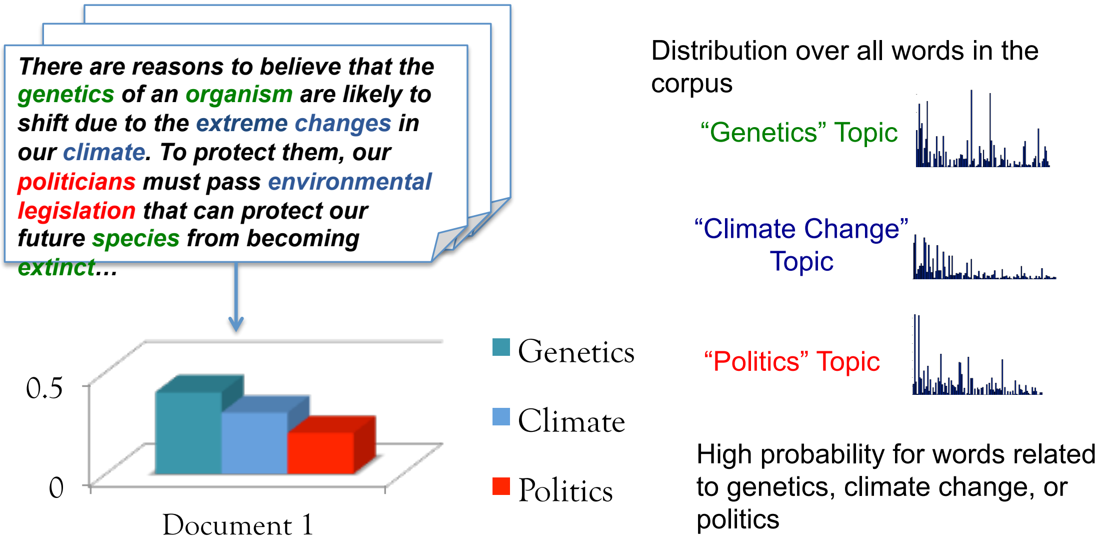
This thesis deals with the development of probabilistic models that scale with large datasets for two common types of data - documents and graphs. We focus on the problem of unsupervised learning where we need to learn a latent structure inherent in the data rather than map our input to a targeted set of labels or continuous values within the supervised learning framework. The unsupervised learning approach frees us to analyze a much richer set of data, but at the cost of relying primarily on our assumptions about the way data might be generated. The framework we will use to model these assumptions will rely on the language of graphical models [40, 47], which offers a powerful framework for concisely describing the relationships between random variables. A further benefit comes from the interpretability of such models, especially for systems related to medical field where these are necessary factors for actual use.

When modeling the latent structure associated with rich datasets such as documents or networks, we will be interested in developing a way to learn two distinct characteristics common to both datasets. The first will be metadata, where we focus on understanding how observed side information can be used to help better model our data. Such examples might be the year in which a document was made or the ages of individuals within a social network. By learning this relationship, we can then leverage the generative properties of our model to understand how a hypothetical document would look given a specific piece of metadata. This has useful implications in problems such as personalization, where we are faced with the "cold start" problem of recommending documents to a new user. The second will be correlations. By understanding how the learned latent structure is correlated, we can understand richer relationships between our latent structures which should help improve our ability to model our data.

A significant challenge in model design is determining the complexity of the model when fitting our data. Our approach will be to use Bayesian nonparametric models which provide a rich framework for tackling these types of problems [61], [73]. By using Bayesian nonparametric priors, we can provide support to infinite-dimensional parameter spaces. This allows us to develop inference procedures that allow the data to dictate our model complexity during training rather than an approach that utilizes an expensive cross-validation procedure. We show how this can be accomplished using Markov Chain Monte Carlo which allows us to develop a principled approach towards dynamically truncating our model via retrospective MCMC techniques.

Finally the last part of our thesis will focus on designing inference algorithms that can scale to large datasets. The rapid advancements in digital storage and archiving have now resulted in datasets that have rapidly outpaced the scale in which many of these inference procedures were developed for [51]. For example, the United States Library of Congress, the research library of the US Congress and de facto national library of America was estimated to have archived approximately 235 terabytes of data by April 2011. This might seem significant, but McKinsey and Company found in their research that 15 out of 17 sectors in the United States have more data stored per company than the US Library of Congress [51]. To accomplish this, we will apply variational methods for learning our model parameters. This is a deterministic optimization technique that uses the calculus of variations to directly optimize our model parameters, which has significant computational advantages over MCMC techniques. Ideas from the stochastic optimization literature will be borrowed to scale our training even further as well as clever memoization techniques that improve our model selection abilities for Bayesian nonparametric models.

1.1 Probabilistic Models for Large Document Collections



There are reasons to believe that the genetics of an organism are likely to shift due to the extreme changes in our climate. To protect them, our politicians must pass environmental legislation that can protect our future species from becoming extinct…

0.5
0
Document 1
Genetics
Climate
Politics
Distribution over all words in the corpus
“Genetics” Topic
“Climate Change” Topic
“Politics” Topic
High probability for words related to genetics, climate change, or politics

Documents are represented as mixtures of “topics” which in turn are distributions over a distinct vocabulary.

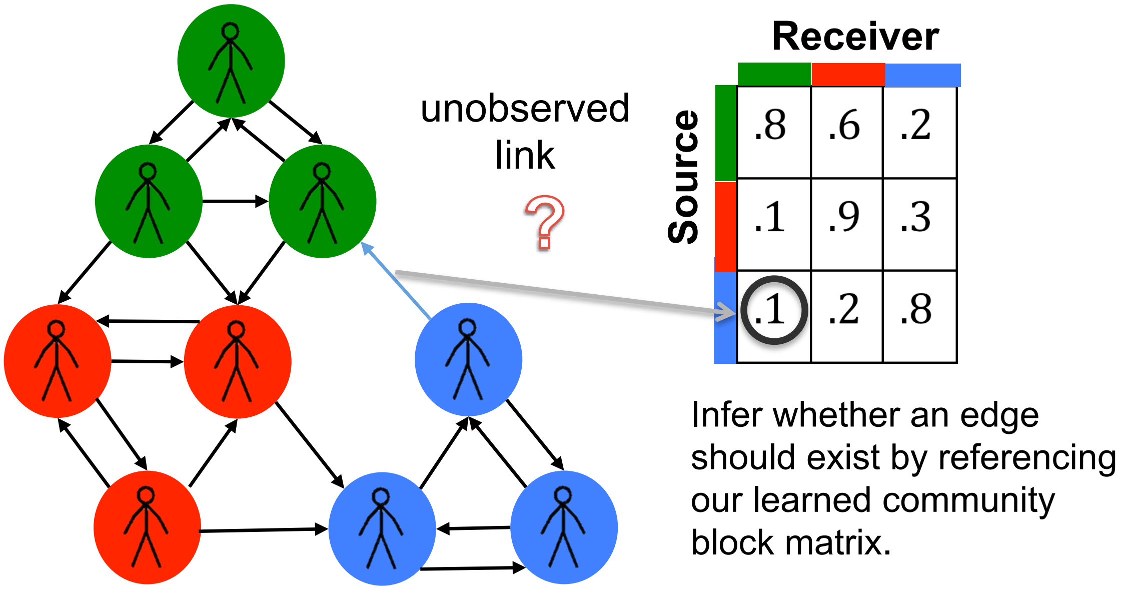
A major goal in the automated analysis of any document collection is an abstraction of its contents that captures the higher level semantic themes which tie the documents together. Significant advancements in this area came with the advent of topic models, the first known as Latent Dirichlet Allocation (LDA) [15]; a probabilistic generative model for documents. The "topic" in topic models refers to a discrete distribution over the unique words within a given document corpus. The popularity of LDA resulted from being the first admixture model for documents which modeled not only these global set of topics, but also defined a document-level latent structure that described the degree to which each of these topics played a role for any given document. Conditioned on these document-specific parameters, the generative process for words becomes order-agnostic or exchangeable. This conditional exchangeability represents the cornerstone of Bayesian modeling through DeFinetti's theorem. Though this assumption might be counter-intuitive to the structure of language, the discovered topics provide a rich semantic structure useful for tasks such as corpus exploration, information retrieval, and document prediction.

Since the introduction of LDA, significant research has expanded this basic topic model to allow for the dynamic learning of new topics [83], the incorporation of metadata to influence these topics [43, 53], as well as the discovery of correlations that might exist between topics [13, 43, 63]. Our work on the Doubly Correlated Nonparametric Topic model was the first to incorporate all three of these properties into a single probabilistic graphical model and the underlying framework to this model is highly similar to the DCNR model that we describe above. The major differences are found in how we define our likelihood terms which are specific to the generation of either words within a document or edges between pairs of nodes. In this proposal, we describe both these models under a unifying framework that we refer to as Doubly Correlated Nonparametric Models in Chapter 3.

1.2 Probabilistic Models for Large Relational Datasets

In the machine learning community, one of the most popular unsupervised learning approaches for network analysis deals with the discovery of node-specific community structures. The term community intuitively represents a clustering of nodes and within the statistics literature, these communities typically define their edge generation behavior. In other words, once their community assignments are known, we know everything there is to know about the nature of their interaction. The cardinality of this community structure, to be useful, is often significantly smaller than the number of distinct nodes within the graph. Thus, this discovery process attempts to provide an optimal compression of our network structure which is often followed by a human guided analysis to determine whether these communities refer to interesting real world phenomena.

Many statistical models for community discovery are based off a probabilistic generative model known as the stochastic block model [84]. This model defines a generative process for binary-valued adjacency matrices. It assumes that edges are exchangeable conditioned on their respective



	🟩	🟥	🟦
🟩	.8	.6	.2
🟥	.1	.9	.3
🟦	.1	.2	.8

Infer whether an edge should exist by referencing our learned community block matrix.

latent community assignments. Extensions to this model have allowed for the dynamic learning of the number of communities through Bayesian nonparametric priors [41], the incorporation of multiple memberships or features for a given node [2], and the ability to incorporate node-specific metadata [44]. Our work builds upon the stochastic block modeling literature to develop a single model that incorporates the properties above as well as the discovery of the correlations that exist between our latent communities. We call this the Doubly Correlated Nonparametric Relational (DCNR) model and show the applicability of this to a wide range of real-world networks to uncover rich underlying latent community structures. The components that we use to develop this comes partly from the ability to define a flexible Bayesian nonparametric prior known as the logistic stick-breaking process. Efficient MCMC inference procedures are then derived to estimate our posterior distribution over our latent statistics.

Other approaches to community discovery rely on well defined heuristics such as modularity scores [59] or entropy-minimization principles [70], but these approaches are not probabilistic nor generative. One of the benefits of probabilistic generative models over these techniques can be seen in their ability to deal with missing data as well as the interpretability of their learned structures. We provide a short discussion of these non-probabilistic approaches, but focus our efforts on describing and comparing to research that is most closely related to our work.

1.3 Scalable inference in probabilistic models

The general approach for approximate posterior inference in the machine learning community tends to be split between two distinct philosophies. The first approach, known as Markov Chain Monte Carlo [4, 56], tackles the problem of estimating a posterior distribution through a sampling based stochastic approach. The strategy is to obtain an approximate posterior distribution over our latent variables by empirically obtaining samples from a carefully designed Markov chain whose stationary distribution is the true posterior. This approach guarantees unbiased samples of our posterior, but the difficulty of assessing the convergence of this Markov chain as well as a generally debilitating computational cost makes this a less attractive option for inference in large scale datasets. Further approximation methods that speed up MCMC strategies exist, but at the cost of losing guarantees regarding the chain's convergence to the true posterior.

A more computationally scalable approach for training our model can be seen in variational inference, which poses the problem of posterior inference as an optimization problem [79]. Variational inference proceeds by defining a simpler distribution (known as the variational distribution) over our latent variables that are indexed by its set of variational parameters. The optimization of these parameters is then a search for members of this variational family of distributions that is closest in KL divergence to the true posterior. Variational inference is typically faster than MCMC approaches, but since the approximating family of distributions is simpler, it can be biased. The typical style of variational inference most often used for inference is known as a naive mean-field approach which assumes a fully factorized family of distributions and this is the variational family that we will employ for the rest of our proposal.

For scaling variational inference algorithms, a popular technique known as stochastic variational inference (SVI) [33] can be used. It was shown that if the members of our variational mean-field family of distributions are members of the exponential family, we can derive a simple stochastic natural gradient update for our global parameters. These gradients can be calculated from only a subset of the data and are noisy approximations of the true natural gradient for the variational objective, but represent an unbiased estimate of that gradient. Following directions on this noisy gradient with an optimization schedule that guarantees convergence (i.e Robbins-Monro criterion [68]), we obtain an approximation to the posterior that results in a highly scalable optimization technique. We develop these techniques for the HDP Relational Model and gain further computational savings when we make a reasonable assortativity assumption on the way communities interact.

One of the significant challenges of scalable variational techniques such as SVI is in developing a way to perform model selection during training. Choosing the number of clusters during training requires the model to typically have information across the entire dataset. However, SVI works by analyzing only a subset of the data before updating the global parameters of the model, resulting in a noisy representation of the lower bound that is typically needed to determine whether one model is better than another.

1.4 Thesis Organization

The outline of this thesis will be as follows. In the next chapter we will cover some background material necessary to understand the contributions we made. Chapter 3 is a discussion of the Doubly Correlated Nonparametric Model which can model both the correlations and the effect of metadata on our latent variables as well as a truly nonparametric inference procedure that can grow the number of clusters/communities during training. Chapter 4 will focus on the Hierarchical Dirichlet Process relational model that was developed along with a variational inference procedure that scales to large graphs. Chapter 5 will be a discussion of the memoized variational approach as applied to topic models. This technique caches the sufficient statistics of our latent variables to allow scalable learning of new clusters during training within the variational framework. Chapter 6 will focus on work involving Refinery, an open source web application that allows non machine learning experts to apply topic modeling to their own document corpuses. We conclude in summarizing our contributions and exciting future directions for this work.

Chapter 2: Background

This section begins by going over some commonly used Bayesian nonparametric priors in relational and topic models, notably the Dirichlet Process (DP) and the Hierarchical Dirichlet process (HDP). The limitations of these priors forms the motivation of our work in Chapter 3. The logistic and probit stick-breaking process is then described as a way of accounting for these limitations, but at the cost of conjugacy which results in a more difficult inference challenge. We review the work of the HDP as the Bayesian nonparametric prior for both relational and topic models to define a foundation for the work described later in our proposal. We then discuss inference and learning in these models, starting with Markov Chain Monte Carlo techniques and several variations of this stochastic approach to training our model. The final part of this background portion shifts to variational inference techniques which represent a deterministic optimization procedure that allows us to scale to larger datasets.

Chapter 3: Doubly Correlated Nonparametric Models

In this chapter the DCNM model and its generative process is discussed in detail. We show how the logistic stick-breaking process plays a prominent role in allowing for both the modeling of our observed metadata and the correlation structure between our clusters. For posterior inference, we use MCMC and through a careful selection of our priors and the marginalization of our our likelihood terms, we are able to dynamically grow/shrink the number of clusters during this inference procedure. Benefits of incorporating a richer latent structure is described in experiments over similar models.

Chapter 4: Scaling Inference in BNP Relational Models

We focus here on the development of the assortative Hierarchical Dirichlet Process model and its corresponding variational inference algorithm. The derivation for a stochastic variational inference approach is outlined and shown to be significantly better than the standard batch variational inference procedure. Along the way, we show how node-specific learning rates and initialization strategies can boost inference performance on several datasets. Finally, we showcase an analysis of the LittleSis network, a large social network containing tens of thousands of individuals who are the heights of business and government.

Chapter 5: Scaling Inference in BNP Topic Models

We focus our efforts on developing a memoized variational approach and apply it to the HDP Topic model. The memoized variational inference approach is motivated by the problem of growing/shrinking the number of topics in a topic model when analyzing small batches of data. Given that the full bound cannot be calculated during mini-batch training, memoized variational inference caches the sufficient statistics to be used later for model comparison. We show how this approach allows for a principled approach to learning the number of topics in a topic model and experiments that show significant improvements over competitor models.

Chapter 6: Refinery - Topic Modeling for the Masses

Our last contribution is focused on the development of a web platform that allows easy access to the topic models developed in the previous chapters. The web application allows for a simple drag and drop operation of text files that can be installed using two command lines. The user interface is optimized for simplicity and visualizations are developed to help understand the recovered topics.

Chapter 7: Conclusion and Future Directions

We conclude this thesis with directions for future research in scalable inference. Notably ideas around automated differentiation variational inference point to a future where very little time is spent deriving the inference equations for these models. Finally, having spent two years at the New York Times, a discussion around how machine learning can help drive investigative journalism concludes an extraordinary run of this doctoral journey.

Chapter 2

Background

This section outlines some of the fundamental concepts that the contributions of this thesis relies on. We begin by talking about the Dirichlet Process, a stochastic process prior that forms the basis for our Bayesian nonparametric models. We then discuss the Hierarchical Dirichlet Process

2.1 Dirichlet Processes

Bayesian nonparametric models are characterized by their use of stochastic processes as priors to provide support to infinite-dimensional parameter spaces. For Bayesian nonparametric mixtures models, the Dirichlet process [21] is often used since it generates random probability measures that integrate to one which are almost surely discrete [9, 71]. The original representation of the Dirichlet process defined a measurable space Ω, a positive concentration parameter α and a base measure H on Ω such that for all finite measurable partitions $(T_1, T_2, ..., T_K)$ of Ω:

$$(G(T_1), G(T_2), ..., G(T_K)) \sim \text{Dir}(\alpha H(T_1), \alpha H(T_2), ..., \alpha H(T_K)) \tag{2.1}$$

where G is a random probability measure on Ω. This is known as the Dirichlet process and is denoted as $G \sim \text{DP}(\alpha, H)$. An explicit representation of the Dirichlet process known as the stick-breaking construction was developed by [71] which allowed samples to be directly drawn from a Dirichlet Process. If $\beta_k \sim \text{Beta}(1, \alpha)$, we can define the same random measure G as:

$$\pi_k = \beta_k \prod_{\ell=1}^{k-1} (1 - \beta_\ell) \qquad\qquad G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k} \tag{2.2}$$

where ϕ_k are independent random variables distributed according to H and δ_{ϕ_k} is an atom at ϕ_k. Additionally, it was shown that with probability one, $\sum_k^\infty \pi_k = 1$ which allows for π to be interpreted as a random probability measure. The distribution over our stick-breaking weights π is typically denoted as $\pi \sim \text{GEM}(\alpha)$.

2.1.1 Hierarchical Dirichlet Processes

The natural nonparametric extension to LDA can be found in the hierarchical Dirichlet Process. Typically, a two-level hierarchical Dirichlet Process is formally defined as:

$$G \sim \mathrm{DP}(\gamma H) \qquad\qquad G_i \sim \mathrm{DP}(\alpha G), \text{ for each i} \tag{2.3}$$

where i could represent the index group for a particular document or node. It was shown by [77] that the almost surely discreteness of G allowed for the sharing of atoms at the individual cluster level indexed by i. Alternatively, if G was continuous or non-atomic, multiple draws from the DP would place their probability mass on a disjoint set of atoms s.t $G_1, G_2, ... \overset{iid}{\sim} \mathrm{DP}(\gamma G)$, [62]. Though many formulations of this prior exist, we focus on the explicit stick-breaking formulation of the HDP introduced by [77]. Let β now represent a set of global weights where β_k defines the expected frequency of membership in a possible community or topic k for a given node or document. The stick-breaking construction of β is defined in the same manner as before:

$$\beta_k = v_k \prod_{\ell=1}^{k-1}(1 - v_\ell), \qquad v_k \sim \beta(1, \gamma), \qquad k = 1, 2, ... \tag{2.4}$$

where $\gamma > 0$ is a now a positive concentration parameter that will control the global variance across our stick weights. The second or group-level DP can be defined as follows:

$$\pi_i \sim \mathrm{DP}(\alpha\beta), \qquad \mathbb{E}[\pi_i|\alpha, \beta] = \beta, \qquad \alpha > 0 \tag{2.5}$$

where β is the base measure defined above s.t $\beta \sim \mathrm{GEM}(\gamma)$. Here small values of α encourage documents or nodes to place most of their mass into a sparse subset of clusters.

2.1.2 Logistic and Probit Stick-breaking Process

The stick-breaking construction of the DP and HDP generate an infinite vector of membership probabilities π. However, this generative process is limited by the near-independence of any two topic indices π_j and π_k. If we wish to model correlations between clusters or incorporate side information to influence this partition structure, we need a more flexible prior. Instead of using many independent beta random variables $\beta_1, \beta_2, \ldots \beta_k \ldots$ to create the membership vector π, we can instead consider any other process for generating many values in the unit interval $(0, 1)$ which can then be transformed via stick-breaking into a probability vector π.

Our chosen process has two steps. First, we draw a set of real-valued membership weights $v = v_1, \ldots v_k \ldots v_K$ from a multivariate normal distribution:

$$v_1, v_2, \ldots v_k \ldots v_K \sim \mathrm{Normal}(m, \Sigma) \tag{2.6}$$

Here, we define a mean vector $m \in \mathbb{R}^K$ and a $K \times K$ positive-definite covariance matrix Σ. We can take the limit $K \to \infty$ to obtain an infinite-length activation vector v.

Given a specific set of activations v, we can transform each activation from a real value $v_k \in \mathbb{R}$ to the unit interval via a sigmoid-shaped squashing function $\psi : \mathbb{R} \rightarrow [0, 1]$. Two possible options for the squashing function ψ are the logistic and the probit:

$$\psi_{\text{logistic}}(v_k) = \frac{1}{1 + e^{-v_k}} \tag{2.7}$$

$$\psi_{\text{probit}}(v_k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{v_k} e^{-\frac{1}{2}x^2} dx \tag{2.8}$$

The logistic squashing function, whose form above is sometimes called the standard logistic function, arises in calculations of log-odds. The probit squashing function is the common name for the cumulative distribution function of the standard normal distribution. The required integral can be evaluated by standard black-box functions available in numerical software packages. Both logistic and probit functions satisfy the reflection property: $1 - \psi(v_k) = \psi(-v_k)$. Fig. 2.1 plots these functions side-by-side and compares their influences on the values of the membership vectors π.

Given any valid squashing function, we can complete our transformation of the real-valued activations v into a membership vector π via a stick-breaking transformation:

$$\pi_k \leftarrow \psi(v_k) \prod_{\ell=1}^{k} (1 - \psi(v_k)) \tag{2.9}$$

The logistic stick-breaking process has been studied previously by [67], while the similar probit stick-breaking process was introduced by [69]. Our novel contribution is to apply these processes to the topic modeling and relational modeling domains via an approach which accounts for both correlations and metadata.

Our DCNM model incorporates metadata and correlations by specifying appropriate values of the mean vector m and covariance matrix Σ. These values define the distribution of real activation weights v and thus membership probabilities π. The full details are presented in the next section, but the crucial idea is that available metadata determines the mean vector m while the correlations are captured by the covariance Σ. Both relationships are learned from data.

2.2 BNP Models for Documents and Networks

2.2.1 Hierarchical Dirichlet Process for Topic Models

Topic models in machine learning represent a class of statistical models for discovering the hidden "topics" within a document corpus. A topic is typically defined as a probability distribution over the unique vocabulary words across all documents. For example, an "animal" topic would place significantly more mass on words such as "dogs" and "horses" rather than "planes" and "boats". Furthermore, the model assumes that documents contain several topics, which are learned by the

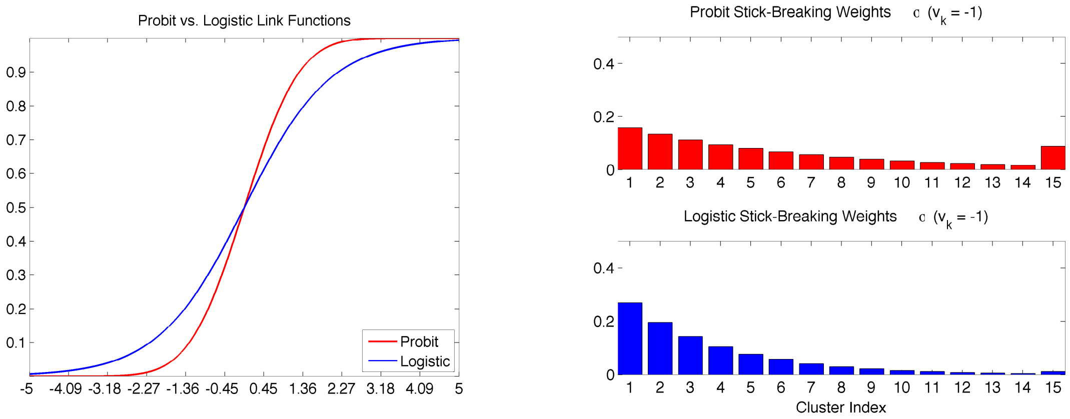


Figure 2.1: Left: Values of $v \in [-5, 5]$ squashed through the probit (red) and logistic (blue) functions. Note the heavier tail of the logistic function versus the probit. Right: Fixing $v_k = -1$ for all values of K to generate stick breaking weights where $\sum_{k=1}^{K=15} \pi_k = 1$, the resulting weights show that the logistic function will place significantly more mass for values that are further away from zero, resulting in a shorter tail versus those squashed by the probit function.

model. This results in the extraction of a rich latent structure that defines a set of global semantic themes that are allocated in various amounts for every given document.

Using the HDP representation described above, we define its generative process for the Hierarchical Dirichlet Process (HDP) Model first introduced by [77]. Let β be the set of global stick-breaking weights where $\beta_k \sim \text{GEM}(\gamma)$ as described above. We define our second level DP representing our document topic proportions for document i as $\pi_i \sim DP(\alpha\beta)$. Given a fixed vocabulary size, let $\Omega_k \sim \text{Dir}(\tau)$ be a finite Dirichlet distribution over our unique vocabulary words. To generate word w for a given document i, we first sample an indicator variable $z_{iw} \sim \text{Mult}(\pi_i)$ which defines the topic to which that particular word is drawn from. We then draw our observed word $y_{iw} \sim \text{Mult}(\Omega_{z_{iw}})$. This generative process assumes that documents are mixtures of global topic distributions. The inference challenge is to learn both these global topics and the proportion to which each topic exists for any given document. Latent Dirichlet Allocation [15], which is the parametric version of the HDP model uses a simpler representation for π. It draws its distribution over topics for a given document i as $\pi_i \sim \text{Dir}(\alpha)$, where α is a hyperparameter that controls the sparsity of topic weights. The major difference between the two models is the finite Dirichlet prior used for the document specific topic weights.

2.2.2 Hierarchical Dirichlet Process Relational Model

A similar approach to modeling documents can be taken with relational datasets, but where the latent topics or clusters model the behavior of nodes within a network rather than documents. Here

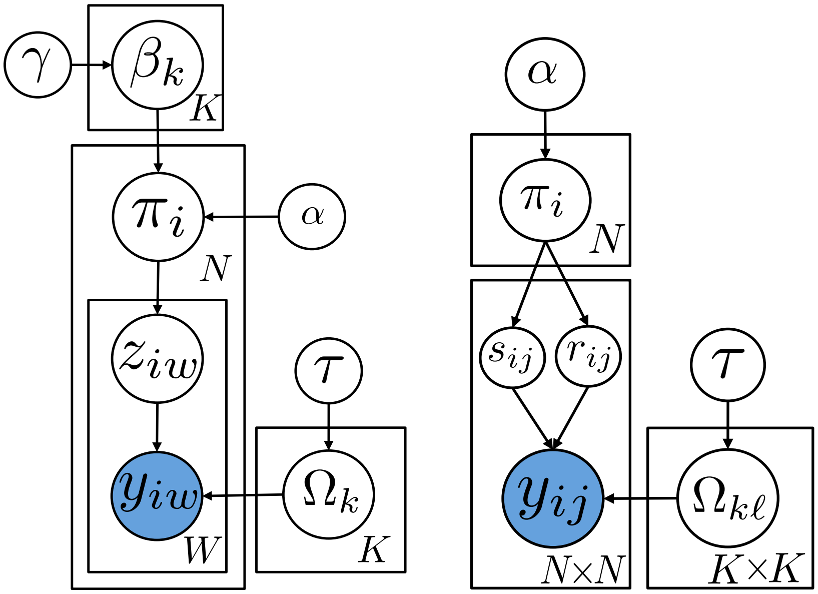


Figure 2.2: Left: Graphical model for Latent Dirichlet Allocation. Right: Graphical model for the Mixed-Membership Stochastic Block Model. We introduce an intersection within the graphical model between π_i, π_j to indicate a set of shared indices labeled from $1, .., N$.

we introduce the Hierarchical Dirichlet Process relational (HDPR) model and use the same stick-breaking formulation of the HDP model, but the major difference coming from the likelihoods which are built upon the class of stochastic block models for binary graphs [84]. We describe the inference procedure in significantly more detail in Chapter 4. For now, we briefly describe its generative process. Let Y be a binary adjacency matrix representing our graph with a single edge denoted as $y_{ij} \in \{0, 1\}$. Assuming N unique nodes indices in the graph, for each node $i \in N$ we first draw its global stick-breaking weights in the same manner as our HDP model by defining $\beta \sim \text{GEM}(\gamma)$. The mixed-membership community distribution for node i is then a DP s.t $\pi_i \sim \text{DP}(\alpha\beta)$. We then draw the community assignment for node i s.t $s_i \sim \text{Mult}(\pi_i)$ and the community assignment for the node j that is on the receiving end s.t $r_j \sim \text{Mult}(\pi_j)$. The relational model defines its likelihood by drawing a stochastic block matrix $\Omega_{k\ell} \sim \text{Beta}(\tau_a, \tau_b)$ to define the probability that community k will interact with community ℓ. Thus, to generate an edge y_{ij}, we need to sample two community assignments, s_i, r_j (source/receiver community indicators), which then denotes the correct entry in our stochastic block matrix s.t $y_{ij} \sim \text{Ber}(s_i^T \Omega r_j)$.

2.3 Markov Chain Monte Carlo for Inference and Learning

Markov Chain Monte Carlo (MCMC) techniques are one of the cornerstones for Bayesian inference and learning. The foundational ideas that pioneered these techniques started in the mid 20th century with the work of Stan Ulam to consider solving difficult combinatorial problems using simulation. Since then a significant number of algorithms have been developed that play upon the central ideas of simulation to approximate difficult integrals in problems as diverse as gambling or neutron diffusion. For the purposes of this thesis, our focus will be the application of MCMC for posterior inference.

MCMC for Bayesian inference will be applied in the context of parameter estimation for intractable posterior distributions. The strategy begins by assuming that the distribution of interest (i.e our posterior which we will call $p^*(x)$) cannot be sampled from directly, but can be evaluated up to a normalizing constant. If this is the case, we can start with an initial state x^0 and apply a transition operator that allows us to evolve the state of x to regions that spends most of its time in areas we care about, i.e our posterior. The term “Markov” comes from the fact that the transition operator we apply to evolve our initial state x is invariant to its previous evolution. In other words, the only necessary elements to evolve the state of x_t to x_{t+1} is its current state and the transition matrix as previous values of $x_{t-1}, x_{t-2}, ..., x_0$ are independent from what x_t will be. The elegance behind MCMC algorithms and their differences lie in the assumptions underlying the transition operator we apply to x, which ultimately leads to regions of our posterior which we wish to sample from.

All MCMC techniques must guarantee two fundamental properties when it comes to its transition operator. The first is irreducibility. For simplicity, assume that the space of x is discrete. Irreducibility states that for any state x, there is a positive probability that it can visit any other discrete state. The second property, aperiodicity assumes that once a state is visited, it does not become trapped in a cycle which visits the same set of states, which can also violate the first property of irreducibility. Assuming that our transition operator follows these principles we have a valid Markov chain and we are guaranteed that this chain will converge to a stable state, in which our case represents our posterior.

2.4 Variational Inference and Learning

Variational inference is a deterministic optimization procedure for Bayesian inference that leverages the calculus of variations to estimate our posterior. The introductory calculus courses taught in high school focuses on the problem of optimizing a function with respect to a point estimate, which is what we perform when we use techniques such as maximum likelihood estimation or iterative algorithms such as expectation maximization. However, when we wish to optimize a function or a distribution with respect to another function, then we’ll need the calculus of variations to accomplish this task.

When applied to problems of Bayesian inference, variational methods start by assuming a simpler set of distributions that approximate the true posterior. We then optimize our objective, which in this case would be the marginal log likelihood of our model, with respect to this distribution. To define this more concretely, assume that we have our data X and a set of latent variables Z along with a fixed set of hyperparameters α. We wish to learn the posterior over Z so to do this we introduce a variational distribution $q(Z)$ which is conveniently chosen to be simpler to optimize over. For example, a mean-field assumption is often placed over $q(Z)$ which assumes a fully factorized family of distributions. Furthermore, $q(Z)$ is often chosen to be a member of the exponential family of distributions, leading to updates that are analytically tractable. We can define the process of optimizing the marginal log likelihood within the variational framework as follows:

$$\log p(X \mid \alpha) = \log \int p(X, Z \mid \alpha)\, dZ = \log \int \frac{p(X, Z \mid \alpha)\, q(Z)}{q(Z)}\, dZ \tag{2.10}$$

$$= \log \mathbb{E}_{q(Z)} \left[\frac{p(X, Z \mid \alpha)}{q(Z)} \right] \tag{2.11}$$

Calculating the log of this expectation is intractable as we need to sum over the combinatorial explosion associated with our latent variables Z. So instead, variational inference proceeds by applying Jensen's inequality to this logarithmic convex function.

$$\log \mathbb{E}_{q(Z)} \left[\frac{p(X, Z \mid \alpha)}{q(Z)} \right] \geq \mathbb{E}_{q(Z)} \left[\log \frac{p(X, Z \mid \alpha)}{q(Z)} \right] = \mathcal{L}(q) \tag{2.12}$$

Jensen's inequality allows us to simplify the intractable computation into a simpler one by swapping the expectation with the convexity of our log operator. Doing this bounds the marginal log likelihood and we can then use the log operator within our expectation to break apart the terms associated with this new term $\mathcal{L}(q)$ often referred to as the ELBO or the evidence lower bound. The lower bound can also be reframed as the sum between our marginal log likelihood and then negative KL divergence between our true posterior and the variational distribution:

$$\mathcal{L}(q) = \log p(X \mid \alpha) - \mathrm{KL}\Bigg(q(Z) || p(Z \mid \alpha) \Bigg) \tag{2.13}$$

Such that when this KL divergence becomes zero, we recover the true marginal log likelihood. This framework is the basis for much of the literature in variational inference and research in this area is often focused on scaling these methods to larger datasets, more sophisticated assumptions over the variational family $q(Z)$, and analysis of where these families fail to capture aspects of the true posterior.

Chapter 3

Doubly Correlated Nonparametric Models

Hierarchical Bayesian modeling offers promising capabilities for automatically understanding the internal structure of modern datasets. One common application is modeling collections of text documents, such as many news articles or many academic publications. Here, the goal of modeling might be to discover common semantic themes across the corpus without expensive manual reading of each text. Another application is in modeling observed relationships between objects, which are represented via a directed graph. Here, observations could define relationships between related proteins or neurons or between friends in a social network.

Though the observed words in a document and the observed edges in a network might not seem to have much in common, both types of data can be explained via similar generative processes. The key shared characteristics are a common set of universal clusters or topics and a local mixed-membership representation of these shared topics. In the text domain, the popular topic model known as Latent Dirichlet Allocation or LDA [14] posits many latent clusters of semantically related words (such as “volcano, lava, ash” or “DNA, biology, genome”), and represents each document as a mixture of a sparse subset of the possible topics. A similar approach exists in relational modeling, where the popular mixed-membership stochastic block model or MMSB [2] assumes that many latent community clusters exist (such as “doctors” or “artists” in a social network). Each node belongs to a sparse subset of these communities and its edges are formed based on these memberships. The LDA and MMSB models and their close relatives have generated widespread interest due to the interpretable latent topics and communities they discover.

Our goal in this work is to present a new probabilistic model – the Doubly Correlated Nonparametric Model (DCNM) – for the latent mixed-membership structure needed for both topic and relational models. These contributions specifically fix some of the inherent limitations of existing models: the lack of correlations between latent topics and communities, the inability to use document-specific or node-specific metadata to inform the topic memberships, and the finite number of topics. We address each concern in detail below.

First, we consider correlations between topics. Explicitly modeling correlations is beneficial because we expect strong correlations in real-world data. For example, in news articles we might expect that political words occur much more frequently with economic words than words about paleontology. Similarly, in social networks we might expect that bankers and politicians interact much more than bankers and teachers. However, in standard Latent Dirichlet Allocation the Dirichlet prior over mixed-membership vectors is by definition as weakly-correlated as possible. One can exactly sample from a Dirichlet distribution by drawing a vector of independent, uncorrelated gamma random variables, and normalizing so they sum to one. This sum constraint induces slight correlations. Previous work that attempted to model correlations can be found in the correlated topic model or CTM [13] which utilized a logistic-normal prior to express correlations via a latent Gaussian distribution. However, its usage of a "soft-max" (multinomial logistic) transformation requires a global normalization, which in turn presumes a fixed, finite number of topics. Its relational counterpart, which was also a dynamic model used a similar soft-max transformation to model this prior over community memberships [32].

Second, both LDA and MMSB models are unable to incorporate side information about documents or nodes to help inform the learned membership values. Document corpuses contain rich metadata such as year of publication or associated authors. Similarly, social networks might include information regarding the age, gender, and education of an individual that can provide a more accurate representation her underlying community memberships. In the realm of topic models, the Dirichlet Multinomial Regression model [52] utilized a parametric Dirichlet prior whose topic weights were an exponentiated linear combination of observed feature values. The Gaussian process topic model [1] modeled correlations at the topic level via a topic covariance and incorporated metadata at the document level via an appropriate GP kernel function. This model remains parametric in its treatment of the number of topics, and computational scaling to large datasets is challenging since learning scales super-linearly with the number of documents.

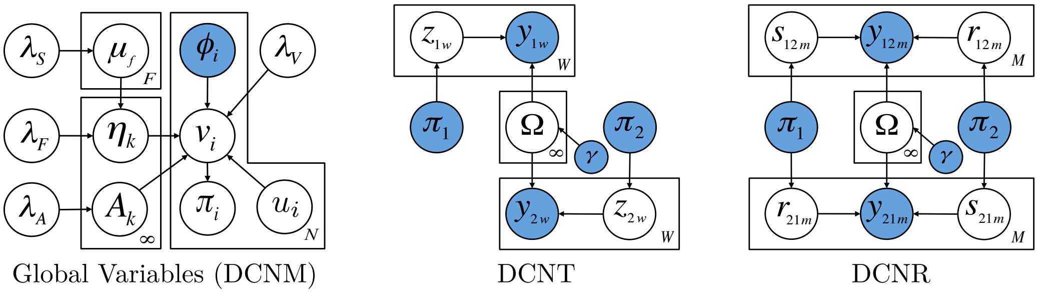
Finally, a third limitation of the LDA and MMSB models is the assumed fixed cardinality of the latent topic/community set. Both LDA and MMSB models require the practitioner to fix the exact number of represented clusters in the model before inference begins. Instead, the ideal model would not require such a restrictive parametric assumption and instead allow learning the number of clusters from data. The most direct nonparametric extension of LDA is the hierarchical Dirichlet process or HDP [76]. The HDP prior allows an unbounded set of topics and thus the posterior will induce a dataset-specific distribution over the preferred number of topics. However, like its finite counterpart LDA, the HDP cannot capture correlations or metadata. Alternatively, the nonparametric Bayes pachinko allocation model [49] captures correlations within an unbounded topic collection via an inferred, directed acyclic graph. More recently, the discrete infinite logistic normal model or DILN [62] captures topic correlations used an exponentiated Gaussian process (GP) to rescale the HDP's mixed-membership probabilities. This construction is based on the gamma process representation of the Dirchlet process [21]. While our goals are similar, we suggest a different approach to capturing correlations based on the stick-breaking representation of the DP [72]. This choice leads to arguably simpler learning algorithms and also facilitates our modeling of document metadata.

Our proposed DCNM mixed-membership model unites and extends the ideas from two previous conference papers: the text-data-specific Doubly Correlated Nonparametric Topic model (DCNT) [43] and the relational-data-specific Nonparametric Metadata Dependent Relational Model (NMDR) [44]. The important improvements we have made since those preliminary publications include the ability to model correlation structure for network data and the ability to learn the number of topics via retrospective MCMC. Additionally we improve the non-conjugate learning of mixed-membership probabilities by using an elliptical slice sampler [55], which we find to be more efficient that our previous Metropolis-Hastings proposals from the prior. We also offer a new reparameterization to improve non-identifiability issues when both correlations and metadata are modeled. We then evaluate our proposed approach on a range of datasets, including document collections with tens of thousands of articles and networks with a few hundred of nodes.

The next section introduces the entire graphical model and generative process for both topic-modeling and relational-modeling applications. We emphasize how the allocation of cluster labels within our model are identical for the purposes of document or relational datasets. Section 4 details the inference and learning for the model including details regarding our implementation of the elliptical slice sampler and retrospective MCMC techniques. Section 5 shares experimental results. Finally, Section 6 concludes with possible future directions and open questions.

3.1 Model

In this section, we provide a full mathematical description of our generative process for the DCNM. The DCNM incorporates both metadata and correlations to produce mixed-membership probabilities via a stick-breaking construction. This process is common to both topic modeling and relational modeling applications. Each application then has a specific downstream generative process for producing observed words or observed edges given fixed mixed-membership probabilities. These observation models are directly analogous to the corresponding portions of the LDA topic model or MMSB relational model. Graphical models for the common DCNM model as well as the observed-word observation model and the observed-edge observation models are in Fig. 3.1. Fig. 3.2 provides a practical illustration of how our model might use metadata and correlations in a topic-modeling application.



Global Variables (DCNM)

DCNT

DCNR

Figure 3.1: Left: The graphical model for the global variables is the same for either document or relational datasets. Correlations and metadata are also captured in this portion of the model. Center: Downstream from π, the graphical model for the DCNT is analogous to LDA. Here we show the plate notation for two documents with our global topic distribution parameterized by Ω. Right: A similar case applies to the DCNR model, where cluster indicators s, r now reference the source and receiver elements respectively within the stochastic block matrix Ω.

3.1.1 Metadata-informed mixed-membership probabilities

Most real world datasets often contain valuable metadata. For example, document collections almost always contain an author that might have written several other documents within that corpus. Information such as the data in which a document was written and its geographical location could be crucial in helping influence the discovery of our clustering structures. For networks, node metadata may come in the valuable form of personal details for individuals within a social network.

Let there be N total documents or nodes in the observed dataset, with each individual document or node indexed by $n \in 1, 2, \ldots N$. We assume that each document or node n is associated with an observed metadata vector $\phi_n \in \mathbb{R}^{F+1}$, representing F total attributes plus a constant bias term set to one: $\phi_n = [\phi_{n1}\ \phi_{n2} \ldots \phi_{nF}\ 1]$. Each scalar value ϕ_{nf} may be either continuous or binary. For example, the value of ϕ_{nf} might represent the age attribute of node n or the year of publication for document n. Categorically-valued metadata can be transformed to an appropriate one-hot binary vector. One limitation of our current metadata approach is that it does not handle missing features. We require all possible metadata values to be fully observed in all documents or nodes in the training set.

Given observed metadata ϕ_n for node n, our model generates the activation weights $v_n \in \mathbb{R}^K$ via the multivariate normal sampling from Eq. (2.6). The metadata values determine the mean vector of the activations via a linear transformation:

$$v_n \sim \text{Normal}(\eta^T \phi_n, \Sigma) \tag{3.1}$$

The weight parameter $\eta \in \mathbb{R}^{F+1 \times K}$ defines the linear projection of the metadata feature vector ϕ_n to the mean of the normal distribution. Each topic k is associated with a weight vector $\eta_{:k} \in \mathbb{R}^{F+1}$

for each attribute of the observed metadata, including the constant bias term. The integer k indexes a particular cluster or topic from a potentially unbounded set of these clusters. If no metadata is available, the model is simply:

$$v_n \sim \text{Normal}(\eta, \Sigma) \tag{3.2}$$

where $\eta \in \mathbb{R}^K$ is a learned mean parameter shared by all nodes or documents. This vector may be infinite in length if the model is nonparametric.

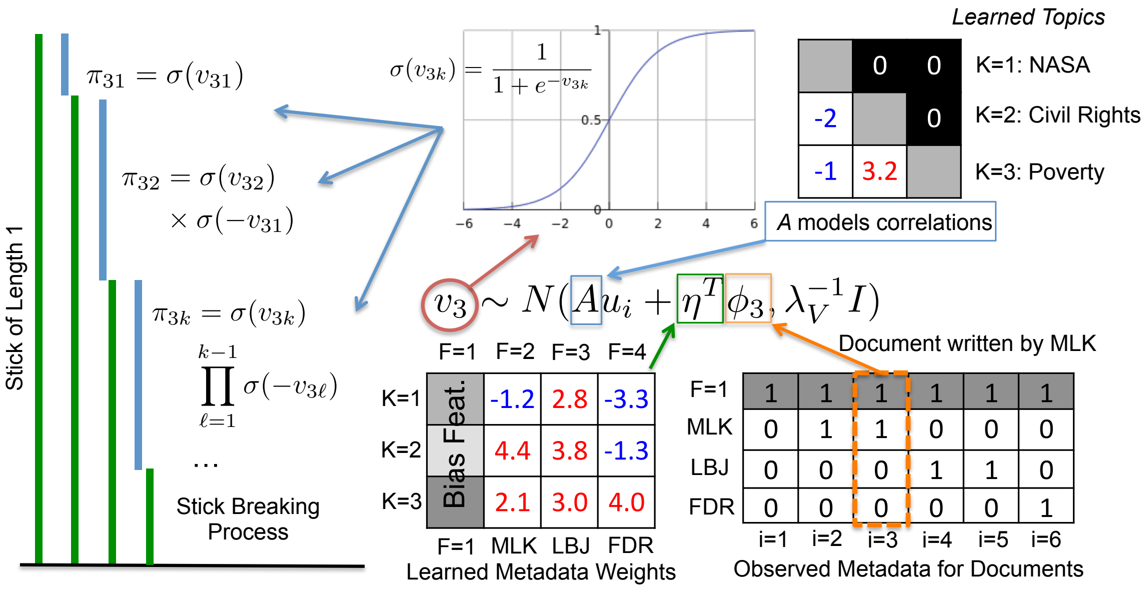


Figure 3.2: The figure above illustrates a hypothetical generative process for a corpus containing documents related to three topics, (NASA and Space, Civil Rights, and Poverty). Furthermore, assume we have documents written by famous individuals such as Martin Luther King Jr. (MLK), Lyndon B. Johnson (LBJ), and John F. Kennedy (JFK). Given such documents and metadata, our correlation matrix A might find positive correlations between *Civil Rights* and *Poverty* (i.e speeches or documents containing both themes), while negative correlations might exist between those topics and something like *NASA*. Consider the generative process now for v_3, a document written by MLK. Our learned metadata weights η show that documents written by MLK tend to have high weight for topics related to *Civil Rights* and *Poverty*, but negative weights for *NASA*. These metadata weights combined with our observed metadata for that document shifts the mean for v_3, while A adds further information about which particular topics to emphasize. We then follow the rest of the generative process and push v through a squashing function such as a logistic or probit link function to generate the individual weights for our stick-breaking process. Looking at π_{32} and π_{33}, we see that significantly more mass has been given to topics related to *Civil Rights* and *Poverty*, which is inline with what we expect for documents written by MLK.

Priors. As in standard Bayesian linear regression models, we place an independent Gaussian prior over the metadata weight value $\eta_{fk} \sim \mathcal{N}(\mu_f, \lambda_f^{-1})$ for each covariate $f \in 1, 2, \ldots F + 1$ and each topic $k \in 1, 2, \ldots$. Additional priors are placed over the global mean value of each metadata feature: $\mu_f \sim \mathcal{N}(0, \lambda_s^{-1})$ for each $f \in 1, 2, \ldots F + 1$. We can place a Gamma prior over this precision variable $\lambda_s \sim \text{Gam}(a, b)$.

3.1.2 Correlations in mixed-membership probabilities

To model correlations between topics, we specify a full-rank covariance matrix Σ for the Gaussian prior on the activation score vector v_n for document n. To handle the possibility of an unbounded number of topics k, we do not specify this covariance matrix directly. Instead, we parameterize this covariance using a square-root representation, where the parameter of interest is a lower-triangular matrix A of linear transformation weights. For each topic k, there are a set of k non-zero transformation weights indexed by ℓ: $\{A_{k\ell} : \ell = 1, \ldots, k\}$. Given the lower-triangular matrix A, we can sample the scalar activation score v_{kn} for topic k as:

$$v_{kn} \sim \text{Normal}\left(\eta_{:k}^T \phi_{:n} + \sum_{\ell=1}^{k} A_{k\ell} u_{\ell n}, \lambda_v^{-1}\right) \tag{3.3}$$

Here, the value $u_{\ell n} \sim \mathcal{N}(0, 1)$ is a Gaussian noise auxiliary variable and $\lambda_v > 0$ is a scalar precision parameter. We can compactly write the generative process for the entire vector v_n as:

$$v_{:n} \sim \text{Normal}(Au_{:n} + \eta^T \phi_{:n}, L), \qquad L = \text{diag}([\lambda_v^{-1} \ldots \lambda_v^{-1}]) \tag{3.4}$$

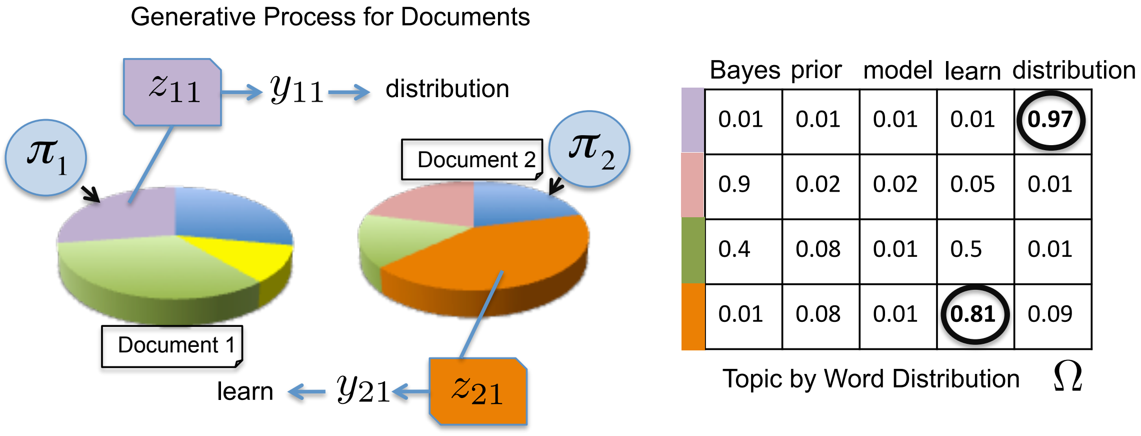
Here, the $K \times K$ matrix L is a diagonal covariance matrix, taking infinite dimensions as $K \to \infty$. The intuition underlying the use of a lower triangular matrix for A is similar to the idea of how correlations are captured within the square-root representation of an output covariance matrix. This can be seen more clearly when we marginalize out u, the covariance matrix for $v_{:n}$ is $\text{Cov}[v_{:n}] \triangleq \Sigma = AA^T + L$.

If no correlations are used but metadata is available, the model simplifies to a diagonal-covariance prior on the activation scores:

$$v_n \sim \mathcal{N}(\eta^T \phi_n, L). \tag{3.5}$$

Related work. Our integration of input metadata has similar connections to the semiparametric latent factor model [75], but we replace their kernel-based GP covariance representation with a feature-based regression. Furthermore, our metadata input is an offset to the mean for $v_{:n}$ which is different from the representation used in the original DCNT model [43] where A and η was coupled through u so that $v_{:n} \sim \mathcal{N}(Au_{:n}, L^{-1})$ and $u_{:n} \sim \mathcal{N}(\eta^T \phi_{:n}, \lambda_f^{-1} I_F)$. This change was motivated by the identifiability issues common in these types of models where a rotation of A could result in identical model likelihoods which can results in local optima issues.

Hyperpriors. The standard symmetric Gaussian prior over $A_{k\ell} \sim N(0, \lambda_A^{-1})$ in standard probabilistic factor analysis models poses a serious issue when it comes to Bayesian nonparametric models where k is potentially unbounded. Under this prior, $\mathbb{E}[\Sigma_{kk}] = k\lambda_A^{-1}$ grows linearly with k and though this may cause small artifacts in standard parametric models, the unbounded cardinality of our latent space enforces a need for an alternative prior where $A_{k\ell} \sim N(0, (k\lambda_A)^{-1})$. This plays the role of dampening the contribution of entries for A in the k^{th} row by reducing the variance for its entries by a factor of k. This shrinkage is carefully chosen so that $\mathbb{E}[\Sigma_{kk}] = \lambda_A^{-1}$ remains constant.



	Bayes	prior	model	learn	distribution
🟪	0.01	0.01	0.01	0.01	**0.97**
🟥	0.9	0.02	0.02	0.05	0.01
🟩	0.4	0.08	0.01	0.5	0.01
🟧	0.01	0.08	0.01	**0.81**	0.09

Figure 3.3: The figure above illustrates a very simplified generative process for two independent documents with fixed topic memberships π_1 and π_2 in a standard Bayesian topic model analogous to LDA. Topic indicator variables z_{11} and z_{21} represent the topic indicators that we will generate a word from s.t $z_{11} \sim \text{Cat}(\pi_1)$ and $z_{21} \sim \text{Cat}(\pi_2)$. In this simplistic example, the words distribution and learn are sampled from the purple and orange topics.

3.1.3 DCNT: Observation model for text data

For topic modeling applications, the DCNM generative process provides a mixed-membership probability vector π_n for each document n in the corpus. Given this value, we follow the LDA generative model to produce the W_n observed word tokens in document n: $y_{n1}, y_{n2}, \dots y_{nW_n}$. Each token indexed by w is generated in two steps. First, we sample an integer-valued topic assignment $z_{nw} \in \{1, 2, \dots K \dots\}$ from the document-specific membership vector:

$$z_{nw} \sim \text{Cat}(\pi_{n1}, \dots \pi_{nk} \dots) \tag{3.6}$$

Second, we draw the observed word $y_{nw} \in \{1, 2, \dots V\}$, which indicates which of V possible vocabulary types occurs at word token w.

$$y_{nw} | z_{nw} = k \sim \text{Cat}(\Omega_{k1} \dots \Omega_{kV}) \tag{3.7}$$

Here, the parameter Ω_k is a vector of length V that sums to one, indicating the probability of each vocabulary word under topic k. This generative process for documents is illustrated in Fig. 3.3. The left panel shows possible assignments z and word values y for two possible documents, and the right panel shows an example of topic-by-word parameter Ω. We emphasize that this observation model is directly similar to Latent Dirichlet Allocation [14].

3.1.4 DCNR: Observation model for relational data

For relational modeling applications, assume the observed data is a directed graph. That is, for any two nodes i and j, there are two distinct edges: the edge from i to j, denoted (i, j), and the edge from j to i, denoted by (j, i). We will assume that each edge has three possible states: either it is unobserved and thus not modeled, or it has an observed binary value of either 0 or 1.

The DCNM generative process provides a mixed-membership probability vector π_n for each node n in the network. Given these values, we follow the MMSB generative model to produce each observed directed edge (i, j) in two steps. First we draw assignments for the source node i and receiver node j independently from their respective node memberships:

$$s_{i,j} \sim \mathrm{Cat}(\pi_{i1}, \ldots \pi_{ik} \ldots), \qquad r_{i,j} \sim \mathrm{Cat}(\pi_{j1}, \ldots \pi_{jk} \ldots). \tag{3.8}$$

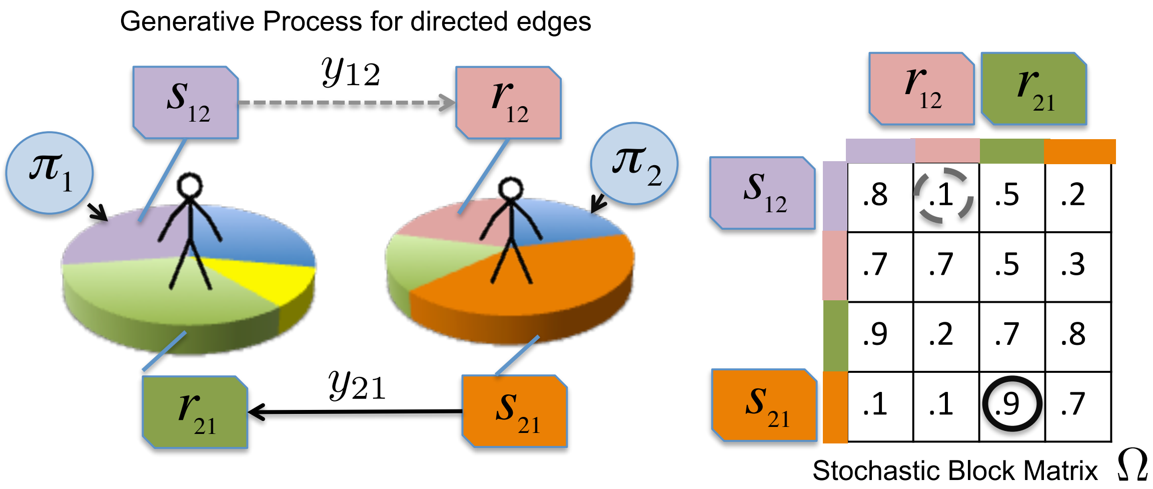
These integer assignments index one of the possible topics, so $s_{ij} \in \{1, 2, \ldots k \ldots\}$ and likewise $r_{ij} \in \{1, 2, \ldots k \ldots\}$. Given these assignments, we model the binary value of edge y_{ij} as a draw from a Bernoulli distribution:

$$y_{ij} \sim \mathrm{Bern}(\Omega_{k\ell}), \quad \text{if } s_{ij} = k, r_{ij} = \ell \tag{3.9}$$

Here, the parameter $\Omega_{k\ell} \in (0, 1)$ gives the probability of a directed edge from source community k to receiver community ℓ. We place a conjugate beta prior on each $\Omega_{k\ell}$ scalar parameter, usually setting $\Omega_{k\ell} \sim \mathrm{Beta}(0.1, 0.1)$.

This generative process for networks is illustrated in Fig. 3.4. The left panel shows possible assignments z and word values y for two possible documents, and the right panel shows an example of topic-by-word parameter Ω. We emphasize that this observation model is directly similar to the MMSB [2]. Our contribution is in the incorporation of metadata and correlations into the mixed-membership probabilities, which the MMSB itself cannot capture.

Multiple relations. It is straightforward to use the same node-specific membership probabilities to jointly model multiple relationships between the nodes. For example, in our later analysis of the social networks of lawyers, we have information about advice-seeking relationships and works-with relationships. In this scenario, we use common node-specific membership probabilities π_n for all N nodes. Then, for each relation $m \in 1, 2, M$, we draw distinct source and receiver assignments s_{ijm}, r_{ijm} for every edge (i, j), as well as distinct observed values y_{ijm}. See the appendix for details.



.8	.1	.5	.2
.7	.7	.5	.3
.9	.2	.7	.8
.1	.1	.9	.7

Figure 3.4: The figure above illustrates the generative process for two nodes with fixed mixed-memberships π_1 and π_2. To generate edges y_{12} and y_{21}, we first sample our community indicator variables s_{12}, r_{12} for y_{12} and s_{21}, r_{21} for y_{21}. These community indicator variables index the Bernoulli parameter from our stochastic block matrix.

3.1.5 Modeling Latent Correlations

The DCNT can model both positive and negative correlations among topic frequencies, but due to the nonlinearity associated with the logistic stick-breaking transformation, these covariances cannot be determined in closed form. We instead use a Monte Carlo estimate based on S samples from the covariance of each document, computed as follows:

$$\mathbb{E}[\pi_{:d}] = \frac{1}{S}\sum_{s=1}^{S} \pi^{s}_{:d} \tag{3.10}$$

$$\mathrm{Cov}[\pi_{:d}] = \frac{1}{S}\sum_{s=1}^{S} (\pi^{s}_{:d} - \mathbb{E}[\pi_{:d}])(\pi^{s}_{:d} - \mathbb{E}[\pi_{:d}])^{T} \tag{3.11}$$

$$\hat{\Sigma} = \frac{1}{D}\sum_{d=1}^{D} \mathrm{Cov}(\pi_{:d}) \tag{3.12}$$

Here, $\pi^{s}_{:d}$ is computed by mapping a single sample of $v_{:d}$, conditioned on the learned model parameters, through the logistic stick breaking transformation. For our visualizations, we set $S = 5000$ for each document d. We used a similar Monte Carlo estimator for the LDA model, conditioned on its Dirichlet topic weights α.

3.2 Monte Carlo Learning and Inference

After developing a full probabilistic model that uses topic correlations and metadata to inform mixed-membership values, we now develop algorithms for training this model from observed data. We develop a Markov chain Monte Carlo (MCMC) algorithm which in each iteration performs a sweep over all hidden variables, updating each one while holding the others fixed. When possible, updates occur in blocked fashion rather than one scalar at a time. Each update is designed so that after many iterations, the joint state of all variables can be considered a sample from the true posterior.

While there many latent variables in our model which need to be sampled, most of these have been specifically given conjugate prior densities so they have closed-form posterior updates and thus admit well-known Gibbs sampling updates. Variables A, u, Ω, μ and all precision parameters all admit such Gibbs updates.

3.2.1 Metropolis Hastings for Topic/Community weights v

The posterior distribution of $v_{:i}$ does not have a closed analytical form due to the logistic nonlinearity underlying our stick-breaking construction. In order to obtain samples for v, one approach is to apply a standard Metropolis-Hasting algorithm. The Metropolis-Hastings algorithm is a MCMC technique for obtaining samples from almost any arbitrary distribution. The basic idea is to define an acceptance ratio $\mathcal{A}$ to determine whether we accept or reject a particular proposal for v which we define as

$$\mathbb{A}(v^*, v) = \frac{p(y, v^*, \theta)q(v \mid v^*)}{p(y, v, \theta)q(v^* \mid v)} \tag{3.13}$$

where θ refers to the current setting of our latent parameters and v^* is a set of new parameters that we sample from some proposal distribution $q(v)$. We then draw a uniform random variable $\zeta \sim \text{Unif}(0, 1)$ and accept if $\mathbb{A} > \zeta$. Here $p(y, v^*, \theta)$ is the joint distribution of our model given a proposal for v and $q(v^*|v)$ is the transition probability of going from state v to v^*. If $q(v|v^*) = q(v)$ and $q(v^*|v) = q(v^*)$, then we assume a variant of this algorithm known as the Metropolis-Hastings Independence sampler which we use for the DCNM.

To implement this for the DCNT, we need to define a proposal distribution $q(v_{:i})$ for the topic activation weights for document i. For simplicity, this proposal is set to its prior. For our acceptance ratio, due to the conditional independencies in our graphical model, most terms not related to v cancel out and we are left with the ratio of our likelihood terms for z_i. The proposal is accepted with probability $\min(\mathbb{A}(v^*_{:i}, v_{:i}), 1)$, where $q(v^*_{:i} \mid v_{:i}, A, u_{:i}, \eta, \phi_{:i}\lambda_v) = N(v^*_{:i} \mid Au_{:i} + \eta^T \phi_{:i}, \lambda_v^{-1} I_K)$:

$$\begin{aligned}\mathbb{A}(v^*_{:i}, v_{:i}) &= \frac{p(v^*_{:i} \mid A, u_{:i}, \lambda_v) \prod_{w=1}^{W_i} p(z_{iw} \mid v^*_{:i}) q(v_{:i} \mid v^*_{:i}, A, u_{:i}, \eta, \phi_{:i}, \lambda_v)}{p(v_{:i} \mid A, u_{:i}, \eta, \phi_{:i}, \lambda_v) \prod_{w=1}^{W_i} p(z_{iw} \mid v_{:i}) q(v^*_{:i} \mid v_{:i}, A, u_{:i}, \eta, \phi_{:i}, \lambda_v)} \\ &= \prod_{w=1}^{W_i} \frac{p(z_{iw} \mid v^*_{:i})}{p(z_{iw} \mid v_{:i})} = \prod_{k=1}^{K} \left(\frac{\pi^*_{ki}}{\pi_{ki}} \right)^{\sum_{w=1}^{W_i} \delta(z_{iw}, k)} \qquad (3.14)\end{aligned}$$

where W_i are the number of words in document i. Because the proposal cancels with the prior distribution in the acceptance ratio $\mathbb{A}(v^*_{:i}, v_{:i})$, the final probability depends only on a ratio of likelihood functions, which can be easily evaluated from counts of the number of words assigned to each topic by z_i. When we are dealing with the DCNR model, the only changes to the acceptance ratio comes in the form of the way the counts are estimated in a relational model

$$\mathbb{A}(v^*_{:i}, v_{:i}) = \prod_{j=1}^{N} \frac{p(s_{ij} \mid v^*_{:i}) p(r_{ji} \mid v^*_{:i})}{p(s_{ij} \mid v_{:i}) p(r_{ji} \mid v_{:i})} = \prod_{k=1}^{K} \left(\frac{\pi^*_{ki}}{\pi_{ki}} \right)^{\sum_{j=1}^{N} \delta(s_{ij}, k) + \delta(r_{ji}, k)} \qquad (3.15)$$

3.2.2 Elliptical Slice Sampling

The logistic mapping for $v_{:i}$ to our stick breaking weights $\pi_{:i}$ results in a posterior distribution for $v_{:i}$ that does not have a closed analytical form. Previous work on both the DCNT and NMDR dealt with this non-conjugate prior by employing the Metropolis-hastings independence sampler as described before, where proposals $q(v_{:i})$ are drawn from the prior. This results in simple updates for v, but can mix poorly due to high rates of rejection, resulting in experiments that require longer MCMC chains before empirical convergence can be assessed. Recent work by Murray et. al introduced an elliptical slice sampling technique for latent variables with Gaussian priors that has been shown to be an effective alternative to Metropolis-Hastings techniques resulting in a sampler for v that always accepts as well as empirically mixing more quickly than our previous independence sampler.

The elliptical slice sampler for the DCNM is a variant of a standard slice sampler that maps the range of possible proposals for $v_{:i}$ to lie on an ellipse. The application of slice sampling arises in the context of picking a suitable bracket of possible proposals from this ellipse and adjusting its size after each rejection until acceptance. The algorithm will always accept a value for $v_{:i}$ unless the $v_{:i}$ that we started with is the only valid value for our likelihood. This provides significant benefits over Metropolis-Hastings which can have high rates of rejection depending on a poor choice for the proposal distribution.

The algorithm proceeds as follows. Let $\mu = \eta^T \phi_i$ and $f = v_{:i} - \mu$. To define our ellipse we draw a value $\tau \sim N(\mu, (AA^T + \lambda_v^{-1} I)^{-1})$ centered around μ. Note that this distribution is the prior for $v_{:i}$ with $u_{:i}$ marginalized out, resulting in dependencies between entries of $v_{:i}$. The slice is defined by thresholding our log likelihood function $L(f)$ which will be different depending on whether we are modeling documents or networks. Here $f = v_{:i} - \eta^t \phi_{:i}$ (the elliptical slice sampler requires the prior to have zero mean) and our slice can be defined as $\log x = \log L(f + \mu) + \log \zeta$ where $\zeta \sim \text{Unif}(0, 1)$. Now that the ellipse and the slice on the ellipse is specified, the algorithm now defines a bracket to search in by drawing $\theta \sim \text{Unif}(0, 2\pi)$ and setting the bracket width to be $[\theta_{min}, \theta_{max}] \leftarrow [\theta_{min} - 2\pi, \theta]$. We then represent our new model as $f^* \leftarrow f \cos\theta + \tau \sin\theta$ and accept $v^*_{:n} = f^* + \mu$ if and only if $\log L(f^* + \mu) > \log x$. Otherwise, we shrink our bracket by setting $\theta_{min} = \theta$ if $\theta < 0$ or $\theta_{max} = \theta$ is positive and then try again until a value for $v_{:i}$ is determined that lies on the proposed slice.

As in previous applications of similar Chib-style estimators, we set the length of the transition chain to be $S = 1000$, and run $T = 1000$ iterations to determine a high posterior probability state. Due to the use of our ESS sampler, we no longer need to reweight the final predictive likelihood as done originally in [43].

Algorithm 3.1 Elliptical Slice Sampler for $v_{:n}$ with change of variables

```
Input: Let $\mu = \eta^T \phi_{:n}$ and $f = v_{:n} - \mu$. Log likelihood function $L(f)$
Output: $v^*_{:n} = f^* + \mu$
1:  Draw $\tau \sim N(\mu, (AA^T + \lambda_V^{-1} I)^{-1})$
2:  Draw $u \sim \text{Uniform}[0, 1]$
3:  Define Slice $\log x \leftarrow \log L(f + \mu) + \log u$
4:  Draw initial proposal and bracket:
    $\theta \sim \text{Uniform}[0, 2\pi]$
    $[\theta_{min}, \theta_{max}] \leftarrow [\theta_{min} - 2\pi, \theta]$
5:  $f^* \leftarrow f \cos\theta + \tau \sin\theta$
6:  if $\log L(f^* + \mu) > \log x$ then
7:      Exit and keep $v^*_{:n} = f^* + \mu$
8:  else
9:      Shrink bracket and try again:
10:     if $\theta < 0$ then $\theta_{min} = \theta$ else $\theta_{max} = \theta$
11:     $\theta \sim \text{Uniform}[\theta_{min}, \theta_{max}]$
12:     Goto 8
```

3.2.3 MCMC for SCNM-M

For this particular model, our corresponding inference updates are modified s.t our correlation loading matrix A and its corresponding precision variable λ_a are no longer sampled as well as u, our document/node specific noise variable. For η and λ_{vk}, we have slightly modified posteriors which appropriately exclude the contribution of A and u. As before, columns of $\eta_{:k}$ are conditionally independent, with Gaussian posteriors:

$$\begin{aligned} p(\eta_{:k} \mid \phi, \mu, \lambda_f) &\propto N(\eta_{:k} \mid \mu, \lambda_f^{-1} I_F) N(v_{k:}^T \mid \phi^T \eta_{:k}, \lambda_v^{-1} I_N) \\ &\propto N\Big(\eta_{:k} \mid (\lambda_f I_F + \lambda_v \phi\phi^T)^{-1}(\lambda_v \phi v_{k:}^T + \lambda_f \mu),\ (\lambda_f I_F + \lambda_v \phi\phi^T)^{-1}\Big) \end{aligned} \tag{3.16}$$

Also, updates for λ_{v_k} now have the following posteriors, which will also be another Gamma distribution:

$$
\begin{aligned}
p(\lambda_{v_k} \mid \eta_{:k}, \phi, a_v, b_v) &\propto \mathrm{Gam}(\lambda_{v_k} \mid a_v, b_v) \prod_{i=1}^{N} N(v_{ki} \mid \eta_{:k}^T \phi_{:i}, \lambda_{v_k}^{-1}) \\
&\propto \mathrm{Gam}\Bigg(\lambda_{v_k} \mid \frac{N}{2} + a_v,\ \frac{1}{2}\sum_{i=1}^{N}(v_{ki} - \eta_{:k}^T \phi_{:i})^2 + b_v\Bigg)
\end{aligned}
\tag{3.17}
$$

3.2.4 Retrospective MCMC

Bayesian nonparametric models based on unconventional stick-breaking priors often use a finite, truncated approximation to the true infinite model. While this approach can be effective, selection of an appropriate truncation level K is challenging. When K is conservatively large, substantial computational resources can be expended resampling "wasted" variables, and model interpretability often suffers. When K is small, learning and inference are potentially biased, and the benefits which originally motivated the nonparametric approach are lost. To avoid these issues, we implement a dynamic truncation technique based on retrospective sampling [64] of our latent community assignments.

DCNR Retrospective MCMC

Consider the resampling of a source indicator $s_{ij} \sim \mathrm{Cat}(\rho)$ from node i to node j, given fixed values of all other indicators and variables represented by our posterior ρ. A similar approach can be used for resampling r_{ij}, or for blocked resampling of $\{s_{ij}, r_{ij}\}$. Because we employ a conjugate beta prior, our sampler analytically marginalizes the relation parameters $\Omega_{k\ell}$, expressing part of the posterior ρ in terms of various edge counts. Suppose that $r_{ij} = \ell$. Excluding node pair (i, j), let $C_{k\ell}^{\setminus ij}$ denote the number of present directed edges from nodes whose indicators associate that pairing to communities (k, ℓ). Similarly, let $D_{k\ell}^{\setminus ij}$ denote the number of absent edges with the same community indicators.

Let K denote the index of the largest community, in stick-breaking order, which currently has at least one assigned node. The retrospective sampler explicitly instantiates $v_{k:}$ and $\eta_{:k}$ for $k \leq K$. Computing π_{ki} based on these variables, as in Eq. (2.9), we let

$$
\rho_k \propto \pi_{ki} \left(\frac{(C_{k\ell}^{\setminus ij} + \gamma_a)^{y_{ij}} (D_{k\ell}^{\setminus ij} + \gamma_b)^{1-y_{ijm}}}{C_{k\ell}^{\setminus ij} + D_{k\ell}^{\setminus ij} + \gamma_a + \gamma_b} \right) \qquad \text{for } \ k = 1, \ldots, K \tag{3.18}
$$

$$
\rho_{K+1} \propto \left(1 - \sum_{k=1}^{K} \pi_{ki}\right) \left(\frac{\gamma_a^{y_{ij}} \gamma_b^{1-y_{ij}}}{\gamma_a + \gamma_b} \right). \tag{3.19}
$$

The proportionality constant in Eqs. (3.18, 3.19) is selected so that ρ is a properly normalized $(K+1)$-dimensional multinomial distribution. For $k \leq K$, ρ_k is the posterior probability of selecting community k. ρ_{K+1} is the aggregate posterior probability of the infinite "tail" of communities with indexes greater than K.

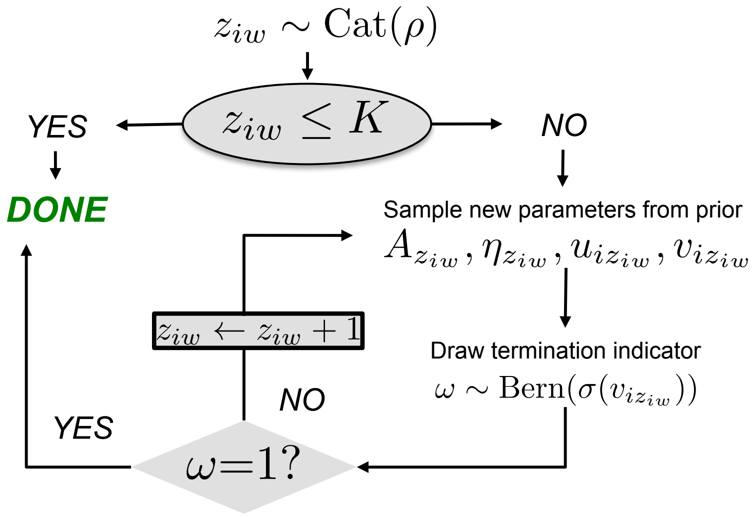

$z_{iw} \sim \text{Cat}(\rho)$
$z_{iw} \leq K$
YES
NO
DONE
Sample new parameters from prior
$A_{z_{iw}}, \eta_{z_{iw}}, u_{iz_{iw}}, v_{iz_{iw}}$
$z_{iw} \leftarrow z_{iw} + 1$
Draw termination indicator
$\omega \sim \text{Bern}(\sigma(v_{iz_{iw}}))$
NO
YES
ω=1?

Figure 3.5: The retrospective MCMC procedure is show here using the indices for a topic model. It begins by sampling from $\text{Cat}(\rho)$ to determine whether z_{iw} should be assigned to one of the already instantiated topics, or some new topic cluster. If the sampled $z_{iw} \leq K$, as is common after the first few sampling iterations, we simply not choose that topic. Otherwise, we select a new community by simulating from our stick-breaking prior, since all potential new topics have indistinguishable marginal likelihoods. Such dynamic creation of variables is the key to retrospective samplers. Because our likelihood parameters Ω_{kw} have conjugate Dirichlet priors, we can exactly compute the posterior normalization constant, and the more complex Metropolis-Hastings proposals of [64] are unnecessary. A related approach has been used for inference in infinite depth nested CRP models [16].

DCNT Retrospective MCMC

For the DCNT, the retrospective MCMC comes into play when we sample z_{iw}. Let $M_{kt}^{\setminus iw}$ denote the number of word instances of type t assigned to topic k, excluding token w in document i. Similarly, let $M_{k\cdot}^{\setminus iw}$ be the number of total tokens assigned to topic k. For a vocabulary with T unique word types, our posterior ρ can be expressed as

$$\rho_k \propto \pi_{ki} \left(\frac{M_{kt}^{\setminus iw} + \beta}{M_{k\cdot}^{\setminus iw} + \beta T} \right) \qquad \text{for } k = 1, \ldots, K \tag{3.20}$$

$$\rho_{K+1} \propto \left(1 - \sum_{k=1}^{K} \pi_{ki} \right) \left(\frac{\beta}{\beta T} \right). \tag{3.21}$$

Similarly to the DCNR model, if $z_{iw} > K$, we select a new topic by simply simulating the logistic stick-breaking prior. Our marginalization of Ω and the fact that all new topics have indistinguishable marginal likelihoods allow us to also bypass the expensive Metropolis-Hastings step that would otherwise be necessary.

Algorithm 3.2 Retrospective MCMC resampling of source community s_{ijm}, given a $(K+1)$-dim. posterior distribution ρ defined as in Eqs. (3.18, 3.19).

```
   Draw $s_{ijm} \sim \text{Mult}(\rho)$
1: **if** $s_{ijm} = K + 1$ **then**
   Draw $\eta_{:s_{ijm}} \sim N(\mu, \Lambda^{-1})$
   Draw $v_{s_{ijm}i} \sim N(\eta^T_{:s_{ijm}} \phi_{:i}, \lambda_V^{-1})$
   Draw $\omega \sim \text{Ber}(\psi(v_{s_{ijm}i}))$
2:     **if** $\omega = 1$ **then**
   Exit and keep all instantiated variables
3:     **else**
   Increment $s_{ijm} \leftarrow s_{ijm} + 1$
   Goto line 3
```

3.2.5 Held Out Likelihood for Topic Models

Chib Style Estimation of Predictive Likelihoods

To assess held-out likelihood scores for our topic model experiments, we rely on a Chibs style estimator, which was found to be far more accurate than alternatives like the harmonic mean estimator. The Chib style estimator can be used to approximate the predictive likelihood of a held out document by marginalizing out the topic assignment variables z_d, and topic weights $v_{:d}$ and $u_{:d}$, to obtain $p(w_d \mid \zeta, \Gamma)$, where w_d refers to the set of N words in a held out document d, $\zeta = \{A, \Omega, \eta, \phi, \lambda_V\}$ are the parameters learned from training data, and Γ is the set of hyperparameters specified before training. The Chib-style estimator is based on a distinguished high-probability set of latent variables $(z^*_d, v^*_{:d}, u^*_{:d})$, chosen so that:

$$p(w_d \mid \zeta, \Gamma) = \frac{p(w_d, z^*_d, v^*_{:d}, u^*_{:d} \mid \zeta, \Gamma)}{p(z^*_d, v^*_{:d}, u^*_{:d} \mid w_d, \zeta, \Gamma)} \tag{3.22}$$

$$p(w_d \mid \zeta, \Gamma) \approx \frac{p(w_{:d}, z^*_d, v^*_{:d}, u^*_{:d} \mid \zeta, \Gamma)}{\frac{1}{S}\sum_{s=1}^{S} T(z^*_d, v^*_{:d}, u^*_{:d} \leftarrow z^s_d, v^s_{:d}, u^s_{:d})} \tag{3.23}$$

where $T(z^*_d, v^*_{:d}, u^*_{:d} \leftarrow z^s_d, v^s_{:d}, u^s_{:d})$ is a reversible Markov chain operator used to numerically approximate the marginalization of z_d, $v_{:d}$, and $u_{:d}$ by calculating the transition probabilities from S samples from their respective posterior given w_d. These can be obtained via our standard Gibbs sampling updates for z_d and $u_{:d}$, and our ESS sampler for $v_{:d}$ which we denote by $\mathcal{ESS}(\cdot)$. Depending on the direction of this chain, the respective posterior distributions used to evaluate the transition operators will be different. We denote the forward transition operator as $T(\boldsymbol{z}^*, \boldsymbol{v}^*, \boldsymbol{u}^* \leftarrow \boldsymbol{z}^s, \boldsymbol{v}^s, \boldsymbol{u}^s)$ and the reverse transition operator as $\tilde{T}(\boldsymbol{z}^*, \boldsymbol{v}^*, \boldsymbol{u}^* \leftarrow \boldsymbol{z}^s, \boldsymbol{v}^s, \boldsymbol{u}^s)$ which can be defined as follows:

$$T(\boldsymbol{z}^*, \boldsymbol{v}^*, \boldsymbol{u}^* \leftarrow \boldsymbol{z}^s, \boldsymbol{v}^s, \boldsymbol{u}^s) = p(\boldsymbol{z}^* \mid \boldsymbol{v}^s, \boldsymbol{z}^s) q(\boldsymbol{v}^* \mid \boldsymbol{v}^s, \boldsymbol{z}^*, \boldsymbol{u}^s) p(\boldsymbol{u}^* \mid \boldsymbol{v}^*) \tag{3.24}$$

$$\tilde{T}(\boldsymbol{z}^*, \boldsymbol{v}^*, \boldsymbol{u}^* \leftarrow \boldsymbol{z}^s, \boldsymbol{v}^s, \boldsymbol{u}^s) = p(\boldsymbol{z}^* \mid \boldsymbol{v}^*, \boldsymbol{z}^s) q(\boldsymbol{v}^* \mid \boldsymbol{v}^s, \boldsymbol{z}^s, \boldsymbol{u}^*) p(\boldsymbol{u}^* \mid \boldsymbol{v}^s) \tag{3.25}$$

The log posterior distributions have the following form for the forward transition $T(\cdot)$:

$$\log p(\boldsymbol{z}^* \mid \boldsymbol{v}^s, \boldsymbol{z}^s, \Omega) = \sum_{n=1}^{N} \log \left[\frac{p(w_{dn} \mid z^s_{dn} = z^*_{dn}) p(z^s_{dn} = z^*_{dn} \mid \pi^s)}{\sum_{k=1}^{K} p(w_{dn} \mid z^s = k) p(z^s = k \mid \pi^s)} \right] \tag{3.26}$$

$$\log q(\boldsymbol{v}^* \mid \boldsymbol{v}^s, \boldsymbol{z}^*, \boldsymbol{u}^s) = \log N(\boldsymbol{v}^* \mid A\boldsymbol{u}^s, L) \tag{3.27}$$

$$\log p(\boldsymbol{u}^* \mid \boldsymbol{v}^*) = \log N(\boldsymbol{u}^* \mid (I_{\bar{K}} + A^T LA)^{-1}(A^T L\boldsymbol{v}^* + \eta^T \phi_d), (I_{\bar{K}} + A^T LA)^{-1}) \tag{3.28}$$

Our stick breaking weights for $v_{:d}$ are constructed as $\pi^s_{kd} = \psi(v^s_{kd}) \prod_{\ell=1}^{k-1} \psi(-v^s_{\ell d})$, and the topic counts for document d are denoted by $n^s_k = \sum_{n=1}^{N} \delta(z^s_{dn}, k)$. The precision matrix for $u_{:d}$ under our prior is denoted by $L = \lambda_V I_{\bar{K}}$.

As in previous applications of similar Chib-style estimators, we set the length of the transition chain to be $S = 1000$, and run $T = 1000$ iterations to determine a high posterior probability state. Due to the use of our ESS sampler, we no longer need to reweight the final predictive likelihood as done originally in [43].

3.3 Experiments

For all our experiments, we created five train/test splits where we randomly removed 20% of the edges for testing. The splits we found in practice enforce more variability in the AUC performance rather than averaging over random initializations of the model. For all our experiments, we ran our MCMC chain for 10000 iterations, using the saved model parameters at every 50 iterations to calculate our expected edged probabilities. To initialize, we implement a sequential Gibbs sampler that begins by sampling token/edge assignments incrementally. We perform this for 100 iterations, before using these assignments to approximate our latent variables higher in the generative process.

We set our hyperparameters in the following manner for both the relational and topic model version of the DCNM. We initialize λ_V with the corresponding variance associated with the representation of the Dirichlet as being sampled from marginal beta distributions. This mimics to some degree the finite approximation of the stick-breaking process such that $\pi_1 \sim \text{Beta}(1, \alpha)$, where $\alpha = 1/K$. For other precision parameters, we set $\lambda_F = 0.05, \lambda_A = 0.01, \lambda_\mu = 0.25$, representing a general willingness to allow for more variance for latent variables higher within the generative process. All other variables are learned either from Gibbs Sampling or the Elliptical Slice Sampler.

3.4 DCNR: Relational Datasets

To better understand the capabilities of the DCNR, we will look at several datasets to acquire quantitative and qualitative metrics. For quantitative results, we will focus on AUC (Area under the Curve), which represents the measure to which the model is capable of predicting unseen edges.

Our comparison models are as follows. The first will be the Mixed Membership Stochastic Block Model [3], which represents the parametric baseline for comparison to our DCNR model. For the MMSB there are two variants, where Gibbs sampling is used for inference, while the other uses the NUTS (No U-Turn) from STAN [34]. We also implement a similar version of the DCNR model in STAN without correlation or metadata, using a normal logistic stick-breaking prior. Unlike the DCNR, the model is truncated permanently. With these three comparison models, we test against a variety of DCNR variants with or without correlations, metadata, or truncation.

3.4.1 Lazega Lawyers Network

The Lazega lawyers dataset [48] is a social network between partners and associates of several New England law firms. Collected from 1988-1991, it contains three directed binary relations encoding friendship, coworker, and advisory relationships among 71 lawyers. The dataset contains a rich set of metadata describing status, gender, office location, years employed, age, practice, and law school. The strong coworkers network represents relationships where two lawyers performed professional work with together on at least one case. The advice network represents lawyers who had sought one another for professional advice to ensure that they were accomplishing their duties (such as handling a case properly) or consulting someone for a professional opinion that was of great value. Finally, the friendship network represents lawyers who have socialized together outside of work and whose families might know one another 3.5.2.

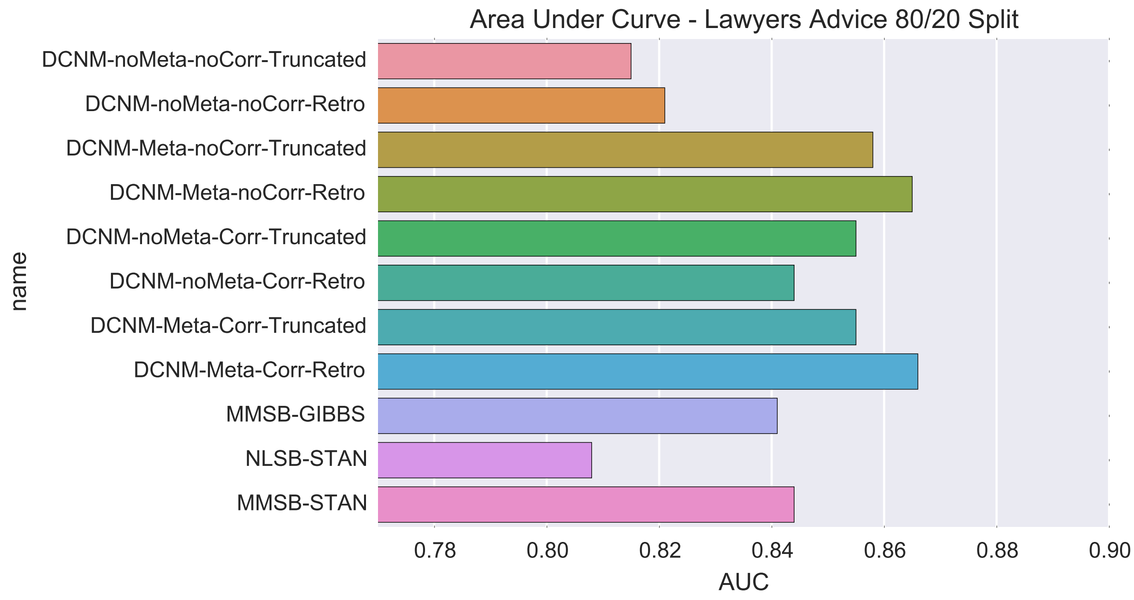


Figure 3.6: AUC Results for the Lawyers Advice Relation.

The current results from the lawyers dataset suggest a few interesting trends. The first is that the current DCNM model without metadata or correlations performs comparably with the MMSB and that incorporating metadata and the retrospective MCMC ensures that the model will perform significantly better. For AUC, modeling correlations does not seem to improve this measure as much as the other two features for the DCNM.

3.4.2 Otago Harbour Food Web

The Otago Harbour dataset [54] contains a single "who-eats-whom" binary relationship for 123 organisms from an intertidal mudflat ecosystem in New Zealand. In addition to predation links, the dataset contains metadata that broadly classifies each node as one of 21 possible organism types (e.g., annelids, birds), and assigns one of three mobility ratings (low, intermediate, high). A variety of organisms populate this food web including secondary predators (ducks, fish), primary predators (rock crabs), and autotrophs (seagrass). We explore whether unsupervised learning from metadata can contribute to knowledge discovery in complex, real-world networks.

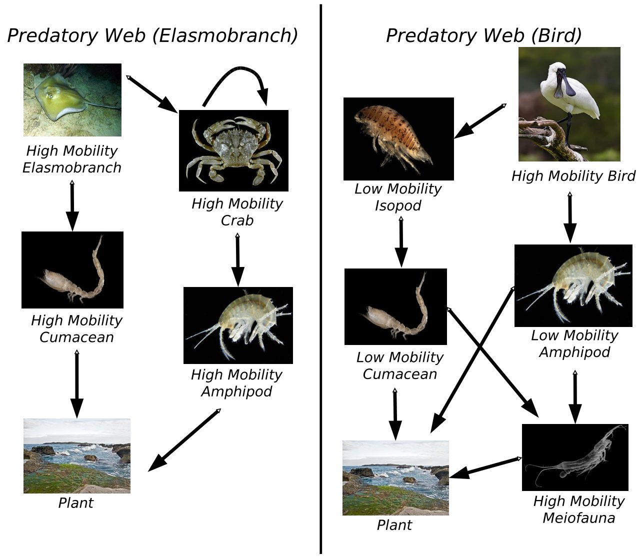


Figure 3.7: Food webs traced from two top predator archetypes, a *high mobility elasmobranch* (sharks, rays, and skates) and a *high mobility bird*. Given learned connections between metadata and predation relationships, links are drawn from each node to the most likely prey archetypes, continuing until the bottom of the web. The model recovers biologically relevant structure from binary relations among 123 species. Images courtesy Wikipedia. Results are from the DCNM model without correlations.

3.5 DCNT: Documents

Similarly for our topic modeling experiments, we created five train/test splits where 20% of the documents were randomly removed for held-out likelihood metrics. For quantitative results, we implement a Chibs Style estimator [81] as described above across several topic models. We use the last iteration of our MCMC chain to represent our global latent variables and run the Chibs Style estimator for 1000 iterations per held-out document.

3.5.1 NIPS Data

We analyze a collection of conference papers from NIPS (Neural Information Processing Systems), which contains a total of 1392 documents with metadata pertaining to the year and author. The conference represents a broad range of machine learning topics and makes it an ideal dataset to discover themes that pertain to the field's specialties, such as neural networks and Bayesian modeling. After pruning for stopwords, the final vocabulary contains 13649 unique terms. The richness of this dataset and its associated metadata allows us to not only report held-out likelihood scores, but also visualize the influence of metadata and to see if there are any interesting correlation structures for our recovered topics.

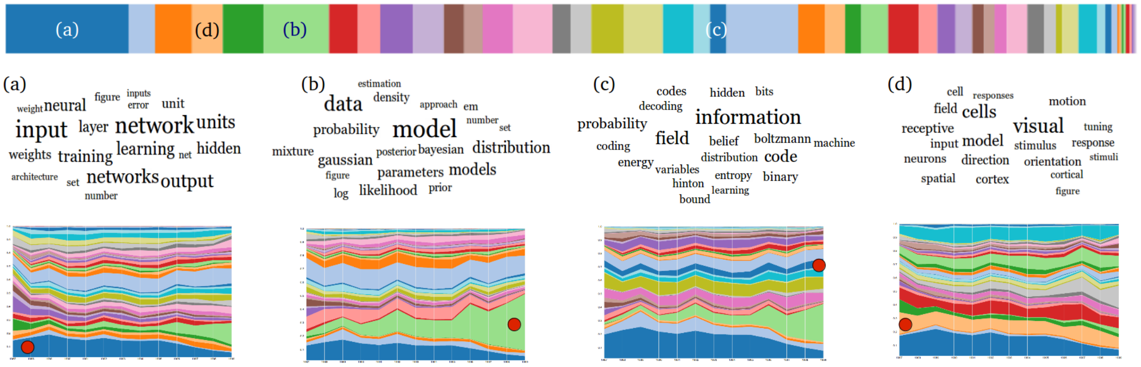
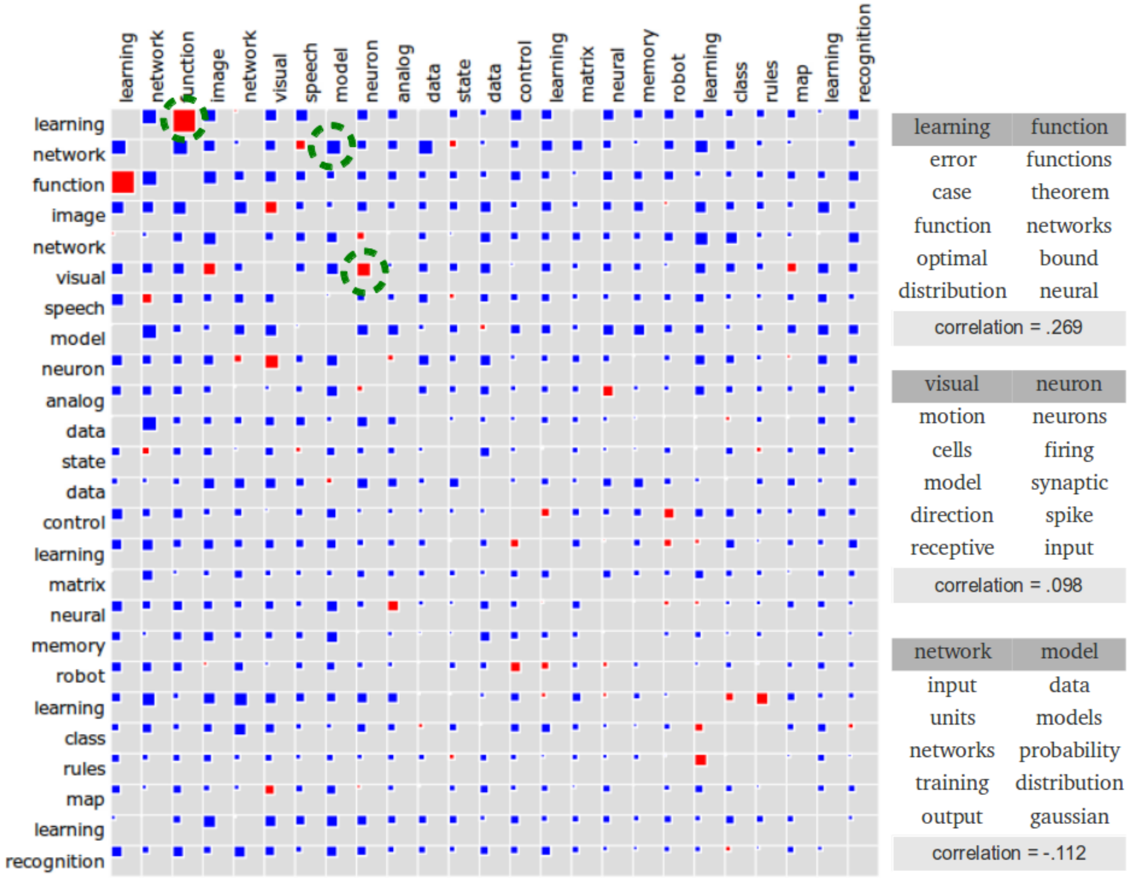


Figure 3.8: (a) The topic masses for a set of test documents conditioned by years only, years and Michael I. Jordan (b), years and Geoff Hinton (c), and years with Terrence Sejnowski (d). A stacked histogram at the top of the figures shows the topic masses across the whole corpus. Due to the logistic stick-breaking process prior, the topic masses are naturally shifted towards the first few topics. For each column of this figure, a cloud of words from a relevant topic are shown that represents the effect of conditioning on metadata. Larger words have more probability mass within that topic while the red dot indicates the topic mass associated with the test document below.

One of the advantages of using a generative model such as the DCNM is the ability to explore topic distributions that vary conditioned on specific features. In 3.5.1, a test document was generated for each year and conditioned on either the author Michael I. Jordan, Geoffrey Hinton, or Terrence

Sejnowski. For each respective author, words from a relevant topic show how conditioning on a particular author can change the topic masses for that test document. For example, the visualization associated with Michael Jordan shows how the topic associated with terms related to probabilistic models gradually increases over the years while the topic associated with neural networks decreases. Conditioning on Geoffrey Hinton puts larger mass on a topic that contains his actual name and aspects of his research work. Finally, conditioning on Terrence Sejnowski shows how a large proportion of the topic mass is shifted to topics related to neuroscience.



learning	function
error	functions
case	theorem
function	networks
optimal	bound
distribution	neural
correlation = .269	

visual	neuron
motion	neurons
cells	firing
model	synaptic
direction	spike
receptive	input
correlation = .098	

network	model
input	data
units	models
networks	probability
training	distribution
output	gaussian
correlation = -.112	

Figure 3.9: A Hinton diagram of correlations from a DCNT model where values in red and blue represent positive and negative correlations respectively. To the right are the top six words in order of their frequency.

The DCNM model can also find correlations between topics and in 3.5.1, we represent this using a Hinton diagram where the size of a colored grid is proportional to the magnitude of the correlation between two topics. The results displayed in this figure are from a model where no metadata was used and to the right of the Hinton diagram are words associated with topics from three correlations. From this figure, we can see that the DCNM model found strong positive correlations between the ”learning” and ”function” topic where the two topics have strong semantic similarities, but are not necessarily identical topics. Another correlation that the DCNT discovered was a positive correlation between the topics ”visual” and ”neuron”, where the two topics represent meaningful semantic overlaps, particularly in papers that represent the study of the brain’s visual cortex. Finally, a negative correlation was found between ”network” and ”model” which might reflect the ideological differences that existed then between the field of neural networks and probabilistic graphical models.

3.5.2 MusixMatch Song Lyrics

The MusixMatch dataset is a collection of song lyrics that have been converted into a bag of words format. The original dataset consisted of approximately 1 million songs and contains metadata for many of these songs such as the year and artist. For the DCNM analysis of this data, we focus on a small subset of this data, which contains approximately 2537 songs. We obtain this subset by filtering for tracks that contain only the most popular artists with a total of at least 100 tracks. The MusixMatch results shows a similar pattern of performance with the AUC results from the relational datasets. In general, the DCNM model without metadata or correlations performs similarly with the HDP-Gibbs Topic model and performs significantly better than LDA-STAN. The DCNM benefits most from metadata and the retrospective MCMC and marginally better from modeling correlations.

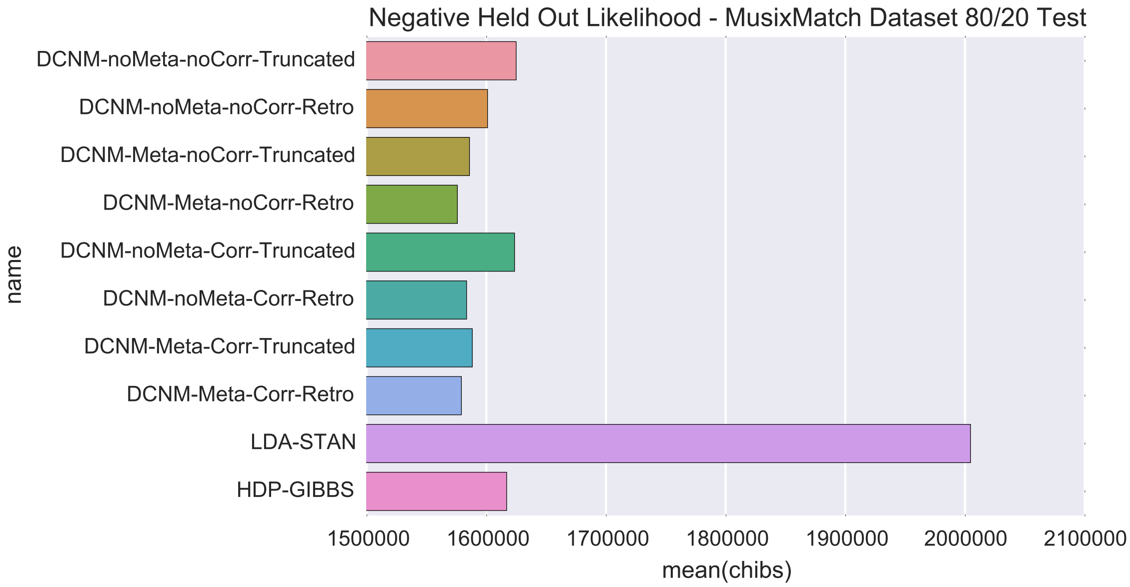


Figure 3.10: Negative Held Out Likelihood scores for the MusixMatch dataset.

Chapter 4

Scalable Inference in Bayesian Nonparametric Relational Models

One of the significant limitations of standard MCMC techniques arises when it comes to scalability. Variational inference, which casts the problem of posterior approximation as a deterministic optimization problem is typically faster with guarantees of convergence, but at the cost of solution quality. For networks, since the number of edges grows quadratically with the size of the nodes, scalable inference techniques become necessary for most real-world networks. We begin by first describing previous work on variational inference for the HDP topic model. We then introduce an HDP version for relational datasets and derive its naive mean-field variational inference updates. An assortativity assumption for the HDPR is introduced where our stochastic block matrix is constrained to be a diagonal matrix with a fixed offset for non-assortative communities. We show how this constraint allows updates to be performed in linear time and complexity with the number of communities. Finally, we show how stochastic variational inference may be applied to the aHDPR model to scale to networks with tens of thousands of nodes.

4.1 Variational Inference for HDP-Relational Models

We briefly review the generative process for the HDPR model from Chapter 2 to define the notation used in deriving a variational inference algorithm for this model. Note that the HDP prior over community membership distributions for the HDPR is exactly the same as the prior over the HDP topic model. Specifically, we define a set of global stick-breaking weights drawn as $\beta \sim \mathrm{GEM}(\gamma)$ and define our second level DP representing our mixed-membership community distributions as $\pi_i \sim \mathrm{DP}(\alpha\beta)$ for node i in our graph Y with N nodes. Note that this part of the model is equivalent to the HDPT. To generate an edge y_{ij} which connects node i to node j, we first sample a pair of indicator variables from their corresponding community membership distributions, $s_{ij} \sim \mathrm{Cat}(\pi_i)$, $r_{ij} \sim \mathrm{Cat}(\pi_j)$ which indicate the row/column entry of our stochastic block matrix $\Omega_{k\ell} \sim \mathrm{Beta}(\tau_a, \tau_b)$. We define the probability of generating an edge as $p(y_{ij} = 1 | s_{ij} = k, r_{ij} = \ell) = \Omega_{k\ell}$. This generative process models a binary directed network that is capable of expressing a variety of community structures such as assortative communities, dissassortative communities, core-periphery networks, and even hierarchical community structures.

4.1.1 Undirected Networks and the Assortative HDPR

Consider the scenario where we are working with binary undirected networks. The full stochastic block matrix will no longer be an appropriate parameter space for these types of networks due to the redundant set of parameters that were originally needed to capture the direction of a particular edge. This implies that either a lower or upper triangular formulation of our stochastic block matrix would be more appropriate. However, when we allow for nodes to be members of multiple communities, the membership communities π can compete with the entries associated with our triangular block matrix Ω [26]. To mitigate this, we assume an assortativity constraint on our stochastic block matrix and redefine our likelihood terms as follows:

$$p(y_{ij} = 1 \mid s_{ij} = r_{ij} = k) = \Omega_k, \qquad p(y_{ij} = 1 \mid s_{ij} \neq r_{ij}) = \epsilon. \tag{4.1}$$

For our assortative HDPR model, each community has its own self-connection probability $\Omega_k \sim \mathrm{Beta}(\tau_a, \tau_b)$. To capture the sparsity of real networks, we fix a very small probability of between-community connection, $\epsilon = 10^{-30}$. The small epsilon term forces the model to rely on the assortative communities to explain most of the data.

4.1.2 Structured Variational Inference

The original HDPR model associate a pair of community assignments, s_{ij} and r_{ij}, with each potential edge y_{ij}. In assortative models these variables are strongly dependent, since present edges only have non-negligible probability for consistent community assignments. To improve accuracy and reduce local optima, we thus develop a structured variational method based on joint configurations of these assignment pairs, which we denote by $e_{ij} = (s_{ij}, r_{ij})$. See Figure 4.1.

Given this alternative representation, we aim to approximate the joint distribution of the observed edges y, local community assignments e, and global community parameters π, Ω, β given fixed

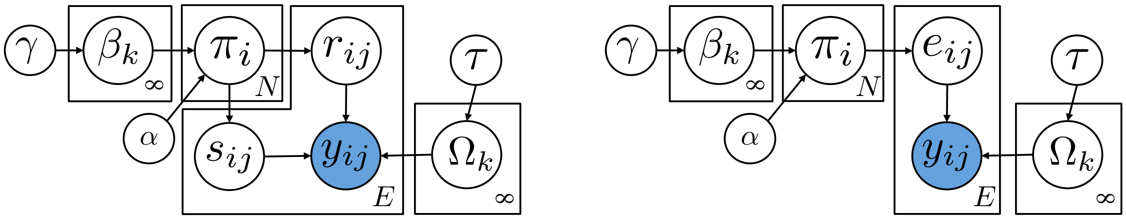


Figure 4.1: Alternative graphical representations of the aHDPR model, in which each of N nodes has mixed membership π_i in an unbounded set of latent communities, w_k are the community self-connection probabilities, and y_{ij} indicates whether an edge is observed between nodes i and j. *Left:* Conventional representation, in which source s_{ij} and receiver r_{ij} community assignments are independently sampled. *Right:* Blocked representation in which $e_{ij} = (s_{ij}, r_{ij})$ denotes the pair of community assignments underlying y_{ij}.

hyperparameters τ, α, γ. The overall variational objective can now be defined as:

$$
\begin{aligned}
\mathcal{L}(q) &= \mathbb{E}[\log p(y|\Omega, e)] + \mathbb{E}[\log p(e|\pi)] + \mathbb{E}[\log p(\pi \mid \alpha, \beta(v^*))] + \mathbb{E}[\log p(\Omega \mid \tau_a, \tau_b)] + \mathbb{E}[\log p(v^* \mid \gamma)] \\
&\quad - \mathbb{E}[\log q(e|\phi)] - \mathbb{E}[\log q(\pi \mid \theta)] - \mathbb{E}[\log q(\Omega \mid \lambda)] \qquad (4.2) \\
&= \sum_{ij}^{E} \sum_{k=1}^{K} \left[\phi_{ijkk} \log f(\Omega_k, y_{ij}) \right] + \sum_{ij}^{E} \left[1 - (\sum_{k=1}^{K} \phi_{ijkk}) \log f(\epsilon, y_{ij}) \right] \qquad (4.3) \\
&\quad + \sum_{ij}^{E} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \left[\phi_{ijk\ell} (\mathbb{E}_q[\log(\pi_{ik})] + \mathbb{E}_q[\log(\pi_{j\ell})]) \right] \qquad (4.4) \\
&\quad + \sum_{k=1}^{K} (\gamma - 1) \log(1 - v_k) + \sum_{i=1}^{N} \left[\log \Gamma(\sum_{k=1}^{K^+} \alpha\beta_k) - \sum_{k=1}^{K^+} \log \Gamma(\alpha\beta_k) + \sum_{k=1}^{K^+} (\alpha\beta_k - 1) \mathbb{E}[\log \pi_{ik}] \right] \\
&\quad + \sum_{k=1}^{K^+} \left[\log \left(\frac{\Gamma(\tau_a + \tau_b)}{\Gamma(\tau_a)\Gamma(\tau_b)} \right) + (\tau_a - 1)\mathbb{E}_q[\log(\Omega_k)] + (\tau_b - 1)\mathbb{E}_q[\log(1 - \Omega_k)] \right] \\
&\quad - \sum_{ij}^{E} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \phi_{ijk\ell} \log(\phi_{ijk\ell}) \qquad (4.5) \\
&\quad - \sum_{i=1}^{N} \left[\log \Gamma(\sum_{k=1}^{K^+} \theta_{ik}) - \sum_{k=1}^{K^+} \log \Gamma(\theta_{ik}) + \sum_{k=1}^{K^+} (\theta_{ik} - 1) \mathbb{E}[\log \pi_{ik}] \right] \\
&\quad - \sum_{k=1}^{K^+} \left[\log \left(\frac{\Gamma(\lambda_{ka} + \lambda_{kb})}{\Gamma(\lambda_{ka})\Gamma(\lambda_{kb})} \right) + (\lambda_{ka} - 1)\mathbb{E}_q[\log(\Omega_k)] + (\lambda_{kb} - 1)\mathbb{E}_q[\log(1 - \Omega_k)] \right].
\end{aligned}
$$

Here we define $f(\Omega_k, y_{ij}) = \exp\{y_{ij}\mathbb{E}_q[\log(\Omega_k)] + (1 - y_{ij})\mathbb{E}_q[\log(1 - \Omega_k)]\}$. The sufficient statistics for $\mathbb{E}_q[\log \pi_{ik}]$ are equivalent to the HDPT. The sufficient statistics related to our stochastic block matrix can be decomposed as follows:

$$
\mathbb{E}_q[\log \Omega_{k\ell}] = \psi(\lambda_{k\ell a}) - \psi(\lambda_{k\ell a} + \lambda_{k\ell b}), \qquad \mathbb{E}_q[\log \Omega_{k\ell}] = \psi(\lambda_{k\ell b}) - \psi(\lambda_{k\ell a} + \lambda_{k\ell b}) \qquad (4.6)
$$

By taking partial derivatives with respect to our free variational parameters, we obtain coordinate ascent updates similar to the HDP Topic model.

$$\phi_{ijk\ell} \propto \exp\{\mathbb{E}_q[\log \pi_{ik}] + \mathbb{E}_q[\log \pi_{j\ell}]\} f(\epsilon, y_{ij}) \tag{4.7}$$

$$\phi_{ijkk} \propto \exp\{\mathbb{E}_q[\log \pi_{ik}] + \mathbb{E}_q[\log \pi_{j\ell}]\} f(\Omega_k, y_{ij}) \tag{4.8}$$

$$\lambda_{ka} = \tau_a + \textstyle\sum_{ij}^{E} \phi_{ijkk} y_{ij}, \tag{4.9}$$

$$\lambda_{kb} = \tau_b + \textstyle\sum_{ij}^{E} \phi_{ijkk}(1 - y_{ij}), \tag{4.10}$$

$$\theta_{ik} = \alpha\beta_k + \textstyle\sum_{(i,j)\in E} \sum_{\ell=1}^{K} \phi_{ijk\ell}. \tag{4.11}$$

Here, the final summation is over all potential edges $(i, j) \in E$ linked to node i. For these updates, the mixed-membership distributions associated with each node π is treated as a global variable and is of interest. This is partially due to the value of the latent structure associated with π and the lack of interpretability associated with Ω once the final communities are learned.

4.1.3 Linear Time and Storage Complexity for the aHDPR

A naive implementation of these updates would require $\mathcal{O}(K^2)$ computation and storage for each assignment distribution $q(e_{ij} \mid \phi_{ij})$. Note, however, that the updates for $q(\Omega_k \mid \lambda_{ka}, \lambda_{kb})$ in Eq. (4.9),(4.10) depend only on the K probabilities ϕ_{ijkk} that nodes select the same community. Using the updates for $\phi_{ijk\ell}$ from Eq. (4.7), the update of $q(\pi_i \mid \theta_i)$ in Eq. (4.11) can be expanded as follows:

$$\begin{aligned} \theta_{ik} &= \alpha\beta_k + \textstyle\sum_{(i,j)\in E} \phi_{ijkk} + \frac{1}{Z_{ij}} \sum_{\ell \neq k} \tilde{\pi}_{ik}\tilde{\pi}_{j\ell} f(\epsilon, y_{ij}) \\ &= \alpha\beta_k + \textstyle\sum_{(i,j)\in E} \phi_{ijkk} + \frac{1}{Z_{ij}} \tilde{\pi}_{ik} f(\epsilon, y_{ij})(\tilde{\pi}_j - \tilde{\pi}_{jk}). \end{aligned} \tag{4.12}$$

where $\tilde{\pi}_{ik} \triangleq \exp\{\mathbb{E}_q[\log(\pi_{ik})]\} = \exp\{\psi(\theta_{ik}) - \psi(\sum_{\ell=1}^{K+1} \theta_{i\ell})\}$ and $\tilde{\pi}_i \triangleq \sum_{k=1}^{K} \tilde{\pi}_{ik}$. Note that $\tilde{\pi}_j$ need only be computed once, in $\mathcal{O}(K)$ operations. The normalization constant Z_{ij}, which is defined so that ϕ_{ij} is a valid categorical distribution, can also be computed in linear time:

$$Z_{ij} = \tilde{\pi}_i \tilde{\pi}_j f(\epsilon, y_{ij}) + \textstyle\sum_{k=1}^{K} \tilde{\pi}_{ik}\tilde{\pi}_{jk}(f(w_k, y_{ij}) - f(\epsilon, y_{ij})). \tag{4.13}$$

Finally, to evaluate our variational bound and assess algorithm convergence, we still need to calculate the likelihood and entropy terms dependent on $\phi_{ijk\ell}$. However, we can compute part of our bound by caching our partition function Z_{ij} in linear time. In particular, we focus first on Eq. (4.4) where $\sum_{ij}^{E} \sum_{k=1}^{K} \sum_{\ell=1}^{K} [\phi_{ijk\ell}(\mathbb{E}_q[\log(\pi_{ik})] + \mathbb{E}_q[\log(\pi_{j\ell})])]$ can be further decomposed as:

$$\sum_{ij}^{E} \left[\sum_{k=1}^{K} \mathbb{E}_q[\log(\pi_{ik})] \sum_{\ell=1}^{K} \phi_{ijk\ell} + \sum_{\ell=1}^{K} \mathbb{E}_q[\log(\pi_{j\ell})] \sum_{k=1}^{K} \phi_{ijk\ell} \right]$$

Furthermore, looking at 4.5, we see that $\sum_{ij}^{E}\sum_{k=1}^{K}\sum_{\ell=1}^{K}[\phi_{ijk\ell}\log(\phi_{ijk\ell})]$ can also be further expanded into

$$\sum_{ij}^{E}\Bigg[\sum_{k=1}^{K}\mathbb{E}_q[\log(\pi_{ik})]\sum_{\ell=1}^{K}\phi_{ijk\ell}+\sum_{\ell=1}^{K}\mathbb{E}_q[\log(\pi_{j\ell})]\sum_{k=1}^{K}\phi_{ijk\ell}$$
$$-\log\sum_{k=1}^{K}\phi_{ijkk}+\sum_{k=1}^{K}\log f(w_k,y_{ij})\phi_{ijkk}-\log(Z_{ij})\Bigg] \tag{4.14}$$

In order to calculate our ELBO in an efficient manner the terms we need to simplify from Eq. (4.4) and Eq. (4.5) are the terms related to $\phi_{ijk\ell}$ s.t:

$$\textstyle\sum_{k=1}^{K}\mathbb{E}_q[\log(\pi_{ik})]\sum_{\ell=1}^{K}\phi_{ijk\ell}=\sum_{k=1}^{K}\tilde{\pi}_{ik}\Big[\phi_{ijkk}+\frac{1}{Z_{ij}}\tilde{\pi}_{ik}f(\epsilon,y_{ij})(\tilde{\pi}_j-\tilde{\pi}_{jk})\Big]$$
$$\textstyle\sum_{\ell=1}^{K}\mathbb{E}_q[\log(\pi_{j\ell})]\sum_{k=1}^{K}\phi_{ijk\ell}=\sum_{\ell=1}^{K}\tilde{\pi}_{j\ell}\Big[\phi_{ij\ell\ell}+\frac{1}{Z_{ij}}\tilde{\pi}_{j\ell}f(\epsilon,y_{ij})(\tilde{\pi}_i-\tilde{\pi}_{i\ell})\Big]$$

Note the similarity of this expression with part of the updates in Eq. (4.18). By caching the necessary statistics needed to update θ, we can calculate our ELBO in an efficient manner.

4.1.4 Stochastic Variational Inference for HDPR

Using the same strategy for deriving the stochastic variational updates for the HDP topic model, we also take natural gradients with respect to our new ELBO for λ, θ, v in the HDPR model.

$$\nabla\lambda_{ka}^{*}=\tfrac{1}{g(i,j)}\phi_{ijkk}y_{ij}+\tau_a-\lambda_{ka}; \tag{4.15}$$
$$\nabla\theta_{ik}^{*}=\tfrac{1}{g(i,j)}\textstyle\sum_{(i,j)\in E}\sum_{\ell=1}^{K}\phi_{ijk\ell}+\alpha\beta_k-\theta_{ik}, \tag{4.16}$$

where the natural gradient for $\nabla\lambda_{kb}^{*}$ is symmetric to $\nabla\lambda_{ka}^{*}$ and where y_{ij} in Eq. (4.15) is replaced by $(1-y_{ij})$. Note that $\sum_{(i,j)\in E}\sum_{\ell=1}^{K}\phi_{ijk\ell}$ was shown in the previous section to be computable in $\mathcal{O}(K)$. The scaling term $g(i,j)$ is needed for an unbiased update to our expectation. If $g(i,j) = 2/N(N-1)$, then this would represent a uniform distribution over possible edge selections in our undirected graphs. In general, $g(i,j)$ can be an arbitrary distribution over possible edge selections such as a distribution over sets of edges as long as the expectation with respect to this distribution is equivalent to the original ELBO [26]. When referring to the scaling constant associated with sets, we consider the notation of $h(T)$ instead of $g(i,j)$.

We optimize this ELBO with a Robbins-Monro algorithm which iteratively steps along the direction of this noisy gradient. We specify a learning rate $\rho_t \triangleq (\mu_0+t)^{-\kappa}$ at time t where $\kappa \in (.5, 1]$ and $\mu_0 \geq 0$ down weights the influence of earlier updates. With the requirement that $\sum_t \rho_t^2 < \infty$ and $\sum_t \rho_t = \infty$, we will provably converge to a local optimum. For our global variational parameters

$\{\lambda, \theta\}$, the updates at iteration t are now

$$\lambda_{ka}^{t} = \lambda_{ka}^{t-1} + \rho_t(\nabla\lambda_{ka}^{*}) = (1-\rho_t)\lambda_{ka}^{t-1} + \rho_t(\tfrac{1}{g(i,j)}\phi_{ijkk}y_{ij} + \tau_a); \tag{4.17}$$

$$\theta_{ik}^{t} = \theta_{ik}^{t-1} + \rho_t(\nabla\theta_{ik}^{*}) = (1-\rho_t)\theta_{ik}^{t-1} + \rho_t(\tfrac{1}{g(i,j)}\textstyle\sum_{(i,j)\in E}\sum_{\ell=1}^{K}\phi_{ijk\ell} + \alpha\beta_k); \tag{4.18}$$

$$v_k^t = (1-\rho_t)v_k^{t-1} + \rho_t(v_k^*), \tag{4.19}$$

where v_k^* is obtained via a constrained optimization task using the gradients derived in $\ddagger A.2$. Defining an update on our global parameters given a single edge observation can result in very poor local optima. In practice, we specify a mini-batch T, a set of unique observations in determining a noisy gradient that is more informative. This results in a simple summation over the sufficient statistics associated with the set of observations as well as a change to $g(i,j)$ to reflect the necessary scaling of our gradients when we can no longer assume our samples are uniformly chosen from our dataset.

4.1.5 Restricted Stratified Node Sampling

Large real-world networks are sparse and our optimization algorithm provides us with the ability to choose a sampling scheme that allows us to better exploit this sparsity. Given the success of stratified node sampling on sparse networks as a mini-batch strategy [26] we consider this technique for all our experiments. Briefly, stratified node-sampling randomly selects a single node i and either chooses its associated links or a set of edges from m equally sized non-link edge sets. For this mini-batch strategy, $h(T) = 1/N$ for link sets and $h(T) = 1/Nm$ for a partitioned non-link set. In [26], π was treated as a global parameter where every node parameter was updated after each mini-batch. For our model, we also treat π as a global parameter, but maintain a separate learning rate ρ_i for each node. This allows us to focus on updating only nodes that are relevant to our mini-batch as well as limit the computational costs associated with this global update. To ensure that our Robbins-Monro conditions are still satisfied, we assume that for node i, its learning rate $\rho_{it} = 0$ for nodes that are not part of our current mini-batch. When a new mini batch contains this particular node, we look to the most previous learning rate so that $\rho_{it^*} > 0$ and assume this value as the previous learning rate. This modified subsequence of learning rates maintains our convergence criterion so that the $\sum_t \rho_{it}^2 < \infty$ and that $\sum_t \rho_{it} = \infty$. We show how performing this simple modification results in significant improvements in both perplexity and link prediction scores.

4.1.6 HDPR Pruning Moves

For the HDPR, our nested truncation requires setting an initial number of communities K. A large truncation lets the posterior find the best number of communities, but can be computationally costly. A truncation set too small may not be expressive enough to capture the best approximate posterior. To remedy this, we define a set of pruning moves aimed at improving inference by removing communities that have significantly small probability mass. Pruning moves provide the model with a more parsimonious and interpretable latent structure as well as saving us significant computational costs during inference. To determine candidates for pruning, we define $\Theta_k = (\sum_{i=1}^{N} \theta_{ik})/(\sum_{i=1}^{N} \sum_{k=1}^{K} \theta_{ik})$ for all $k \in K$. We then threshold Θ_k so that any $\Theta_k < \log K/N$ are potential prune candidates. The threshold represents a ratio that promotes pruning moves when the model becomes significantly overparameterized. If these communities are less than this threshold for a consecutive number of t^* iterations (for our experiments we set $t^* = N/2$), we then target them as candidates for pruning. For clarity, assume that $d = 2$ communities k', k'' are considered for removal. We determine a new configuration of our model by estimating the combined mass of our variational posteriors for $\tilde{\theta}_i = \theta_{ik'} + \theta_{ik''} \forall i \in N$. We redistribute this mass uniformly to our remaining communities $k \neq k', k''$ so that $\theta^*_{ik} = \theta_{ik} + \tilde{\theta}_i/(K - d)$. An analogous operation is performed for $\beta, \lambda_a, \lambda_b$, which results in a new set of global variational parameters $\{v^*, \beta^*, \lambda^*_a, \lambda^*_b, \theta^*\}$.

To estimate an informative, but approximate ELBO for this model we need to define a set of relevant observations associated with our prune candidates. To consider all observations would be equivalent to calculating the ELBO within a batch setting, which is intractable for large networks. However, by considering the top n^* nodes (for our experiments we set $n^* = 10$) containing the greatest mass for $\theta_{ik'}, \theta_{ik''} \forall \; i \in N$ and taking all its pairwise edges between these chosen nodes, we obtain a set of relevant edge pairs $y_{ij \in S}$, which we then use to calculate $\phi^*_{ij \in S}$, where S is a set of node indices of relevance to the communities we wish to prune. Given our full set of variational posteriors, we denote this new ELBO as $\mathcal{L}(q^{\text{prune}})$. We also calculate an ELBO before our pruning moves which we denote as $\mathcal{L}(q^{old})$. We accept our new model if $\mathcal{L}(q^{\text{prune}}) > \mathcal{L}(q^{old})$ or reject otherwise. The key idea is that the edge observations formed by nodes with significant mass in $\theta_{ik'}, \theta_{ik''} \forall \; i \in N$ will form an approximation to the ELBO which is more informative then an ELBO formed by a random subset of our graph. Our structured mean-field approach also results in a simple direct update for $\phi^*_{ij \in S}$ which allows us to calculate $\mathcal{L}(q^{old})$ and $\mathcal{L}(q^{\text{prune}})$ more efficiently.

4.2 Experiments

In this section we perform experiments that compare the performance of the aHDPR model to the aMMSB. We show significant gains in AUC and perplexity scores by using the restricted form of stratified node sampling, a quick K-means initialization[1] for θ, and our efficient structured mean-field approach combined with pruning moves. We perform a detailed comparison on a synthetic toy dataset and the relativity collaboration network on a variety of metrics to show the benefits of each contribution. We then show significant improvements over the current aMMSB model in both AUC and perplexity metrics on several real world datasets also analyzed in [26] for the aMMSB. Finally,

[1] Our K-means initialization assumes the rows of our adjacency matrix determined from our non-heldout set as data points with $N - 1$ features, which results in a hard clustering assignment z_i for each node. To initialize, we set $\theta_{iz_i} = N - 1$ and $\theta_{i \setminus z_i} = \alpha$.

we perform a qualitative analysis on the LittleSis network and depict the usefulness of using our learned latent community structure to drive visualizations of large networks.

For all our experiments, we fix the variance across node community memberships $\alpha = 1$ and set our hyperparameters for w_k to $\tau_a = 10$ and $\tau_b = 1$ across communities. We set an aggressive learning rate so that $\mu_0 = 1$ and $\kappa = .5$. We use a restricted stratified node-sampling technique for all our experiments with the non-link partition set $m = 10$, unless stated otherwise. All experiments were run for 250,000 iterations from 5 random initializations with 10% of the links randomly held out along with an equal amount of non-links for testing. For the aMMSB, we used the same settings[2] that were optimized for its original experiments [26], with the exception of changing the Dirichlet prior to be uniform over its mixed-membership distributions ($\alpha = 1$), which we found to improve convergence for the aMMSB across our experiments.

4.2.1 Synthetic and Collaboration Networks

The synthetic network we use for testing is generated from the standards and software outlined in [46] to produce realistic synthetic networks with overlapping communities and power-law degree distributions. For these purposes, we set the number of nodes $N = 1000$, with the minimum degree per node set to 10 and its maximum to 60. On this network the true number of latent communities was found to be $K = 56$. Our real world networks comprise of 5 undirected networks originally ranging from $N = 5,242$ to $N = 27,770$. These raw networks however contain several disconnected components. Both the aMMSB and aHDPR would separate these into distinct non-overlapping communities which defeats the purpose of the model's ability to discover this type of structure. As a result, we took the largest connected component for each graph for analysis.

Initialization and Global Update Strategies. The upper left figures of Fig. 4.2 are within model comparisons of the aHDPR on perplexity for both the synthetic and relativity networks. Here we compare the benefits of initializing θ via K-means and our restricted stratified node sampling procedure. For our random initializations, we initialized θ in the same fashion as the aMMSB. Using a combination of both modifications, we achieve the best perplexity scores on these datasets. The rest of our experiments with the aHDPR model and its variants (naive mean field and pruning) assumes a restricted stratified node sampling approach combined with a K-means initialization.

Naive Mean-Field vs. Structured Mean-Field. The naive mean-field approach is the aHDPR model where the community indicator assignments are split into s_{ij} and r_{ij}. This can result in severe local optima due to their coupling as seen in some experiments in Fig. 4.3. The aMMSB in some instances performs better than the naive mean-field approach, but this can be due to differences in our initialization procedures. However, by changing our inference procedure to an

[2] The aMMSB uses a random initialization for $\theta_{ik} \sim \text{Gam}(100, .01)$ with hyperparameters over w_k set to the expected number of link/non-links across K uniformly distributed communities. It learning rate was set to $\mu_0 = 1024$ and $\kappa = .5$. We found these settings gave the best advantage for the aMMSB on these datasets.

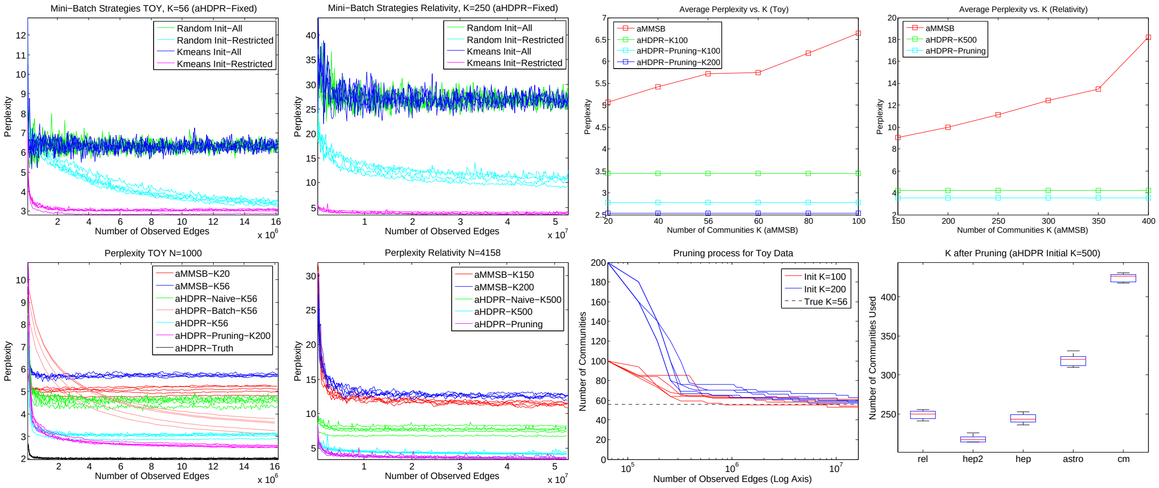


Figure 4.2: The upper left shows benefits of a restricted update and a K-means initialization for stratified node sampling on both synthetic and relativity networks. The upper right shows the sensitivity of the aMMSB as K varies versus the aHDPR. The lower left shows various perplexity scores for the synthetic and relativity networks with the best performing model (aHDPR-Pruning) scoring an average AUC of $0.9675 \pm .0017$ on the synthetic network and $0.9466 \pm .0062$ on the relativity network. The lower right shows the pruning process for the toy data and the final K communities discovered on our real-world networks.

efficient structured mean-field approach, this effect is greatly mitigated across all datasets.

Benefits of Pruning Moves. Pruning moves were applied every $N/2$ iterations with a maximum of $K/10$ communities removed per move. If the number of prune candidates was greater than $K/10$, then $K/10$ communities with the lowest mass were chosen. The lower right portion of Fig. 4.2 shows that our pruning moves can learn close to the true underlying number of clusters (K=56) on a synthetic network even when significantly altering its initial K. Across several real world networks, there was low variance between runs with respect to the final K communities discovered, suggesting a degree of robustness. Furthermore, pruning moves improved perplexity and AUC scores across every dataset as well as reducing computational costs during inference.

4.2.2 The LittleSis Network

The LittleSis network was extracted from the website (http://littlesis.org), which is an organization that acts as a watchdog network to connect the dots between the world's most powerful people and organizations. Our final graph contained 18,831 nodes and 626,881 edges, which represents a relatively sparse graph with edge density of 0.35% (for details on how this dataset was processed see ‡$A.3$). For this analysis, we ran the aHDPR with pruning on the entire dataset using the same settings from our previous experiments. We then took the top 200 degree nodes and generated weighted edges based off of a variational distance between their learned expected variational

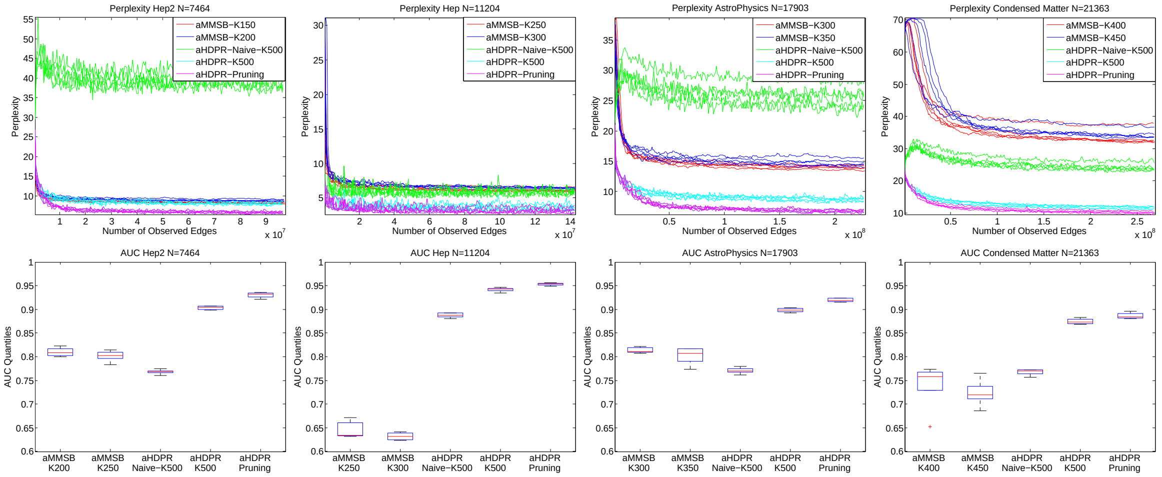


Figure 4.3: The figures above show the best performing model on both perplexity (top) and AUC (bottom) scores to be the aHDPR with pruning moves across four real-world networks.

posteriors such that $d_{ij} = 1 - \frac{|\mathbb{E}_q[\pi_i] - \mathbb{E}_q[\pi_j]|}{2}$. This weighted edge was then included in our visualization software [7] if $d_{ij} > 0.5$. Node sizes were determined by posterior bridgeness [58] where $b_i = 1 - \sqrt{K/(K-1)} \sum_{k=1}^{K} (\mathbb{E}_q[\pi_{ik}] - \frac{1}{K})^2$and measures the extent to which a node is involved with multiple communities. Larger nodes have greater posterior bridgeness while node colors represent its dominant community membership. Our learned latent communities can drive these types of visualizations that otherwise might not have been possible given the raw subgraph (see ‡$A.3$).

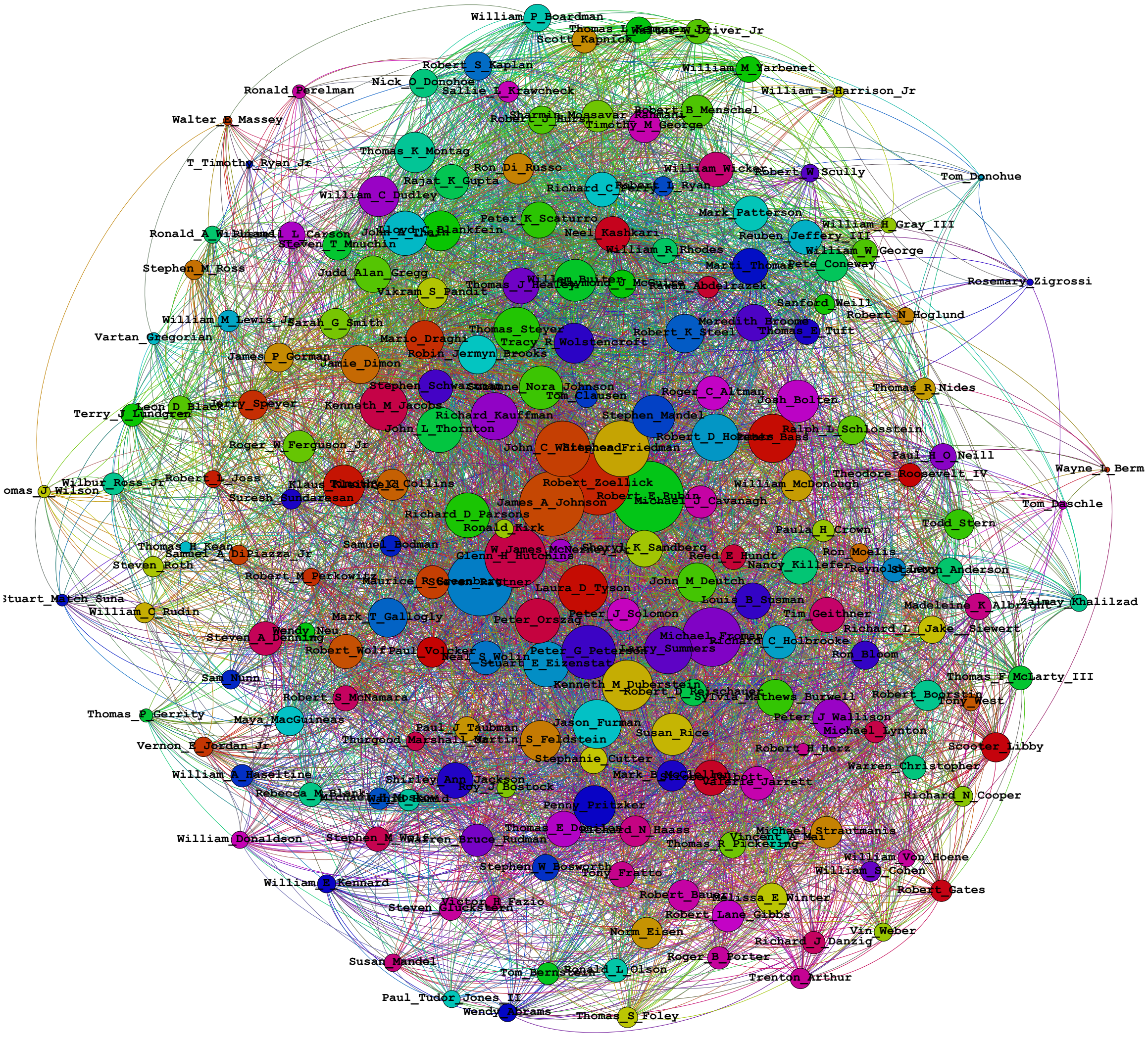


Figure 4.4: The same raw graph of the top 200 degree nodes is displayed using Gephi and the force atlas layout algorithm. Node sizes were determined by its degree and the raw graph represents a cluttered and uninformative view of its underlying structure. We extracted the original graph from its open source database (http://littlesis.org), which was originally a bipartite graph between individuals and the organizations they were involved in. Other types of relationships such as campaign contributions or shared education can also be extracted, but for this study we focused on whether an individual was a member of that organization. We removed individuals and organizations that appeared only once and to generate an undirected network, we assumed an edge existed between people who held positions within the same organization. The largest connected component was found to contain 18,831 nodes and 626,881 edges which we used as our final graph for analysis.

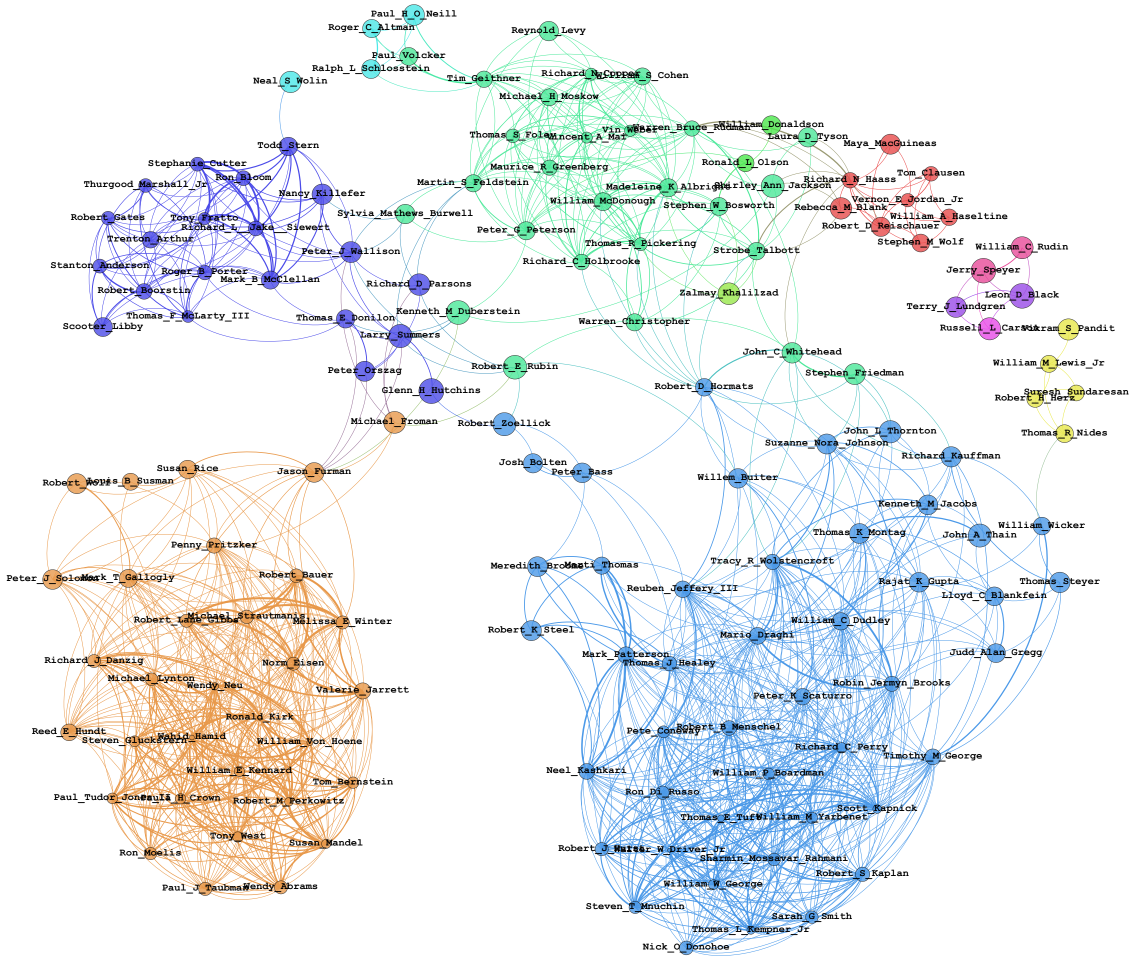


Figure 4.5: **The LittleSis Network**. Near the center in violet we have prominent government figures such as Larry H. Summers (71st US Treasury Secretary) and Robert E. Rubin (70th US Treasury Secretary) with ties to several distinct communities, representative of their high posterior bridgeness. Conversely, within the beige colored community, individuals with small posterior bridgeness such as Wendy Neu can reflect a career that was highly focused in one organization. A quick internet search shows that she is currently the CEO of Hugo Neu, a green-technology firm where she has worked for over 30 years. An analysis on this type of network might provide insights into the structures of power that shape our world and the key individuals that define them.

Chapter 5

Scalable Inference in Bayesian Nonparametric Topic Models

The journey for scalable inference in topic models have taken a variety of approaches. Latent Dirichlet Allocation was first introduced using a variational inference approach, which was seen to be more scalable than traditional MCMC techniques. However, many of the early inferential techniques were prohibited by the need to perform a global update only after all the tokens within the corpus were updated. The next evolution of scalable inference borrows ideas from the stochastic optimization literature. In particular, the idea of approximating our standard objective function with a subset of the data allows us to perform global updates more frequently.

In this chapter, we will develop a variational inference technique known as memoized variational inference that provides significant benefits in model selection and scalability. The original motivation for this starts from research done in the late 90s on incremental EM [57], which posited the idea of updating only a subset of the model parameters before performing a global update. This concept was then applied to the DP mixture model within the variational framework [36] as a memoized variational inference technique. The term memoized comes from the caching of the sufficient statistics used to perform model comparisons so that the inference procedure is truly nonparametric. In this chapter we develop a technique that applied this memoized approach to admixture models, notable topic models with birth/delete moves that help perform model selection.

5.1 HDP Admixture Models

Consider data partitioned into D exchangeable groups $x = \{x_1 \dots x_D\}$, for example documents or images. Each group d contains N_d tokens $x_d = \{x_{d1}, \dots x_{dN_d}\}$, for example words or small pixel patches. For large datasets we divide groups into B predefined batches, where $\mathcal{D}_b$ is the set of

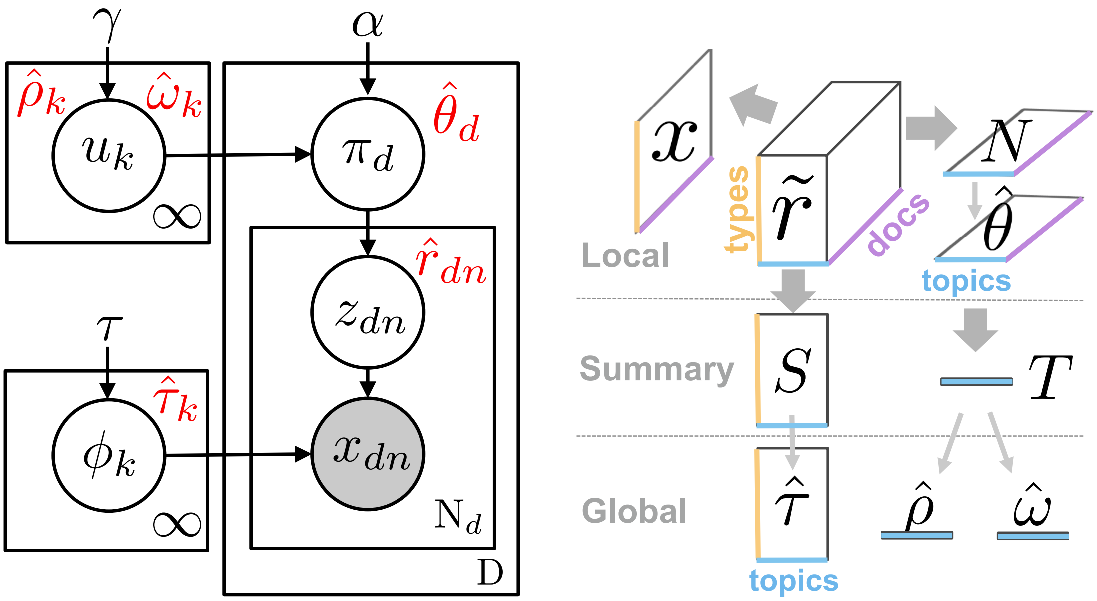


Figure 5.1: *Left:* Directed graphical model for the HDP admixture (Sec. 5.1). Free parameters for mean-field variational inference (Sec. 5.2) shown in red. *Right:* Flow chart for our inference algorithm, specialized for bag-of-words data, where we can use sparse type-based assignments $\tilde r$ instead of per-token variables $\hat r$. We define $\tilde r_{dwk}$ to be the total mass of all tokens in document d of type w assigned to k: $\tilde r_{dwk} = \sum_{n=1}^{N_d} \hat r_{dnk} \delta_{x_{dn},w}$. Updates flow from $\tilde r$ to global topic-type parameters $\hat\tau$ and (separately) to global topic weight parameters $\hat\rho, \hat\omega$. Each variable's shape gives its dimensionality. Thick arrows indicate summary statistics; thin arrows show free parameter updates.

documents in batch b.

To discover themes or topics common to all groups, while capturing group-specific variability in topic usage, we use the HDP admixture model [77] of Fig. 5.1. The HDP uses group-specific frequencies to cluster tokens into an a priori unbounded set of topics. To generate each token, a global topic (indexed by integer k) is first drawn, and an observation is then sampled from the likelihood distribution for topic k.

Topic-specific data generation. HDP admixtures are applicable to any real or discrete data for which an appropriate exponential family likelihood is available. Data assigned to topic k is generated from a distribution F with parameters ϕ_k, and conjugate prior H:

$$\begin{aligned} F: \qquad & \log p(x_{dn}|\phi_k) = s_F(x_{dn})^T \phi_k + c_F(\phi_k), \\ H: \qquad & \log p(\phi_k|\bar\tau) = \phi_k^T \bar\tau + c_H(\bar\tau). \end{aligned}$$

Here c_H and c_F are cumulant functions, and $s_F(x_{dn})$ is a sufficient statistic vector. For discrete data x, F is multinomial and H is Dirichlet. For real-valued x, we take F to be Gaussian and H Normal-Wishart.

Allocating topics to tokens. Each topic k is defined by two global variables: the data-generating exponential family parameters ϕ_k, and a frequency weight u_k. Each scalar $0 < u_k < 1$ defines the conditional probability of sampling topic k, given that the first $k-1$ topics were not sampled:

$$u_k \sim \mathrm{Beta}(1, \gamma), \qquad \beta_k \triangleq u_k \prod_{\ell=1}^{k-1} (1-u_\ell). \tag{5.1}$$

This stick-breaking process [11, 72] transforms $\{u_\ell\}_{\ell=1}^{k}$ to define the marginal probability β_k of selecting topic k.

Each group or document has unique topic frequencies $\pi_d = [\pi_{d1}, \ldots, \pi_{dk}, \ldots]$, where the HDP prior induces a finite Dirichlet distribution on the first K probabilities:

$$[\pi_{d1} \ldots \ \pi_{dK} \ \pi_{d>K}] \sim \mathrm{Dir}(\alpha\beta_1, \ldots \alpha\beta_K, \alpha\beta_{>K}). \tag{5.2}$$

This implies that π_d has mean β and variance determined by the concentration parameter α. The subscript $_{>K}$ denotes the aggregate mass of all topics with indices larger than K, so that $\beta_{>K} \triangleq \sum_{\ell=K+1}^{\infty} \beta_\ell$.

To generate token n in document d, we first draw a topic assignment $z_{dn} \sim \mathrm{Cat}(\pi_d)$, where integer $z_{dn} \in \{1, 2, \ldots\}$ indicates the chosen topic k. Second, we draw the observed token x_{dn} from density F, using the parameter ϕ_k indicated by z_{dn}.

5.2 Variational Inference

Given observed data x, we wish to learn global topic parameters u, ϕ and local document structure π_d, z_d. Taking an optimization approach [80], we seek an approximate distribution q over these variables that is as close as possible to the true, intractable posterior in KL divergence but belongs to a simpler, fully factorized family $q(\cdot) = q(u)q(\phi)q(\pi)q(z)$ of exponential family densities.

Previous variational methods for HDP topic models [83] have employed a Chinese restaurant franchise (CRF) model representation [77]. Here each document has its own local topics, a stick-breaking prior on their frequencies, and latent categorical variables linking each local topic to some global cluster. With this expanded set of highly-coupled latent variables, the factorizations inherent in mean field variational methods induce many local optima. We thus develop an alternative bound based on the direct assignment HDP representation in Fig. 5.1.

5.2.1 Direct Assignment Variational Posteriors

Deferring discussion of the global topic weight posterior $q(u)$ until Sec. 5.2.2, we define other variational posteriors below, marking free parameters with hats to make clear which quantities are optimized:

$$\begin{aligned} q(z|\hat{r}) &= \textstyle\prod_{d=1}^{D} \prod_{n=1}^{N_d} \mathrm{Cat}(z_{dn} \mid \hat{r}_{dn1}, \hat{r}_{dn2}, \ldots \hat{r}_{dnK}), \\ q(\pi) &= \textstyle\prod_{d=1}^{D} \mathrm{Dir}(\pi_d | \hat{\theta}_{d1}, \ldots \hat{\theta}_{dK+1}), \\ q(\phi|\hat{\tau}) &= \textstyle\prod_{k=1}^{\infty} H(\phi_k|\hat{\tau}_k). \end{aligned} \tag{5.3}$$

This posterior models data using K active topics. Crucially, as in Teh et al. [78] and Bryant and Sudderth [17], the chosen truncation level K defines only the form of local factors $q(z)$ and $q(\pi)$. Global factors do not require an explicit truncation, as those with indices greater than K are conditionally independent of the data. This approach allows optimization of K and avoids artifacts that arise with non-nested truncations of stick-breaking processes [11].

Factor $q(z)$. Given truncation level K, token indicator z_{dn} must be assigned to one of the K active topics. The categorical distribution $q(z_{dn})$ is parameterized by a positive vector $\hat{r}_{dn}$ of size K that sums to one.

Factor $q(\pi)$. π_d can be represented by a positive vector of size $K+1$ encoding the K active topic probabilities in document d and (at the last index) the aggregate mass $\pi_{d>K}$ of all inactive topics. Thus, $q(\pi_d)$ is a Dirichlet distribution with parameters $\hat{\theta}_d \in \mathbb{R}^{K+1}$.

Factor $q(\phi)$. Data-generating factors $q(\phi_k)$ for each topic k come from the conjugate family H with free parameter $\hat{\tau}_k$. For discrete data H is Dirichlet and $\hat{\tau}_k$ is a vector the length of the vocabulary W.

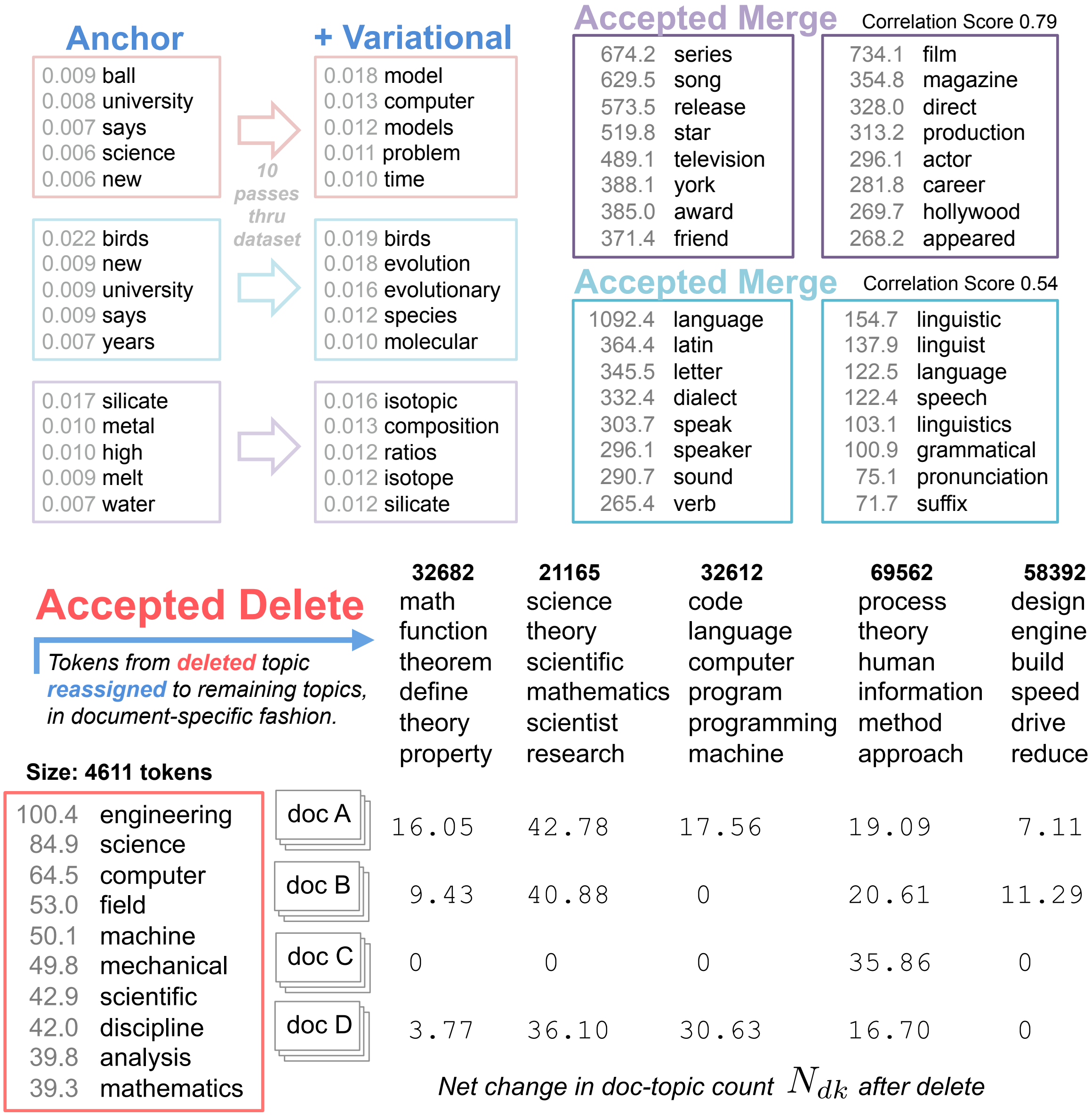
Objective function. Mean field methods optimize an evidence lower bound $\log p(x|\gamma, \alpha, \tau) \geq \mathcal{L}(\cdot)$, where

$$\mathcal{L}(\cdot) \triangleq \mathcal{L}_{data}(\cdot) + H_z(\cdot) + \mathcal{L}_{HDP}(\cdot) + \mathcal{L}_u(\cdot). \tag{5.4}$$

The final term $\mathcal{L}_u(\cdot)$, which depends only on $q(u)$, is discussed in the next section. The first three terms account for data generation, the assignment entropy, and the document-topic allocations. These are defined below, with expectations taken with respect to Eq. (5.3):

$$\begin{aligned} \mathcal{L}_{data}(\cdot) &\triangleq \mathbb{E}_q[\log p(x|z,\phi) + \log \tfrac{p(\phi|\bar{\tau})}{q(\phi|\hat{\tau})}], \\ H_z(\cdot) &\triangleq -\textstyle\sum_{k=1}^{K} \sum_{d=1}^{D} \sum_{n=1}^{N_d} \hat{r}_{dnk} \log \hat{r}_{dnk}, \\ \mathcal{L}_{HDP}(\cdot) &\triangleq \mathbb{E}_q\Big[\log \tfrac{p(z|\pi)p(\pi|\alpha,u)}{q(\pi|\hat{\theta})}\Big]. \end{aligned} \tag{5.5}$$

The forms of $\mathcal{L}_{data}$ and H_z are unchanged from the simpler case of mean-field for DP mixtures. Closed-form expressions are in the Supplement.


Anchor
+ Variational
10 passes thru dataset
Accepted Merge
Accepted Delete
Tokens from deleted topic reassigned to remaining topics, in document-specific fashion.
Size: 4611 tokens
Net change in doc-topic count N_{dk} after delete

Anchor	+ Variational
0.009 ball 0.008 university 0.007 says 0.006 science 0.006 new	0.018 model 0.013 computer 0.012 models 0.011 problem 0.010 time
0.022 birds 0.009 new 0.009 university 0.009 says 0.007 years	0.019 birds 0.018 evolution 0.016 evolutionary 0.012 species 0.010 molecular
0.017 silicate 0.010 metal 0.010 high 0.009 melt 0.007 water	0.016 isotopic 0.013 composition 0.012 ratios 0.012 isotope 0.012 silicate

Accepted Merge — Correlation Score 0.79

674.2	series	734.1	film
629.5	song	354.8	magazine
573.5	release	328.0	direct
519.8	star	313.2	production
489.1	television	296.1	actor
388.1	york	281.8	career
385.0	award	269.7	hollywood
371.4	friend	268.2	appeared

Accepted Merge — Correlation Score 0.54

1092.4	language	154.7	linguistic
364.4	latin	137.9	linguist
345.5	letter	122.5	language
332.4	dialect	122.4	speech
303.7	speak	103.1	linguistics
296.1	speaker	100.9	grammatical
290.7	sound	75.1	pronunciation
265.4	verb	71.7	suffix

100.4	engineering
84.9	science
64.5	computer
53.0	field
50.1	machine
49.8	mechanical
42.9	scientific
42.0	discipline
39.8	analysis
39.3	mathematics

	32682 math function theorem define theory property	21165 science theory scientific mathematics scientist research	32612 code language computer program programming machine	69562 process theory human information method approach	58392 design engine build speed drive reduce
doc A	16.05	42.78	17.56	19.09	7.11
doc B	9.43	40.88	0	20.61	11.29
doc C	0	0	0	35.86	0
doc D	3.77	36.10	30.63	16.70	0

Figure 5.2: *Top Left:* Anchor topics [6] can be improved significantly by variational updates. *Top Right:* Topic pairs accepted by merge moves during run on Wikipedia. Combining each pair into one topic improves our objective $\mathcal{L}$, saves space, and removes redundancy. *Bottom:* Accepted delete move during run on Wikipedia. Red topic is rarely used and lacks semantic focus. Removing it and reassigning its mass to remaining topics improves $\mathcal{L}$ and interpretability.

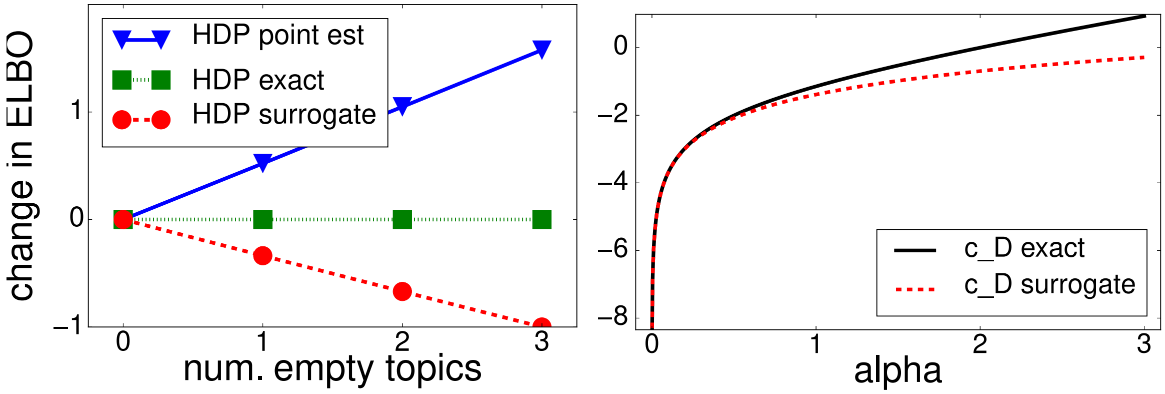


Figure 5.3: *Left:* Comparison of variational objectives resulting from different choices for $q(u)$ on the model selection task of Sec. 5.2.2. Our new surrogate bound sensibly prefers models without empty topics, while using point estimation does not. *Right:* Illustration of Eq. (5.12)'s tight lower bound on $c_D(\alpha\beta)$, shown for $K = 1, \beta = [0.5, 0.5]$. This bound makes our surrogate objective tractable.

5.2.2 Topic Weights and Model Selection

Previous work on the direct assignment HDP suggested a point estimate approximation for topic appearance parameters β [17, 50], or equivalently $q(u_k) = \delta_{\hat{u}_k}(u_k)$. While efficient, this approach creates problems with model selection. The resulting objective lower bounds a joint evidence that includes the point estimate u: $\log p(x, u|\alpha, \gamma, \tau)$. Consequently, the point estimate for u is a MAP estimate, with prior defined by $\mathcal{L}_u$:

$$\mathcal{L}_u^{PE} = \textstyle\sum_{k=1}^{K} \log \mathrm{Beta}(\hat{u}_k|1, \gamma). \tag{5.6}$$

Consider instead a different $q(u)$ that places a proper Beta distribution over each parameter u_k:

$$q(u|\hat{\rho}, \hat{\omega}) = \textstyle\prod_{k=1}^{\infty} \mathrm{Beta}(u_k \mid \hat{\rho}_k\hat{\omega}_k, (1-\hat{\rho}_k)\hat{\omega}_k). \tag{5.7}$$

Here, free parameter $0 < \hat{\rho}_k < 1$ defines the mean: $\mathbb{E}[u_k] = \hat{\rho}_k$, while $\hat{\omega}_k > 0$ controls the variance of u_k. Under this proper Beta family, we can integrate the variable u away to obtain a proper marginal evidence $\log p(x|\alpha, \gamma, \tau)$. Consequently, $\mathcal{L}_u$ term has the form

$$\mathcal{L}_u^{Beta}(\cdot) = \textstyle\sum_{k=1}^{K} \mathbb{E}_q[\log \frac{p(u_k)}{q(u_k)}] \tag{5.8}$$

Model selection. Given our chosen family for $q(z, \pi, \phi)$ in Eq. (5.3) and a proper $q(u)$ in Eq. (5.7), the objective $\mathcal{L}$ can be used to compare two alternative sets of free parameters, even if they have different numbers of active topics K. Our recommended setting of $q(u)$ enjoys the benefits of marginalization, while MAP point estimation can yield pathological behavior when comparing $\mathcal{L}$ at different truncation levels.

To illustrate, consider two candidate models, A and E. Candidate A has K topics and token parameters $\hat{r}^A$. Candidate E_J has the same token parameters as well as J additional topics with zero mass. For each token n, we set vector $\hat{r}^E_n$ so the first K topics are equal to $\hat{r}^A_n$, and the extra J topics are set to zero. We desire an objective that prefers A by penalizing the "empty" topics in E, or at least one that does not favor E.

The behavior of different objectives is shown in Fig. 5.3, where we plot $\mathcal{L}(E_J) - \mathcal{L}(A)$ for $J = \{0, 1, 2, 3\}$ empty topics. When using the Beta form for $q(u)$, we find that exact numerical evaluation of the HDP objective is invariant to empty topics, while our scalable surrogate objective from Sec. 5.2.3 penalizes empty topics slightly. However, point-estimation of $q(u)$ always favors adding empty topics. Thus, we focus on the Beta form of $q(u)$ to learn compact, interpretable models.

5.2.3 Surrogate bound for tractable inference.

Motivated by Fig. 5.3, we wish to employ the proper Beta form for $q(u)$. However, this leads to a non-conjugate relationship between $q(u)$ and $q(\pi)$, complicating inference. Some terms of the resulting objective have no closed-form. To gain tractability, we develop a surrogate bound on the ideal objective.

Consider the ELBO term $\mathcal{L}_{HDP}$ under $q(u)$ in Eq. (5.7).

$$\begin{aligned}\mathbb{E}_q[\log \tfrac{p(z)p(\pi)}{q(\pi)}] &= \textstyle\sum_{d=1}^{D} \mathbb{E}_q[c_D(\alpha\beta)] - c_D(\hat{\theta}_d) \\ &+ \textstyle\sum_{k=1}^{K+1} \Big(N_{dk} + \alpha \mathbb{E}_q[\beta_k] - \hat{\theta}_{dk} \Big) \mathbb{E}_q[\log \pi_{dk}]\end{aligned} \tag{5.9}$$

Here, sufficient statistic N_{dk} counts the usage of topic k in document d: $N_{dk} \triangleq \sum_{n=1}^{N_d} \hat{r}_{dnk}$. Furthermore, two required expectations have closed-form expressions. $\mathbb{E}[\beta_k]$ comes from Eq. (5.1), and

$$\mathbb{E}[\log \pi_{dk}] = \psi(\hat{\theta}_{dk}) - \psi(\textstyle\sum_{\ell=1}^{K+1} \hat{\theta}_{d\ell}). \tag{5.10}$$

However, c_D is the cumulant function of the Dirichlet,

$$c_D(a_1, \dots a_W) = \log \frac{\Gamma(\sum_{w=1}^{W} a_w)}{\prod_{w=1}^{W} \Gamma(a_w)}, \tag{5.11}$$

and $\mathbb{E}_q[c_D(\alpha\beta)]$ has no closed form. To avoid this problematic expectation of log Gamma functions, we introduce a novel bound on $c_D(\cdot)$:

$$\begin{aligned}c_D(\alpha\beta) &\geq K \log \alpha + \textstyle\sum_{k=1}^{K} \log u_k \\ &+ \textstyle\sum_{k=1}^{K} (K{+}1{-}k) \log 1{-}u_k.\end{aligned} \tag{5.12}$$

Fig. 5.3 shows this bound is valid for all $\alpha > 0$. For proof, see the Supplement. We can tractably compute the expectation of Eq. (5.12), because expectations of logs of Beta random variables have a closed form.

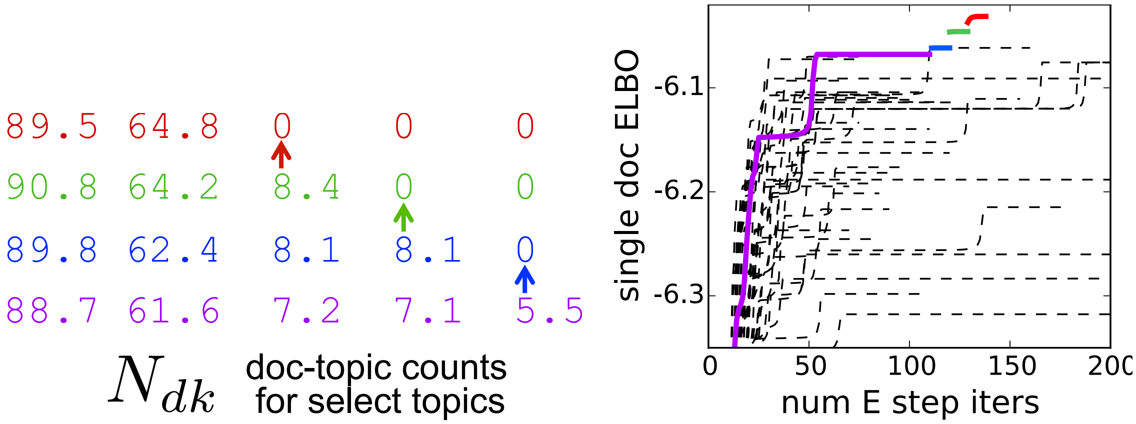


Figure 5.4: Sparsity-promoting restarts for local steps on the Science corpus with $K = 100$. *Left:* Example fixed points of the topic usage statistic N_{dk} for one document. *Right:* Trace of single-document ELBO objective during E-step inference for 50 random initializations (dashed lines), plus one sparsity-promoting run (solid) which climbs through the color-coded fixed points in the adjacent plot.

Substituting Eq. (5.12) into our original objective $\mathcal{L}$ yields a surrogate objective $\mathcal{L}_{sur}$ which can be used for model selection because it remains a valid lower bound on the log evidence $\log p(x|\alpha, \gamma, \bar{\tau})$. Our surrogate objective induces a small penalty for empty components in Fig. 5.3, which is superior to the reward for empty components induced by point estimates.

5.3 Inference Algorithm

We now describe an algorithm for optimizing the free parameters of our chosen approximation family q. We first give concrete updates to local and global factors. Later, we introduce memoized and stochastic methods for scalable online learning.

5.3.1 Local updates.

In the local step, we visit each document d and update token indicators r_{dn} via Eq. (5.13) and document-topic parameters $\hat{\theta}_d$ via Eq. (5.14). These steps are inter-dependent: updating $\hat{r}_{dn}$ requires an expectation computed from $\hat{\theta}_d$, and vice versa. Thus, at each document we need to initialize $\hat{\theta}_d$ and then alternate these updates until convergence. We discuss initialization and convergence strategies in the Supplement.

Update of $q(z)$. We update the free parameter $\hat{r}_{dn}$ for each token n in document d according to

$$\hat{r}_{dnk} \propto \exp\Big(\mathbb{E}_q[\log \pi_{dk}] + \mathbb{E}_q[\log p(x_{dn}|\phi_k)]\Big), \tag{5.13}$$

which uses known expectations. The vector $\hat{r}_{dn}$ is normalized over all topics k so its sum is one.

Update of $q(\pi_d)$. We update free parameter $\hat{\theta}_d$ given N_{dk}, which summarizes usage of topic k across all tokens in document d. The update is

$$\hat{\theta}_{dk} = \alpha \mathbb{E}_q[\beta_k] + N_{dk}, \tag{5.14}$$

where the expectation $\mathbb{E}_q[\beta_k]$ follows from Eq. (5.1). This update applies to all $K+1$ entries of $\hat{\theta}_d$. The last index aggregates all inactive topics, and is simply set to $\alpha \mathbb{E}[\beta_{>K}]$, since $N_{d>K}$ is zero by truncation.

Sparse Restarts. When visiting document d, the joint inference of $\hat{\theta}$ and $\hat{r}$ can be challenging. Many local optima exist even for this single-document task, as shown Fig. 5.4. A common failure mode occurs when a few tokens are assigned to a rare "junk" topic. Reassignment of these tokens may not happen under Eq. (5.13) updates due to a valley in the objective between keeping the current junk assignments and setting the junk topic to zero.

To more adequately escape local optima, we develop sparsity-promoting restart moves which take a final document-topic count vector $[N_{d1} \dots N_{dK}]$ produced by coordinate ascent, propose an alternative which has one entry set to zero, and accept if this improves the ELBO after further ascent steps. In practice, the acceptance rate varies from 30-50% when trying the 5 smallest non-zero topics. We observe huge gains in the whole-dataset objective due to these restarts.

5.3.2 Global updates.

Fig. 5.1 shows global parameter updates to $\hat{\tau}, \hat{\rho}$, and $\hat{\omega}$ require compact sufficient statistics of local parameters. The updates below focus on these summaries.

Update for $q(\phi)$. We update free parameter $\hat{\tau}$ to

$$\hat{\tau}_k = S_k + \bar{\tau}, \qquad S_k \triangleq \textstyle\sum_{d=1}^{D} \sum_n s_F(x_{dnk}) \hat{r}_{dnk}, \tag{5.15}$$

where S_k is the statistic summarizing data assigned to topic k across all tokens. For topic models, S_k is a vector of counts for each vocabulary type.

Update for $q(u)$. Finally, we consider the free parameters $\hat{\rho}, \hat{\omega}$ for all K active topics. No closed-form update exists due to non-conjugacy. Instead, we numerically optimize our surrogate objective, finding the best vectors $\hat{\rho}, \hat{\omega}$ simultaneously. The constrained optimization problem is:

$$\begin{aligned} \hat{\rho}, \hat{\omega} = \operatorname{argmax}_{\rho,\omega} \mathcal{L}_{HDP}(\rho, \omega, T, \alpha) + \mathcal{L}_u(\rho, \omega, \gamma) \\ \text{s.t.} \quad 0 < \rho_k < 1, \omega_k > 0 \text{ for } k \in \{1, 2, \dots K\} \end{aligned} \tag{5.16}$$

where sufficient statistic $T = [T_1 \dots T_K \; T_{K+1}]$ sums the expectation of Eq. (5.10) across documents:

$$T_k(\hat{\theta}) \triangleq \textstyle\sum_{d=1}^{D} \mathbb{E}[\log \pi_{dk}]. \tag{5.17}$$

The Supplement provides implementation details, including the exact function and gradients we provide to a modern L-BFGS optimization algorithm.

5.3.3 Memoized algorithm.

The memoized variational inference approach is an alternative "online" approach to deterministic inference in our models. The benefit over stochastic variational inference is the freedom from having to tune any learning rate parameters. The only parameters necessary are predefined mini-batch sizes and memory for only the sufficient statistics across all mini-batches. The memoized approach is a generalization of previous work on incremental variants of the EM algorithm [57]. Our proposal is to apply the memoized approach originally developed by [38] for a standard DP mixture model to an HDP model. To motivate this line of work, we propose a preliminary procedure for a memoized variational inference approach for the aHDPR.

We now provide a memoized coordinate ascent update algorithm. The update cycle comes from [36], which was inspired by the incremental EM approach of [57]. Data is visited one batch at a time, where the batches are predefined. We call each complete pass through all batches a lap. At each batch, we perform a local step update to $q(z_d), q(\pi_d)$ for each document d in the batch, and then a global-step update to $q(u), q(\phi)$.

Affordable batch-by-batch processing is possible by tracking sufficient statistics and exploiting their additivity. For each statistic, we track a batch-specific quantity (denoted N^b) for each batch and an aggregated whole-dataset quantity (N). By definition, $N_k = \sum_{b=1}^{B} N_k^b$. After visiting each batch b, we perform an incremental update to make the aggregate summaries reflect the new batch summaries and remove any previous contribution from batch b.

This algorithm requires storing per-batch summaries N^b, S^b, T^b for every batch during inference. This requirement is modest, remaining size $\mathcal{O}(BK)$ no matter how many tokens or documents occur in each batch.

ELBO computation. Computing the objective $\mathcal{L}$ is possible after each batch visit, so long as we track sufficient statistics as well as a few ELBO-specific quantities. First, we store the entropy H_z from Eq. (5.5) at each batch, as in [36].

Second, consider the computation of $\mathcal{L}_{HDP}$ in Eq. (5.9). Naively, this computation requires sums over all documents. However, by tracking the following terms we can perform rapid evaluation:

$$\begin{aligned} G_k^b &\triangleq \textstyle\sum_{d \in \mathcal{D}_b} (N_{dk} - \hat{\theta}_{dk}) \mathbb{E}[\log \pi_{dk}], \\ Q_0^b &= \textstyle\sum_{d \in \mathcal{D}_b} \log \Gamma(\sum_{k=1}^{K+1} \hat{\theta}_{dk}), Q_k^b = \sum_{d \in \mathcal{D}_b} \log \Gamma(\hat{\theta}_{dk}). \end{aligned} \tag{5.18}$$

After aggregating these tracked statistics across all batches, such as $Q_k = \sum_{b=1}^{B} Q_k^b$, Eq. (5.9) becomes

$$\begin{aligned} \mathcal{L}_{HDP}(\cdot) = D\mathbb{E}_q[c_D(\alpha\beta)] - Q_0 \\ + \textstyle\sum_{k=1}^{K+1} Q_k + G_k + \alpha\mathbb{E}_q[\beta_k]T_k \end{aligned} \tag{5.19}$$

which given tracked statistics can be evaluated with cost independent of the number of documents D.

5.3.4 Stochastic algorithm.

Our objective $\mathcal{L}$ can also be optimized with stochastic variational inference [33]. The stochastic global step at iteration t updates the natural parameters of $q(u)$ and $q(\phi)$ with learning rate ξ_t. For example, the new $\hat{\tau}_t$ interpolates between the previous value $\hat{\tau}_{t-1}$ and an amplified estimate from the current batch $\hat{\tau}^b$. When ξ_t decays appropriately, this method guarantees convergence to a local optimum.

5.3.5 Computational complexity

Our direct assignment representation is more efficient than the CRF approach of [83]. The dominant cost of both algorithms is the local step for each token. We require $\mathcal{O}(N_d K)$ computations to update the free parameters $\hat{r}$ for a single document via Eq. (5.13). The CRF method requires $\mathcal{O}(N_d K J)$ operations, where $J < K$ is the number of global topics allowed in each document (for more details, see Eq. 18 of [83]). For any reasonable value of $J > 1$, the CRF approach is more expensive. When $J = \mathcal{O}(K)$, the CRF local step is quadratic in the number of topics, while our approach is always linear.

5.4 Merge and Delete Moves

Here, we develop two moves, merge and delete, which help discover a compact set of interpretable topics. As illustrated in Fig. 5.2, merges combine redundant topics, while deletes remove unnecessary "junk" topics or empty topics. Both moves enable faster subsequent iterations by making the active set of topics smaller.

5.4.1 Merge moves.

Each merge move transforms a current variational posterior q of size K into a candidate q' of size $K - 1$ by combining two topics in a single merged topic. During each pass we consider several candidate pairs. For each pair $\ell < m$, we imagine simply pooling together all tokens assigned to either topic ℓ or m in the original model to create topic ℓ in q'. All other parameters are copied over unchanged. Formally,

$$\hat{r}'_{dn\ell} = \hat{r}_{dn\ell} + \hat{r}_{dnm}, \forall d, n, \; \hat{\theta}'_{d\ell} = \hat{\theta}_{d\ell} + \hat{\theta}_{dm}, \forall d. \tag{5.20}$$

A global update to create $\hat{\tau}', \hat{\rho}', \hat{\omega}'$ completes the candidate, and we keep it if the objective $\mathcal{L}$ improves.

For large datasets, explicitly retaining both $\hat{r}$ and $\hat{r}'$ via Eq. (5.20) is prohibitive. Instead, we can exploit additive statistics to rapidly evaluate a proposed merge. Eq. (5.20) implies that $S'_\ell = S_\ell + S_m$ and $N'_\ell = N_\ell + N_m$. This allows constructing candidate $\hat{\tau}'$ values and evaluating $\mathcal{L}_{data}$ without visiting any batches.

Not all statistics can be computed in this way, so some modest tracking must occur. For each candidate merge, we must compute T'^{b}_{ℓ} from Eq. (5.17) as well as the ELBO statistics $G'^{b}_{\ell}, Q'^{b}_{\ell}$ from Eq. (5.18) at each batch. Finally, we track the entropy H_z for each candidate, as did [36].

The first step of a merge is to select candidate pairs using a correlation score [17]:

$$\text{score}(\ell, m) = \text{Corr}(N_{:\ell}, N_{:m}), \quad -1 < \text{score} < 1. \tag{5.21}$$

Large scores identify topic pairs frequently used in the same documents. Before each lap we select at most 50 pairs to track with score above 0.05.

Next, we visit each batch in order, tracking relevant merge summaries during standard memoized updates. Finally, we evaluate each candidate using both tracked summaries and additive summaries, accepting or rejecting as needed. Many merges can be accepted after each lap, so long as no two share a topic in common.

5.4.2 Delete moves

Delete moves provide a more powerful alternative to merges for removing rarely used "junk" topics. For an illustration of an accepted delete on Wikipedia data, see Fig. 5.2. After identifying a candidate topic with small mass to delete, we reassign all its tokens to the remaining topics and then accept if the objective $\mathcal{L}$ improves. This move can succeed when a merge would fail because each document's tokens can be reassigned in a customized way, as shown in Fig. 5.2.

To make this move scalable for our memoized algorithm, we identify a candidate delete topic j in advance and collect a target dataset x' of all documents which use selected topic j significantly: $\{d : N_{dj} > 0.01\}$. Given the target set, we initialize candidate sufficient statistics by simply removing entries associated with topic j. From this initialization, we run several local-global updates on the target and then accept the move if the target's variational objective $\mathcal{L}(\cdot)$ improves. Further details can be found in the Supplement. To be sure of deleting a topic, the target set x' must contain all documents which pass our threshold test. Thus, deletes are only applicable to topics of below some critical size to remain affordable. We set a maximum budget of 500 documents for the target dataset size in our topic modeling experiments.

Acceptance rates in practice. Here, we summarize acceptance rates for merges and deletes during a typical run on the Wikipedia dataset with $K = 200$ initial topics. During the first 4 passes, we accept 73 of 79 proposed deletes (92%), and 12 of 194 merges (6%). These moves crucially remove bad topics from the random initialization. After the first few laps, no further merges are accepted and only 10% of deletes are accepted (at most 1 or 2 attempts per lap).

5.5 Experiments

Our experiments compare inference methods for fitting HDP topic models. For our new HDP objective, we study stochastic with fixed K (SOfix), memoized with fixed K (MOfix), and memoized with deletes and merges (MOdm). For baselines, we consider the collapsed sampler (Gibbs) of [77], the stochastic CRF method (crfSOfix) of [83], and the stochastic split-merge method (SOsm) of [17]. For each method, we perform several runs from various initial K values.

For each run, we measure its predictive power via a heldout document completion task, as in [17]. Each model is summarized by a point-estimate of the topic-word probabilities ϕ. For each heldout document d we randomly split its word tokens into two halves: x'_d, x''_d. We use the first half to infer a point-estimate of π_d, then estimate log-likelihood of each token in the second half x''_d.

$$\text{heldout-lik}(x|\phi) = \frac{\sum_{d \in \mathcal{D}_{test}} \log p(x''_d|\pi_d, \phi)}{\sum_{d \in \mathcal{D}_{test}} |x''_d|} \tag{5.22}$$

Hyperparameters. In all runs, we set $\gamma = 10$, $\alpha = 0.5$ and topic-word pseudocount $\bar{\tau} = 0.1$. Stochastic runs use the learning rate decay recommended in [17]: $\kappa = 0.5, \delta = 1$.

5.5.1 Toy bars dataset.

We study a variant of the toy bars dataset of [28], shown in Fig. 5.5. There are 10 ideal bar topics, 5 horizontal and 5 vertical. The bars are noisier than the original and cover a larger vocabulary (900 words). We generate 1000 documents for training and 100 more for heldout test. Each one has 200 tokens drawn from 1-3 topics.

Fig. 5.5 shows many runs of all algorithms on this benchmark. Variational methods initialized with 50 or 100 topics get stuck rapidly, while the Gibbs sampler finds a redundant set of the ideal topics and is unable to effectively merge down to the ideal 10.

In contrast, our MOdm method uses merges and deletes to rapidly recover the 10 ideal bars after only a few laps. Without these moves, MOfix runs remain stuck at suboptimal fragments of

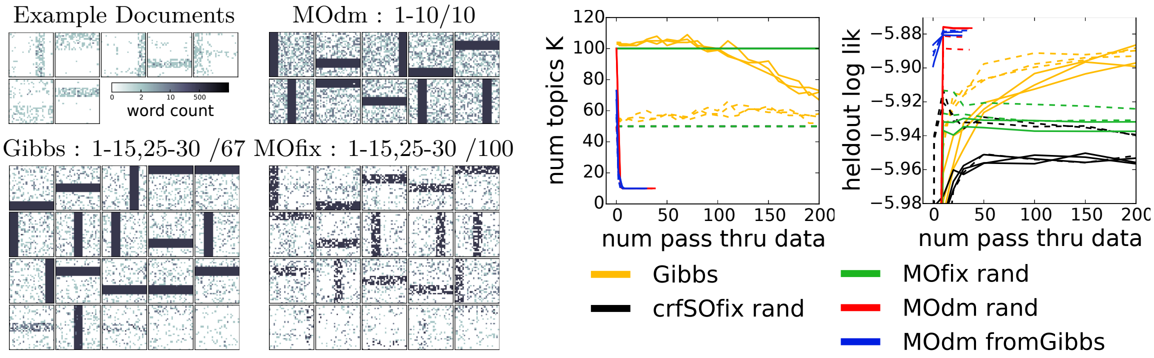


Figure 5.5: Comparison of inference methods on toy bars dataset from Sec. 5.5.1. *Top Left:* Word count images for 7 example documents and the final 10 estimated topics from MOdm. Each image shows all 900 vocabulary types arranged in square grid. *Bottom left:* Final estimated topics from Gibbs and MOfix. We rank topics from most to least probable, and show ranks 1-15 and 25-30. *Right:* Trace plots of the number of topics K and heldout likelihood during training. Line style indicates number of initial topics: dashed is $K = 50$, solid is $K = 100$.

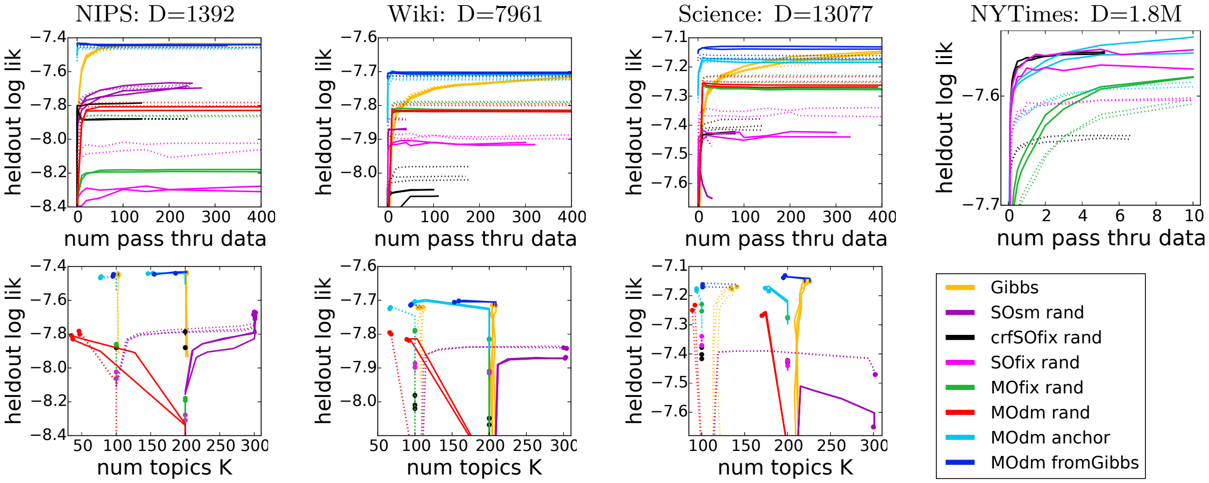


Figure 5.6: Comparison of inference methods on academic and news article datasets (Sec. 5.5.2). Line style indicates initial number of topics K: 100 is dots, 200 is solid. *Top row:* Heldout likelihood (larger is better) as more training data is seen. *Bottom row:* Trace plots of heldout likelihood and number of topics. Each solid dot marks the final result of a single run, with the trailing line its trajectory from initialization. Ideal runs move toward the upper left corner.

bars. Furthermore, our MOdm method initialized with the sampler's final topics (fromGibbs) easily recovers the ideal bars.

5.5.2 Academic and news articles.

Next, we apply all methods to papers from the NIPS conference, articles from Wikipedia, and articles from the journal Science [63], with 80%-20% train-test splits. Online methods process each training set in 20 batches. Trace plots in Fig. 5.6 compare predictive power and model complexity as more data is processed. We summarize conclusions below.

Anchor topics are good; variational is better. Using the anchor word method [6] for initial topic-word parameters yields better predictions than random initialization (`rand`). However, our methods can still make big, useful changes from this starting point. See Fig. 5.2 for some examples.

Deletes and merges make big, useful changes. Across all 3 datasets in Fig. 5.6, merges and deletes remove many topics. On Wikipedia, we reduce 200 topics to under 100 while improving predictions. Similar gains occur from the final result of the Gibbs sampler.

Competitors get stuck or improve slowly. The Gibbs sampler needs many laps to make quality predictions. The CRF method gets stuck quickly, while our methods (using the direct assignment representation) do better from similar initializations. The stochastic split-merge method (SOsm) grows to a prescribed maximum number of topics but fails to make better predictions. This indicates problems with heuristic acceptance rules, and motivates our moves governed by exact evaluation of a whole-dataset objective.

Next, we analyze the New York Times Annotated Corpus: 1.8 million articles from 1987 to 2007. We withhold 800 documents and divide the remainder into 200 batches (9084 documents per batch). Fig. 5.6 shows the predictive performance of the more-scalable methods.

For this large-scale task, our direct assignment representation is more efficient than the CRF code released by [83]. With $K = 200$ topics, our memoized algorithm with merge and delete moves (MOdm) completes 8 laps through the 1.8 million documents in the amount of time the CRF code completes a single lap. No deletes or merges are accepted from any MOdm run, likely because 1.8M documents require more than a few hundred topics. However, the acceptance rate of sparsity-promoting restarts is 75%. With a more efficient, parallelized implementation, we believe our variational approach will enable reliable large-scale learning of topic models with larger K.

5.5.3 Image patch modeling.

Finally, we study 8×8 patches from grayscale natural images as in [85]. We train on 3.5 million patches from 400 images, comparing HDP admixtures to Dirichlet process (DP) mixtures using a zero-mean Gaussian likelihood. The HDP model captures within-image patch similarity via image-specific mixture component frequencies. Both methods are evaluated on 50 heldout images scored via Eq. (5.22).

Fig. 5.7 shows merges and deletes removing junk topics while improving predictions, justifying the generality of these moves. Further, the HDP earns better prediction scores than the DP mixture.

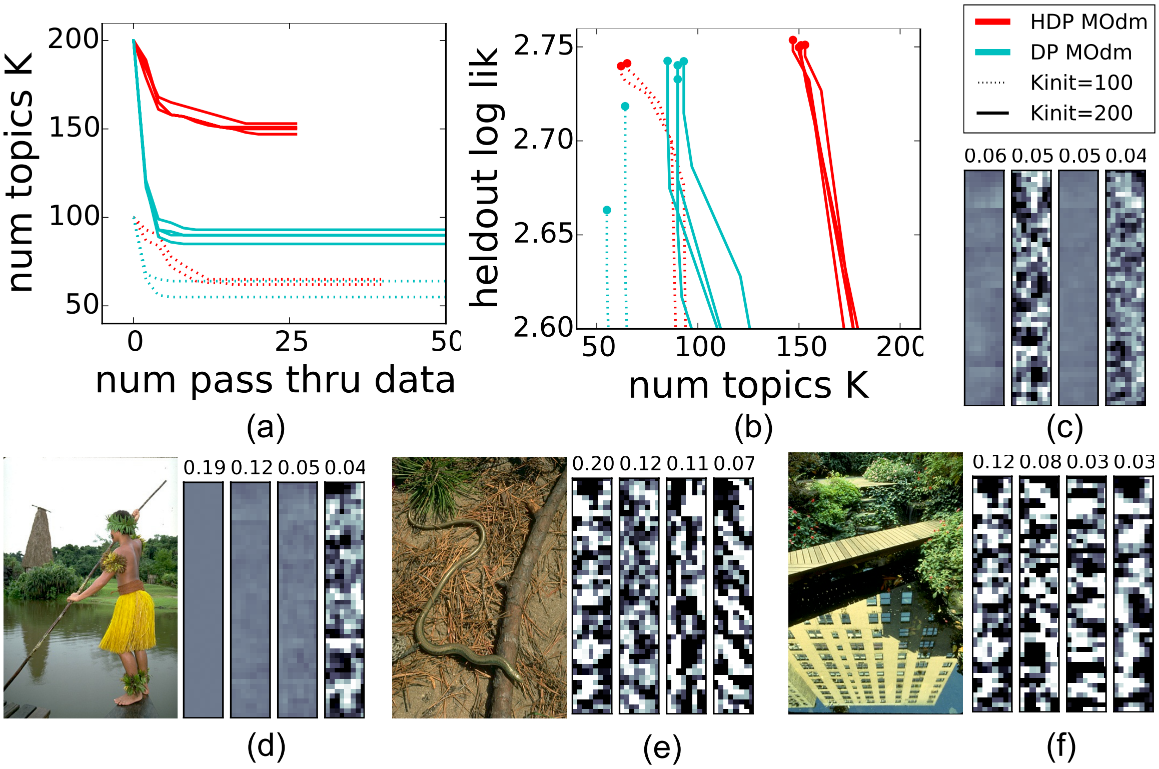


Figure 5.7: Comparison of DP mixtures and HDP admixtures on 3.5M image patches (Sec. 5.5.3). (a-b) Trace plots of number of topics and heldout likelihood, as in Fig 5.6. (c) Patches from the top 4 estimated DP clusters. Each column shows 6 stacked 8×8 patches sampled from one cluster. (d-f) Patches from 4 top-ranked HDP clusters for select test images from BSDS500 [5].

We illustrate this success by plotting sample patches from the top 4 topics (ranked by topic weight π) for several heldout images. The HDP adapts topic weights to each image, favoring smooth patches for some images (d) and textured patches for others (e-f). The less-flexible DP must use the same weights for all images (c).

Chapter 6

Refinery - Topic Modeling for the Masses

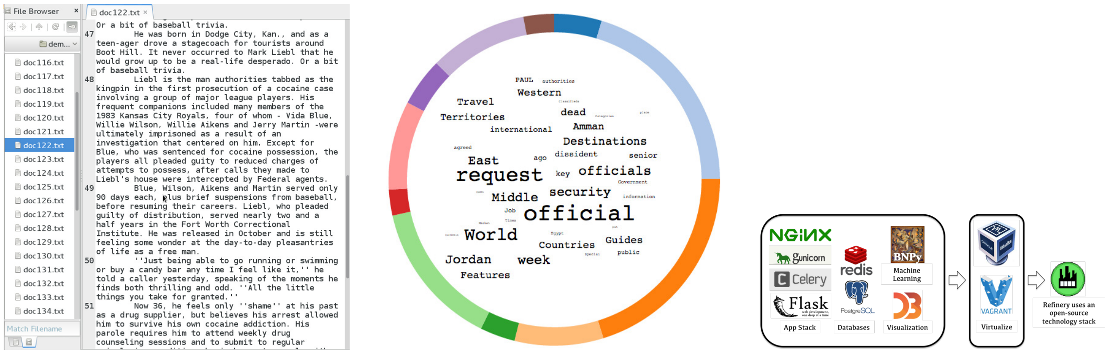
6.1 Refinery: A web platform for topic modeling

Topic models have become a ubiquitous class of tools for the analysis of large document collections [10]. However, there is still a significant adoption barrier for individuals who have little to no background in coding or computer science. For example, journalists could potentially use learned topics to organize large datasets generated from Freedom of Information Act (FOIA) requests, but few have the expertise to implement necessary code nor an intuition for how to fine tune required parameters. Beyond the challenge of running a topic modeling algorithm, there is the added difficulty of interpreting the results.

To make these types of models and their results more accessible, we built an open source web application called Refinery. Originally motivated to help journalists, Refinery allows any professional to explore a large corpus and identify a small set of relevant documents for careful study. Example use cases include a legal firm handed an enormous document dump or a librarian exploring a historical archive. In these cases and many others, Refinery makes learning algorithms and visualizations accessible and easy-to-use.

6.1.1 Running Refinery

Refinery is an in-browser web application that builds on many existing open-source projects for data storage, virtualization, and machine learning, as illustrated in Fig. 6.1. To make installation as simple as possible, it has only three dependencies: the Git version-control system, Virtualbox [60], and Vagrant [30]. At a Unix-like command line with these dependencies already installed, the



Or a bit of baseball trivia.

47 He was born in Dodge City, Kan., and as a teen-ager drove a stagecoach for tourists around Boot Hill. It never occurred to Mark Liebl that he would grow up to be a real-life desperado. Or a bit of baseball trivia.

48 Liebl is the man authorities tabbed as the kingpin in the first prosecution of a cocaine case involving a group of major league players. His frequent companions included many members of the 1983 Kansas City Royals, four of whom - Vida Blue, Willie Wilson, Willie Aikens and Jerry Martin -were ultimately imprisoned as a result of an investigation that centered on him. Except for Blue, who was sentenced for cocaine possession, the players all pleaded guity to reduced charges of attempts to possess, after calls they made to Liebl's house were intercepted by Federal agents.

49 Blue, Wilson, Aikens and Martin served only 90 days each, plus brief suspensions from baseball, before resuming their careers. Liebl, who pleaded guilty of distribution, served nearly two and a half years in the Fort Worth Correctional Institute. He was released in October and is still feeling some wonder at the day-to-day pleasantries of life as a free man.

50 ''Just being able to go running or swimming or buy a candy bar any time I feel like it,'' he told a caller yesterday, speaking of the moments he finds both thrilling and odd. ''All the little things you take for granted.''

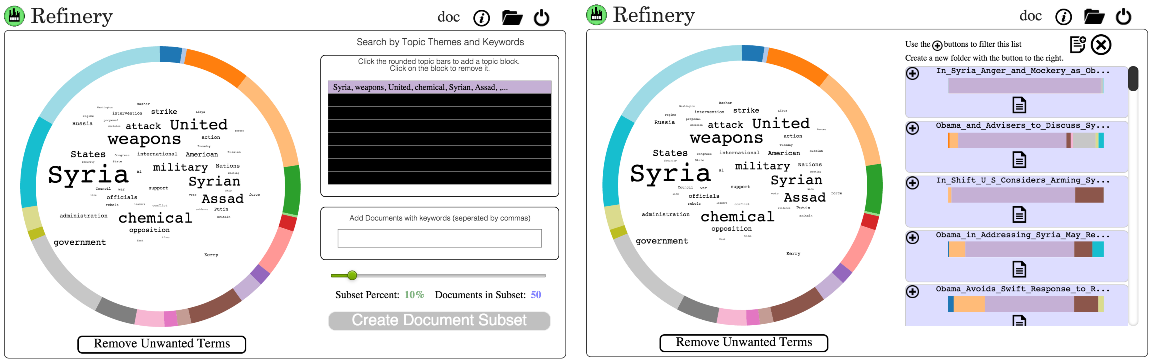
51 Now 36, he feels only ''shame'' at his past as a drug supplier, but believes his arrest allowed him to survive his own cocaine addiction. His parole requires him to attend weekly drug counseling sessions and to submit to regular

(a) *Input:* Plain-text files, one per document

(b) *Output:* Interactive visualization of learned topics

(c) Technology stack

Figure 6.1: Refinery is a web-application that allows users to upload plain-text documents (a) and visualize the semantic themes learned by a topic-model (b). Under the hood, Refinery uses many existing technologies (c) for browser interaction, data management, machine learning, and visualization. A video demo is available online: http://youtu.be/7yRQ1J9Z_LI. TODO: standardize size of third figure.



Syria, weapons, United, chemical, Syrian, Assad, ,...

Subset Percent: 10% Documents in Subset: 50

Use the ⊕ buttons to filter this list
Create a new folder with the button to the right.

⊕ In_Syria_Anger_and_Mockery_as_Ob...

⊕ Obama_and_Advisers_to_Discuss_Sy...

⊕ In_Shift_U_S_Considers_Arming_Sy...

⊕ Obama_in_Addressing_Syria_May_Re...

⊕ Obama_Avoids_Swift_Response_to_R...

Figure 6.2: The results of an analysis on 500 New York Times articles that contained the keyword “obama” during the year 2013. In the figure above, the “Syria” topic has been selected along with a document subset of 50 documents related to this topic.

complete Refinery installation requires just a few lines of code:

```
> git clone https://github.com/daeilkim/refinery.git
> vagrant init ubuntu/trusty64 # create virtual machine
> vagrant up # launch machine and install required Python packages
```

6.1.2 A UI pipeline for Topic Modeling

Uploading documents. Users submit documents for analysis in the form of a zip file, which uncompressed contains a folder of plain-text files, one for each individual document. Users do not need to preprocess documents or identify a vocabulary in advance. This is done automatically by the software where every word is tokenized as a unigram. The final vocabulary terms are thresholded by the constraint that they not appear in more than 80% of the text, or appear in less than 2 documents.

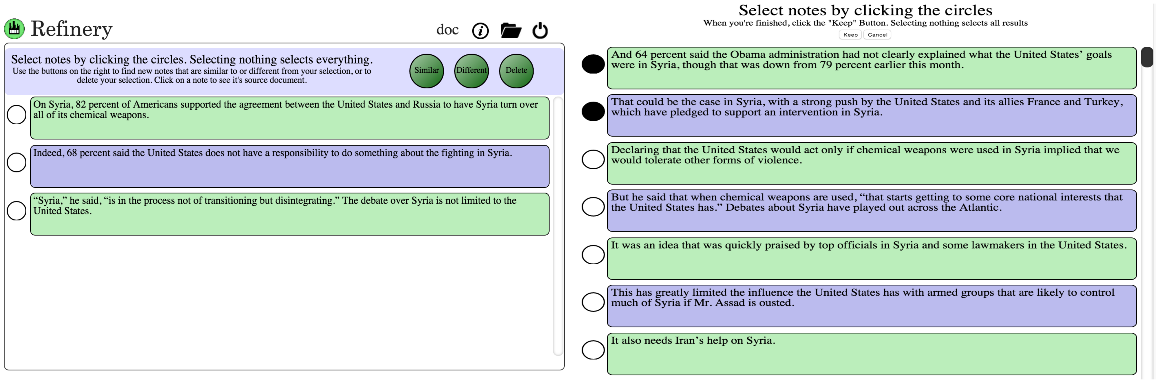
Training a topic model. The topic modeling algorithm used for Refinery comes from the BNPy toolbox [35]. Specifically, we train a hierarchical Dirichlet process (HDP) topic model [77] using a memoized variational inference algorithm [39]. This optimization algorithm can dynamically add or remove topics as it sees more data, guided by an objective function that minimizes a lower bound on the log probability of the data. While the underlying algorithm has many free parameters, we set most of these to smart defaults and only ask the user to suggest an initial number of topics, which the algorithm can adapt as the data suggests. Support for configuring HDP hyperparameters will be added to the UI in future updates, but instructions for a manual modification can currently be found in the BNPy documentation [35].

Browseable visualization. After training a model, users can explore the resulting topics interactively via a multi-colored ring, as shown in Fig. 6.2. Each segment of the ring represents one topic, with its size indicative of the topic's frequency in the overall corpus. Hovering the cursor over a topic's segment shows the top 50 words associated with that topic. In this word cloud visualization, text size is scaled according to the topic-specific word frequency.

Focused exploration of document subsets. Often, analysis of large collections requires identifying subsets of relevant documents and performing more detailed clustering of that subset. Refinery supports this by allowing the main corpus of documents to be filtered by keyword and topic presence. Refinery provides two methods for selecting relevant documents once a topic model over K topics has been learned. The first allows the user to specify a distribution over topics, directly composing a K dimensional query. The second composes a K dimensional query from a list of search terms, averaging the probability of topic given word for each term and normalizing over the K dimensions. These two topic queries are averaged, and documents are ranked based on the KL divergence between their MAP topic distribution and the query.

6.2 Phrase Extraction and Refinement

While topic modeling enables scalable first-pass analysis, often the underlying bag-of-words assumptions are too limiting. After filtering a large corpus to find only documents relevant to a small set of topics, Refinery further allows users to explore documents by phrase similarity and create a summary of the corpus composed of sentences extracted from the documents' text. Here, we use the open-source implementation of the Splitta algorithm [25], which [24] introduced for sentence boundary detection. First, the Splitta algorithm is applied for sentence boundary detection. This



Select notes by clicking the circles. Selecting nothing selects everything.
Use the buttons on the right to find new notes that are similar to or different from your selection, or to delete your selection. Click on a note to see it's source document.

○ On Syria, 82 percent of Americans supported the agreement between the United States and Russia to have Syria turn over all of its chemical weapons.

○ Indeed, 68 percent said the United States does not have a responsibility to do something about the fighting in Syria.

○ “Syria,” he said, “is in the process not of transitioning but disintegrating.” The debate over Syria is not limited to the United States.

Select notes by clicking the circles
When you're finished, click the "Keep" Button. Selecting nothing selects all results

● And 64 percent said the Obama administration had not clearly explained what the United States’ goals were in Syria, though that was down from 79 percent earlier this month.

● That could be the case in Syria, with a strong push by the United States and its allies France and Turkey, which have pledged to support an intervention in Syria.

○ Declaring that the United States would act only if chemical weapons were used in Syria implied that we would tolerate other forms of violence.

○ But he said that when chemical weapons are used, “that starts getting to some core national interests that the United States has.” Debates about Syria have played out across the Atlantic.

○ It was an idea that was quickly praised by top officials in Syria and some lawmakers in the United States.

○ This has greatly limited the influence the United States has with armed groups that are likely to control much of Syria if Mr. Assad is ousted.

○ It also needs Iran’s help on Syria.

Figure 6.3: Refinery’s phrase extraction feature. *Left:* User has selected some initial phrases related to the war in Syria from news articles. *Right:* Hitting the “similar” button brings up another set of phrases similar to the initial set.

algorithm processes raw text and does not make use of the topic model output. Next, to help the user efficiently select a subset of relevant sentences from their corpus, Refinery offers two exploration modes. The first, triggered by the button marked “Variety”, implements the KLSum algorithm for multi-document summarization [29]. This algorithm ranks candidate sentences higher if their addition to the current set would bring the unigram distribution closer to that of the corpus as a whole, intuitively preferring sentences that contain globally relevant information that does not yet appear in the set of selected sentences. The second exploration method, labeled “Similar”, simply ranks candidates by cosine similarity to the current summary’s unigram distribution. Each sentence in the summary is linked to and highlighted in its original source document, facilitating the creation of a comprehensive set of notes with fast access to their provenance.

6.3 Discussion and Related Work

Refinery provides a first step toward allowing non-technical professionals to explore large document collections with modern topic models. Several others groups have strived to simplify and democratize the use of topic models. Notably, [18] created a web-based navigator to help users understand the relationship between topics and documents, while [19] developed a visualization package for assessing the goodness of topics. Topicnets [27] focuses on a graph-based exploration and visualization of topics, while other packages such as Gephi [7] and Tethne [65] are often used to create similar visualizations of an already-learned set of topics. Most prior work has focused on the parametric latent Dirichlet allocation (LDA) topic model [14].

Refinery differs from these packages in the way it simplifies and exposes the entire process of analyzing text with topic models, and its use of scalable Bayesian nonparametric topic models. Refinery also supports phrase extraction, which allows for a more refined search across documents.

Building on the BNPy toolbox allows potential future extensions to more advanced models which cluster observations organized as a time-series or network. Ultimately, we plan to support a larger variety of potential document file types including structured word processor files, spreadsheets, and presentations.

Chapter 7

Conclusions & Future Directions

7.1 Summary of Contributions

Unstructured documents and networks represent a rich source of data that are ripe for analysis and this thesis contributes a series of novel Bayesian nonparametric models for their analysis. Another major part of this thesis represent contributions in dealing with the problem of scaling inference and we show that these new methods are useful for escaping local optima and model selection.

We contribute in Chapter 3 the Doubly Correlated Nonparametric Model (DCNM). This is the first model that captured both the effects of metadata, correlations, and was truly nonparametric. We show how metadata could be useful as a way to not only increase held out likelihood or AUC measures, but also as a qualitative tool to understand new documents or nodes. Correlations allow us to incorporate a much richer structure for our latent topics, allowing us to develop graphs that show how these topics might be related. We also show that by using an elliptical slice sampler, we can get better performances over the Metropolis-Hastings Sampler and can perform a Chibs Style estimation more accurately. Furthermore, we develop a retrospective MCMC approach that samples the topic space allowing for a truly nonparmetric approach. The only issue with the DCNM model is its scalability due to the use of MCMC for learning and inference. This challenge leads us to develop more scalable variational methods.

We then focus our attentions on building an HDP Relational model in Chapter 4 and shifting over to variational inference. This allows us to leverage techniques from the stochastic optimization literature to develop a mini-batch inference scheme with convergence guarantees. Our contribution is the first to develop a Bayesian nonparametric relational model along with a variational inference procedure. By using an assortative assumption for the block matrix, we develop an even more scalable approach. This is valid because the mixed-membership nature of the nodes makes the full stochastic block matrix redundant. To perform model comparisons, we built split-delete moves that help explore the model space allowing for us to more easily escape local optima. Future extensions should allow for some ways to increase the number of topics/communities as the split-delete moves are used to mostly decrease this space.

Sticking with variational inference for topic models, we tackle the problem of model selection in Bayesian nonparametric topic models by developing a scalable memoized variational inference approach in Chapter 5. The significant contribution here comes from extending the memoized variational inference for the DP mixture model to the HDP. This was not a trivial challenge as the admixture aspect of the topic space created additional difficulties when it comes to estimating the entropy terms within the lower bound. We develop moves that can merge and delete topics, but not necessarily grow, but we show that using a memoized variational inference approach significantly beats other nonparametric models that can modify the cardinality of their latent space.

Finally in Chapter 6 we build an open source web platform called Refinery to provide non-technical users the ability to simply drag and drop their data to perform topic modeling. The web app can be installed using two command lines and leverages the latest in BNPy to provide sophisticated topic modeling that has been abstracted away from the user.

7.2 Automating Inference in Probabilistic Models

A broad and important research direction lies in simplifying the process of deriving and implementing variational inference for new models. The standard process for this was often carefully choosing a set of variational distributions that were part of the exponential family, resulting in analytical tractable derivations. Most of the time developing a new model is spent on deriving these update equations, writing this in code, and ensuring both the derivation and its implementation is correct.

BNPy is a python based package that provides a powerful step in that direction. The goal of BNPy was to create a flexible framework for probabilistic clustering models that offer scalable inference along with the ability to do model selection during inference (i.e. learn the number of clusters). The package can perform both batch learning, stochastic variational inference, and memoized variational inference. Currently, it can support a wide variety of standard mixture models, admixture models, hidden markov models, and more as well as their nonparametric extensions 7.1. Other packages like STAN can also perform inference in the same class of models, but since their inferential machinery is based off of automated differentiation techniques, this results in the user having to specify a model that marginalizes out its discrete variables. Furthermore, performance in STAN for these kinds of discrete clustering models have been noticeably worse as seen in the DCNM results.

When the model contains relationships between variables that are non-conjugate, the simple updates afforded to us by the exponential families are no longer possible. Methods have been developed to deal with these non-conjugate relationships such as Taylor approximations [82], which replaces

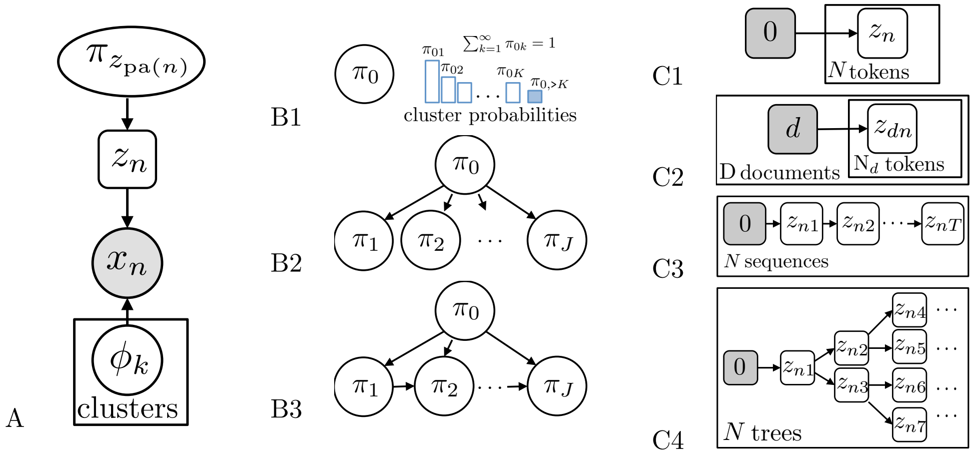
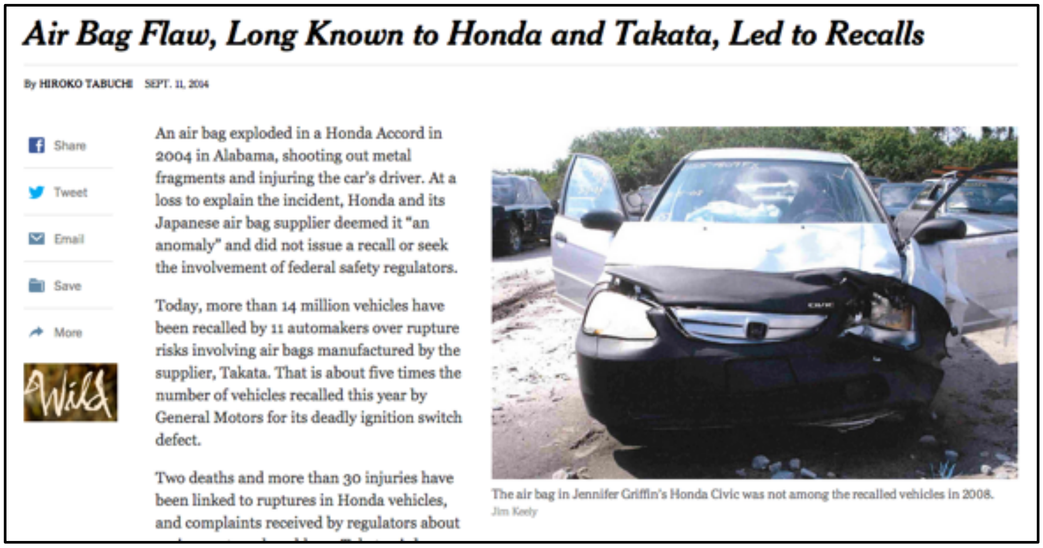


Figure 7.1: BNPy's compositional view of clustering models. *Col. A:* Generative model for one data token x_n. *Col. B:* Possible dependency graphs for cluster probability vectors π: DP (top), HDP (middle), and dependent DPs (bottom). *Col. C:* Possible graphs for cluster indicators z. The BNPy framework defines a single *allocation* model by combining fixed graph structures for π, z from columns B and C. Each model can be either parameteric or nonparametric, based on the prior distribution of the top-level π_0. The pair (B1,C1) yields mixture models [11], while (B2, C2) gives topic models [15, 77], and (B2, C3) gives hidden Markov models [8]. The pair (B2, C4) yields hidden Markov trees used for multi-scale image modeling [20, 45] and text parsing [22, 50]. B3 and C2 could yield a topic model where frequencies vary over time, as in [12]. This framework also extends to relational block models [3, 42], hierarchical or sticky sequential models [23, 31], and spatial models for image segmentation [74]. Figure and caption text taken from [37].

the variational distribution with a Taylor approximation for conjugacy or black box methods [66], which require only that the gradient of the variational distribution can be derived. Of the two, the latter is more general as it allows for both the optimization of continuous and discrete latent variables. Unfortunately, these kinds of approaches suffer from significant local optima issues, especially when the models contain several non-conjugate relationships. However, the benefits of this research direction seem very promising for the development of machine learning systems that are not only robust, but capable of being implemented by non-experts.

7.3 Investigative journalism and Machine Learning

The current thesis has focused on the development of scalable machine learning algorithms for unstructured relational and document datasets. An area where such algorithms can also be useful is in assisting journalists. At the New York Times, data plays a large role not only in providing journalists with additional insights that help shape their narrative, but also as a way to discover new leads. For example, given a large database of vehicle accident reports, how can we determine which cases are relevant to a recall issued by the NHTSA (National Highway Safety Traffic Authority).



Air Bag Flaw, Long Known to Honda and Takata, Led to Recalls

By HIROKO TABUCHI SEPT. 11, 2014

An air bag exploded in a Honda Accord in 2004 in Alabama, shooting out metal fragments and injuring the car's driver. At a loss to explain the incident, Honda and its Japanese air bag supplier deemed it "an anomaly" and did not issue a recall or seek the involvement of federal safety regulators.

Today, more than 14 million vehicles have been recalled by 11 automakers over rupture risks involving air bags manufactured by the supplier, Takata. That is about five times the number of vehicles recalled this year by General Motors for its deadly ignition switch defect.

Two deaths and more than 30 injuries have been linked to ruptures in Honda vehicles, and complaints received by regulators about

The air bag in Jennifer Griffin's Honda Civic was not among the recalled vehicles in 2008.
Jim Keely

This was a problem that one of our business reporters, Hiroko Tabuchi, was dealing with as she began to label which accident reports were linked to the faulty Takata airbags. Machine learning helped save her time and narrow her candidate list from thousands of potential cases to a ranked list that allowed her to focus on the most promising leads. The initial analysis took these reports and tokenized them in a bag-of-words fashion, similar to the preprocessing step used in topic modeling and then used these as features for a predictive model to predict which complaints were related to faulty airbags.

News organizations are also leveraging the Freedom of Information Act to obtain data that could be pertinent to their stories. However, these datasets are often so large that being able to manually inspect each document can be prohibitive. This is where topic models can come and help elucidate the overall structure of this corpus. A classic example is the case of the Hillary Clinton e-mails. With over 30 thousand emails, no reporter has the time and energy to parse through such a collection. Yet there are almost certainly insights in that data that would help reporters understand the way a powerful figure such as Hillary Clinton makes decisions when it comes to national security issues. Furthermore, e-mails represent a natural graph stucture if we consider the recipient and the sender as nodes and an edge corresponding to an e-mail exchange. Processing the data in this fashion, we can then apply our relational models to such a datset to discover communities where e-mails are frequently exchanged. This in turn could help shed light on the individuals who often work closely with one another as a proxy for the political structure within an organization.

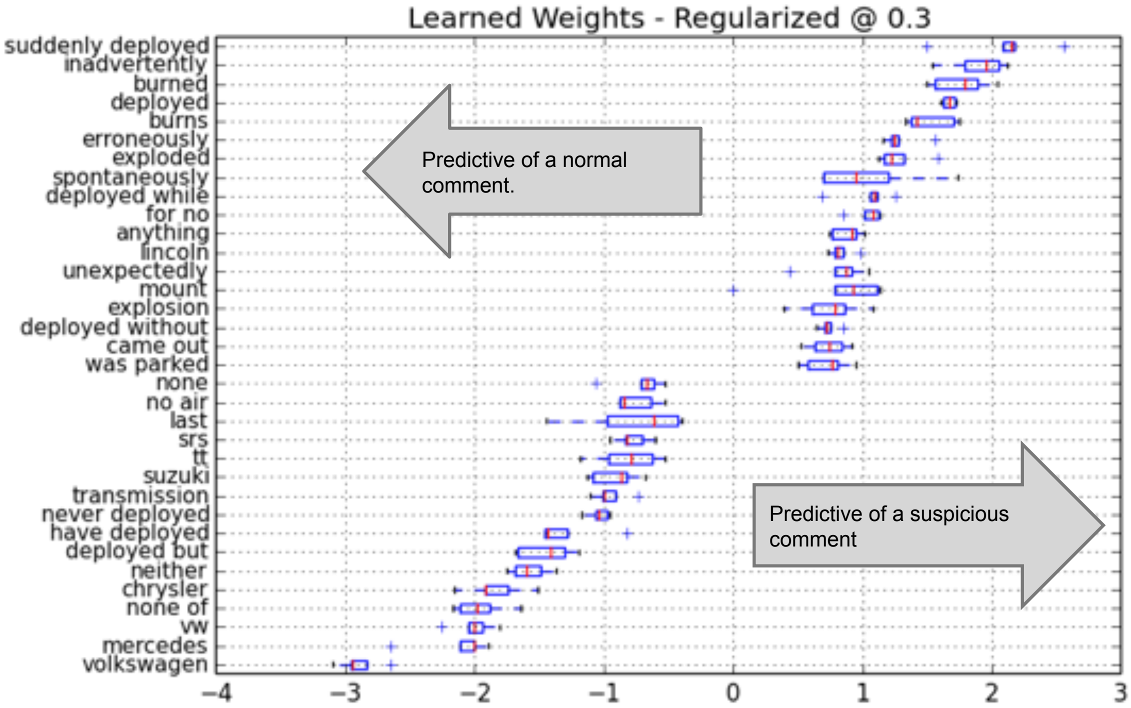


Figure 7.2: The figure above illustrates the terms that were predictive for complaints associated with airbag accidents. A logistic regression model was trained on this database of 33,204 comments where Hiroko Tabuchi hand labeled 2219 as being suspicious of comments associated with faulty airbags. Terms such as *suddenly deployed, inadvertently, burned* were highly predictive of airbag accidents. Counter-intuitively, by looking at the examples in which the algorithm got wrong, we were able to discover additional cases that Hiroko did not find herself, which ultimately led to 7 new cases that she discovered that were linked to faulty Takata airbags. One can imagine using more sophisticated topic models to extract features within this dataset that could have helped further improve the discovery of new cases.

Bibliography

[1] A. Agovic and A. Banerjee. Gaussian process topic models. In UAI, 2010. URL http://event.cwi.nl/uai2010/papers/UAI2010_0184.pdf.

[2] E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership stochastic blockmodels. JMLR, 9, 2008.

[3] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. 2009.

[4] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. Machine learning, 50(1-2):5–43, 2003.

[5] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. 33(5):898–916, 2011.

[6] S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. 2013.

[7] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. In International AAAI Conference on Weblogs and Social Media, 2009. URL http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.

[8] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. 2001.

[9] D. Blackwell and J. B. MacQueen. Ferguson distributions via pólya urn schemes. The annals of statistics, 1(2):353–355, 1973.

[10] D. M. Blei. Probabilistic topic models. Communications of the ACM, 55(4):77–84, 2012.

[11] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. 1(1):121–143, 2006.

[12] D. M. Blei and J. D. Lafferty. Dynamic topic models. 2006.

[13] D. M. Blei and J. D. Lafferty. A correlated topic model of science. AAS, 1(1):17–35, 2007.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. J. Mach. Learn. Res., 3:993–1022, March 2003. ISSN 1532-4435.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. 3:993–1022, 2003.

[16] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. JACM, 57(2):7, 2010.

[17] M. Bryant and E. B. Sudderth. Truly nonparametric online variational inference for hierarchical Dirichlet processes. 2012.

[18] A. J.-B. Chaney and D. M. Blei. Visualizing topic models. In ICWSM, 2012.

[19] J. Chuang, C. D. Manning, and J. Heer. Termite: Visualization techniques for assessing textual topic models. In Advanced Visual Interfaces, 2012. URL http://vis.stanford.edu/papers/termite.

[20] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. 46(4):886–902, 1998.

[21] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. An. Stat., 1(2):209–230, 1973.

[22] J. R. Finkel, T. Grenager, and C. D. Manning. The infinite tree. 2007.

[23] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. A sticky HDP-HMM with application to speaker diarization. 5(2A):1020–1056, 2011.

[24] D. Gillick. Sentence boundary detection and the problem with the U.S. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pages 241–244. Association for Computational Linguistics, 2009.

[25] D. Gillick. splitta: statistical sentence boundary detection, 2010. URL https://code.google.com/p/splitta/.

[26] P. Gopalan, D. M. Mimno, S. Gerrish, M. J. Freedman, and D. M. Blei. Scalable inference of overlapping communities. In NIPS, pages 2258–2266, 2012.

[27] B. Gretarsson, J. O'donovan, S. Bostandjiev, T. Höllerer, A. Asuncion, D. Newman, and P. Smyth. Topicnets: Visual analysis of large text corpora with topic modeling. ACM Transactions on Intelligent Systems and Technology (TIST), 3(2):23, 2012.

[28] T. L. Griffiths and M. Steyvers. Finding scientific topics. 2004.

[29] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 362–370. Association for Computational Linguistics, 2009.

[30] M. Hashimoto. Vagrant: Up and Running. O'Reilly Media, Inc., 2013. URL http://www.virtualbox.org.

[31] K. A. Heller, Y. W. Teh, and D. Görür. Infinite hierarchical hidden Markov models. 2009.

[32] Q. Ho, L. Song, and E. Xing. Evolving cluster mixed-membership blockmodel for time-varying networks. In AISTATS, 2011.

[33] M. Hoffman, D. Blei, C. Wang, and J. Paisley. Stochastic variational inference. 14(1), 2013.

[34] M. D. Hoffman and A. Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. Journal of Machine Learning Research, 15:1593–1623, 2014. URL http://jmlr.org/papers/v15/hoffman14a.html.

[35] M. C. Hughes. BNPy: Bayesian nonparametric Python toolbox, 2015. URL https://bitbucket.org/michaelchughes/bnpy/.

[36] M. C. Hughes and E. B. Sudderth. Memoized online variational inference for Dirichlet process mixture models. 2013.

[37] M. C. Hughes and E. B. Sudderth. bnpy: Reliable and scalable variational inference for bayesian nonparametric models. In NIPS - Probabilistic Programming Workshop, 2014.

[38] M. C. Hughes and E. B. Sudderth. Memoized online variational inference for dirichlet process mixture models. In NIPS, pages 422–430, 2014.

[39] M. C. Hughes, D. Kim, and E. B. Sudderth. Reliable and scalable variational inference for the hierarchical dirichlet process. In AISTATS, 2015.

[40] M. I. Jordan. Graphical models. 19(1):140–155, 2004.

[41] C. Kemp, J. Tenenbaum, T. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In AAAI, 2006.

[42] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. 2006.

[43] D. Kim and E. Sudderth. The doubly correlated nonparametric topic model. In NIPS, 2011.

[44] D. Kim, M. C. Hughes, and E. B. Sudderth. The nonparametric metadata dependent relational model. 2012.

[45] J. J. Kivinen, E. B. Sudderth, and M. I. Jordan. Learning multiscale representations of natural scenes using Dirichlet processes. 2007.

[46] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Phys. Rev. E, 80(1):016118, July 2009. doi: 10.1103/PhysRevE.80.016118. URL http://pre.aps.org/abstract/PRE/v80/i1/e016118.

[47] S. L. Lauritzen. Graphical models. Clarendon Press, 1996.

[48] E. Lazega. The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership. Oxford University Press, 2006.

[49] W. Li, D. Blei, and A. McCallum. Nonparametric Bayes Pachinko allocation. In UAI, 2008.

[50] P. Liang, S. Petrov, M. I. Jordan, and D. Klein. The infinite PCFG using hierarchical Dirichlet processes. 2007.

[51] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The next frontier for innovation, competition, and productivity, May 2011. URL http://www.mckinsey.com/mgi/publications/big_data/pdfs/MGI_big_data_full_report.pdf.

[52] D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In UAI, 2008. URL http://www.cs.umass.edu/~mimno/papers/dmr-uai.pdf.

[53] D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In UAI 24, pages 411–418, 2008.

[54] K. Mouritsen, R. Poulin, J. McLaughlin, and D. Thieltges. Food web including metazoan parasites for an intertidal ecosystem in new zealand. Ecology, 92(10):2006, 2011.

[55] I. Murray, R. Adams, and D. MacKay. Elliptical slice sampling. In AISTATS, 2010.

[56] R. M. Neal. Probabilistic inference using markov chain monte carlo methods. 1993.

[57] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Learning in graphical models, pages 355–368. Springer, 1998.

[58] T. Nepusz, A. Petróczi, L. Négyessy, and F. Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. Phys Rev E Stat Nonlin Soft Matter Phys, 77(1 Pt 2):016107, 2008.

[59] M. E. Newman. Modularity and community structure in networks. Proceedings of the National Academy of Sciences, 103(23):8577–8582, 2006.

[60] V. Oracle. Virtualbox. User Manual–2013, 2013. URL http://www.vagrantup.com.

[61] P. Orbanz and Y. W. Teh. Bayesian nonparametric models. In Encyclopedia of Machine Learning, pages 81–89. Springer, 2010.

[62] J. Paisley, C. Wang, and D. Blei. The discrete infinite logistic normal distribution for mixed-membership modeling. In AISTATS, 2011.

[63] J. Paisley, C. Wang, and D. Blei. The discrete infinite logistic normal distribution for mixed-membership modeling. 2011.

[64] O. Papaspiliopoulos and G. Roberts. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. Biometrika, 4, 2008.

[65] E. Peirson, A. Baker, and R. Subramanian. Tethne: Bibliographic network analysis in python, 2015. URL https://github.com/diging/tethne.

[66] S. G. Ranganath, Rajesh and D. M. Blei. Black box variational inference. In AISTATS, 2014.

[67] L. Ren, L. Du, L. Carin, and D. Dunson. Logistic stick-breaking process. JMLR, 12, 2011.

[68] H. Robbins and S. Monro. A stochastic approximation method. The Annals of Mathematical Statistics, 22(3):400–407, 1951.

[69] A. Rodriguez and D. B. Dunson. Nonparametric bayesian models through probit stick-breaking processes. J. Bayesian Analysis, 2011.

[70] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences, 105(4):1118–1123, 2008.

[71] J. Sethuraman. A constructive definition of Dirichlet priors. Stat. Sin., 4:639–650, 1994.

[72] J. Sethuraman. A constructive definition of Dirichlet priors. 4:639–650, 1994.

[73] E. Sudderth. Graphical models for visual object recognition and tracking. PhD thesis, Massachusetts Institute of Technology, 2006.

[74] E. B. Sudderth and M. I. Jordan. Shared segmentation of natural scenes using dependent pitman-yor processes. 2009.

[75] Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. In AIStats 10, 2005.

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

[77] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. 101(476): 1566–1581, 2006.

[78] Y. W. Teh, K. Kurihara, and M. Welling. Collapsed variational inference for HDP. 2008.

[79] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning, 1(1-2):1–305, 2008.

[80] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning, 1(1-2):1–305, 2008.

[81] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In ICML, 2009.

[82] C. Wang and D. M. Blei. Variational inference in nonconjugate models. arXiv preprint arXiv:1209.4360, 2012.

[83] C. Wang, J. Paisley, and D. Blei. Online variational inference for the hierarchical Dirichlet process. 2011.

[84] Y. Wang and G. Wong. Stochastic blockmodels for directed graphs. JASA, 82(397):8–19, 1987.

[85] D. Zoran and Y. Weiss. Natural images, Gaussian mixtures and dead leaves. 2012.