# Hierarchical Topic Models

**Andrew Leverentz**
aleveren@eng.ucsd.edu

## Abstract

In this paper, we survey techniques for automatically inferring hierarchical structures from unlabeled collections of text documents. We begin with an overview of probabilistic topic modeling, as implemented in models such as Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA). After examining some limitations of these models, we discuss a line of research which has extended the LDA model and has attempted to address some of its weaknesses. This approach treats topics as belonging to a potentially infinite tree; in this context, Bayesian non-parametric techniques based on Dirichlet processes are used to automatically select the structure of the tree. We compare and contrast two topic models that have been used for this approach—the Nested Chinese Restaurant Process (NCRP) and the Nested Hierarchical Dirichlet Process (NHDP)—and we discuss several inference algorithms that have been published in the literature.

## 1 Introduction

In recent decades, large-scale digital archives and information-sharing networks such as the World Wide Web have led to the creation of vast collections of electronic textual data. As these collections grow, the problem of efficiently discovering relationships between documents, or between user queries and documents, has become more prominent. The field of *topic modeling* has produced several automated approaches to solving this problem.

One underlying theme of topic modeling is the notion that words with similar meanings tend to co-occur with similar sets of words. However, due to ambiguities in natural language and variations between different authors, these co-occurrence patterns are not rigid and deterministic. To address this, many techniques within the field of topic modeling use a probabilistic framework. In this probabilistic framework, topics are represented as discrete probability distributions over vocabulary words, and documents may contain mixtures of multiple topics.

In subsequent sections, we briefly review the origins of the probabilistic framework for topic modeling, with particular focus on Latent Dirichlet Allocation (LDA). Then, we trace the development of a line of research which extends the LDA model and is capable of producing not just a flat collection of topics but a nested hierarchy of topics. We conclude with a discussion of potential avenues for future research.

## 2 Motivation

Two of the main applications of topic models are (1) discovering relations between documents and (2) indexing large collections of documents.

**Discovering relations between documents:** We often want to determine which sets of documents within a corpus share the same topic or subject matter. One approach is to formulate this as a clustering task with soft cluster assignments—also known as a *mixed membership model*—where

UC San Diego, Department of Computer Science and Engineering
Research Examination, Fall Quarter 2017

each document can be assigned to a mixture of multiple topics or clusters. For example, a news article about a national politician's economic policies might be represented as a mixture between topics corresponding to both *politics* and *economics*. Thus, if we are looking for related documents, we could return the set of documents which also contain some significant proportion of either *politics* or *economics*.

**Indexing large collections of documents:** In the context of information retrieval, the task is to take a user query (expressed perhaps as a collection of keywords or as a natural-language sentence) and efficiently find a set of documents which are deemed most relevant to that query.

Both of these tasks are complicated by ambiguities in natural language. As noted in Deerwester et al [6], *synonymy* (in which multiple words have nearly identical meanings) and *polysemy* (in which a single word may have multiple meanings, depending on the context) make it difficult to draw conclusions about relationships between documents based on the specific set of words that appear in each document. For example, two closely related documents might express nearly the same concept using completely different words, whereas two unrelated documents might coincidentally both use a particular word in different senses. Because of this, it is necessary to move beyond superficial representations of documents (such as strings of characters or tokens, or histograms of vocabulary words) and instead represent documents using some notion of *latent semantics*, or underlying meaning.

Topic models accomplish this by treating topics as probability distributions over the vocabulary and allowing documents to draw from mixtures of multiple topics. For example, a topic relating to sports would assign relatively high probability to words such as "player," "team," and "game." Similarly, a topic about medicine would assign relatively high probability to "doctor," "illness," and "pharmacy." Then, a document about a basketball player recovering from an injury might consist of $70\%$ *sports* and $30\%$ *medicine*, whereas a document about physical therapy for athletes might consist of $70\%$ *medicine* and $30\%$ *sports*.

Hierarchical topic models extend this idea by acknowledging that whether or not two documents share the same topic may depend on the level of abstraction that the user is interested in. By explicitly treating the set of available topics as a tree, hierarchical topic models can therefore represent documents as mixtures of nested topics at varying levels of abstraction. For example, in some contexts, we might consider articles about the San Diego Padres and articles about the Chicago Bulls to belong to the same topic (namely, *sports*), whereas in other contexts, they belong to distinct topics (*baseball* versus *basketball*). Thus, *baseball* and *basketball* could be considered subtopics under the broader topic of *sports*. For indexing tasks, hierarchical topic models allow the possibility of expanding or narrowing the set of relevant documents, depending on the level of abstraction desired by the user. For example, when searching for documents related to the keywords "Alan Turing," the default system behavior might be to display documents related to *computer science*, with the option of narrowing matches to a more specific subtopic (e.g., *British computer scientists*) or expanding matches to a more general supertopic (e.g., *science and engineering*).

## 3 Probabilistic Topic Models

Before we discuss models which can generate hierarchies of topics, we will first discuss two probabilistic models which generate "flat" (that is, non-nested) collections of topics. These models are Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA).

### 3.1 A Non-Probabilistic Precursor: Latent Semantic Analysis

The first of these, PLSA, was inspired by an earlier, non-probabilistic model known as Latent Semantic Analysis (LSA), which was introduced in [6]. In LSA, a corpus of documents is summarized as a matrix; each column of this matrix represents a document, each row represents a term in the vocabulary, and each entry specifies the number of times a given term appears in a given document. In mathematical notation, we have a matrix $M$ such that the entry $M_{i,j}$ denotes the number of times that the $i^{\text{th}}$ vocabulary word appears in the $j^{\text{th}}$ document. If the corpus contains a sufficiently diverse set of documents, then most documents will only contain a small subset of the full vocabulary, and the resulting matrix will be sparse. After constructing this matrix, we may optionally transform it according to per-term and per-document statistics. That is, if $M$ is an $N \times D$ matrix, we compute a new $N \times D$ matrix $X$ whose entries are given by

$$X_{i,j} = f(M_{i,j}, M_{i,1:D}, M_{1:N,j}). \tag{1}$$

Several variants of LSA exist which use different transformations in this step, as described in [25], although the original paper [6] uses an untransformed matrix, with $X = M$. Then, the singular value decomposition (SVD) of the matrix $X$ is computed. This represents the matrix as a product of three matrices:

$$X = U \Sigma V^\top \tag{2}$$

Here, $U$ and $V$ are matrices of orthogonal column vectors. In $U$, each row corresponds to a term in the vocabulary, and in the transposed matrix $V^\top$, each column corresponds to a document. Moreover, $\Sigma$ is a diagonal matrix containing scaling factors known as *singular values*. If we compute the SVD using a low-rank approximation, this corresponds to truncating the number of columns in $U$ and the number of rows in $V^\top$. At the same time, we also discard all but the largest singular values in $\Sigma$. The truncated rows from $U$ (corresponding to terms) and truncated columns from $V^\top$ (corresponding to documents) are known as *latent semantic vectors*. We can view these vectors as concise representations of terms and documents in a low-dimensional *latent semantic space*. Using these representations, we can measure the similarity between pairs of documents by computing a normalized dot product (also known as *cosine similarity*) of the corresponding latent semantic vectors.

However, one major limitation of this model is that the latent semantic vectors corresponding to terms and documents do not have an intuitive interpretation. Consequently, there are relatively few theoretical principles to guide the design of algorithms for (non-probabilistic) LSA. For example, selecting the transformation $f$ which maps $M$ to $X$ must typically be done by trying many variations and then determining which one happens to yield the most reasonable-looking results in practice. We will see that subsequent models have addressed this limitation by using a probabilistic framework.

Lastly, we note that this model, as with many other topic models, uses the *bag-of-words* simplification, in which the specific sequence of words in a document is ignored, and all that matters is the frequency of terms within each document. Topic models which eliminate the bag-of-words simplification are beyond the scope of this survey, but they constitute an active area of research.

### 3.2 Probabilistic Latent Semantic Analysis

In abstract terms, Probabilistic Latent Sematic Analysis (PLSA) takes a conceptual approach that is similar to LSA, but in this case the latent semantic representations of topics associated with terms and documents are combined using a probabilistic model, rather than simply a linear algebraic one. PLSA, introduced in Hofmann [15], uses an *aspect model* to represent documents as probabilistic mixtures of topics. The following discussion uses a reformulation, outlined in Barber [2], which is equivalent to the original PLSA model.

In this model, the observed dataset is a collection of $D$ documents, and $N_d$ represents the length of document $d$, measured in words. We assume that these documents share a fixed vocabulary $\mathcal{V}$, which is a set of words. Furthermore, we assume that there are a total of $K$ available topics, each of which is represented as a distribution over the words in the vocabulary. Mathematically, we represent each topic $k \in \{1, \dots, K\}$ as a vector $\theta_k$ of length $|\mathcal{V}|$ whose components sum to 1. Each document $d$ is associated with a mixture of topics $\phi_d$, which we represent as a vector of length $K$ whose components sum to 1, and which we view as a probability distribution over topics. As with LSA, we will use the bag-of-words assumption to ignore the order in which words appear.

Returning to a previously mentioned example, suppose topic 1 is *sports* and topic 2 is *medicine*. Then, for a document about sports medicine, with more focus on *sports* than on *medicine*, we might have $\phi_d = ({}^7\!/\!{}_{10}, {}^3\!/\!{}_{10}, 0, \dots, 0)$. If the first two vocabulary words in $\mathcal{V}$ are $v_1 =$ "player" and $v_2 =$ "doctor," then the topic vectors $\theta_1$ (for *sports*) and $\theta_2$ (for *medicine*) ought to satisfy

$$\theta_{1,1} > \theta_{1,2}, \qquad \text{meaning} \qquad \theta_{\textit{sports},\text{“player”}} > \theta_{\textit{sports},\text{“doctor”}}, \text{ and} \tag{3}$$
$$\theta_{2,1} < \theta_{2,2}, \qquad \text{meaning} \qquad \theta_{\textit{medicine},\text{“player”}} < \theta_{\textit{medicine},\text{“doctor”}}. \tag{4}$$

Once we have defined topic vectors $\theta_k$ for $k \in \{1, \dots, K\}$ and per-document topic mixtures $\phi_d$ for $d \in \{1, \dots, D\}$, we assume that documents are generated according to the following probabilistic procedure:

- For each document $d \in \{1, \ldots, D\}$:
  - For each word-slot $n \in \{1, \ldots, N_d\}$:
    - Select a topic $z_{d,n} \in \{1, \ldots, K\}$ according to the current document's topic mixture $\phi_d$.
    - Select a word $t_{d,n} \in \{1, \ldots, |\mathcal{V}|\}$ from the vocabulary according to the topic vector indexed by $z_{d,n}$.

This model can be summarized by the following probability distributions:

$$z_{d,n} \sim \text{Categorical}(\phi_d) \qquad \text{for the } n^{\text{th}} \text{ word in document } d \tag{5}$$

$$t_{d,n} \sim \text{Categorical}(\theta_{z_{d,n}}) \qquad \text{for the } n^{\text{th}} \text{ word in document } d \tag{6}$$

Figure 1 illustrates this model using plate-diagram notation. In this notation, unshaded circles represent latent (i.e., unobserved) random variables, and shaded circles represent observed random variables. Uncircled variables denote hyperparameters, which are variables with no associated probability distribution. Edges represent conditional dependencies between variables, which correspond to the probability distributions listed above. Finally, the rectangular plates denote repetitions of the nodes and edges that they contain.

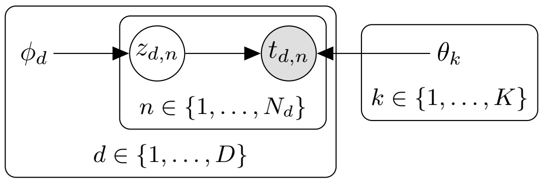

Figure 1: Plate diagram for the PLSA model.

Thus, the joint probability of the PLSA model is

$$p(\vec{t}, \vec{z}) = \prod_{d=1}^{D} \prod_{n=1}^{N_d} p(t_{d,n} \mid z_{d,n})\, p(z_{d,n}) \tag{7}$$

$$= \prod_{d=1}^{D} \prod_{n=1}^{N_d} \theta_{z_{d,n}, t_{d,n}}\, \phi_{d, z_{d,n}} \tag{8}$$

Within this framework, it becomes possible to infer the latent topic mixture of each document using a maximum-likelihood optimization algorithm. It is typically infeasible to directly optimize the likelihood of the data, but another option is the expectation–maximization (EM) algorithm, which was introduced in Dempster et al [7] and summarized concisely in Barber [2]. The EM algorithm generally consists of alternating between estimating distributions over the latent topic assignments $z_{d,n}$ (known as the "E step") and maximizing a quantity related to the likelihood of the data (known as the "M step"). These steps are repeated until the estimates appear to have converged.

### 3.3 Latent Dirichlet Allocation

Since the PLSA model does not specify probability distributions for the unobserved parameters $\phi_d$ and $\theta_k$, Bayesian inference methods are not applicable. It is also prone to overfitting, meaning that its results can be overly sensitive to the randomness that is inherent in generating or sampling the training data. To address these problems, Latent Dirichlet Allocation (LDA) extends PLSA by using the Dirichlet distribution as a Bayesian prior on the space of discrete probability distributions. A simpler version of this model was introduced in Blei et al [5], although the term "LDA" now typically refers to the variant proposed by Griffiths and Steyvers [12].

Recall that an $m$-dimensional Dirichlet distribution is a continuous distribution over the set of length-$m$ vectors whose components are non-negative and sum to one. In other words, it is a meta-distribution over discrete distributions over a set of $m$ points. The Dirichlet distribution is parameterized by a length-$m$ vector of positive real numbers, $\vec{\alpha} = (\alpha_1, \ldots, \alpha_m)$, and its density over the $m$-dimensional simplex is proportional to

$$\text{Dirichlet}(\vec{x} \mid \vec{\alpha}) \propto \prod_{i=1}^{m} x_i^{(\alpha_i - 1)} \tag{9}$$

The Dirichlet distribution is a higher-dimensional generalization of the beta distribution, whose density is proportional to

$$\text{Beta}(x \mid \alpha_1, \alpha_0) \propto x^{(\alpha_1 - 1)}(1-x)^{(\alpha_0 - 1)} \tag{10}$$

The generative model used in LDA is given by:

$$\theta_k \sim \text{Dirichlet}(\alpha) \qquad \text{for each topic } k \tag{11}$$
$$\phi_d \sim \text{Dirichlet}(\beta) \qquad \text{for each document } d \tag{12}$$
$$z_{d,n} \sim \text{Categorical}(\phi_d) \qquad \text{for the } n^{\text{th}} \text{ word in document } d \tag{13}$$
$$t_{d,n} \sim \text{Categorical}(\theta_{z_{d,n}}) \qquad \text{for the } n^{\text{th}} \text{ word in document } d \tag{14}$$

In other words, the per-topic probability distributions ($\theta_k$) over the vocabulary are i.i.d. draws from a Dirichlet distribution. Then, each document is associated with a random draw $\phi_d$ from another Dirichlet distribution; this represents the proportion of topics present in document $d$. Within each document, for each position $n \in \{1, \ldots, N_d\}$, we select a topic indicator $z_{d,n}$ according to the probabilities in the vector $\phi_d$, and we select a vocabulary word $t_{d,n} \in \mathcal{V}$ according to the probabilities in the vector $\theta_{z_{d,n}}$. The plate diagram for this model is shown in Figure 2.

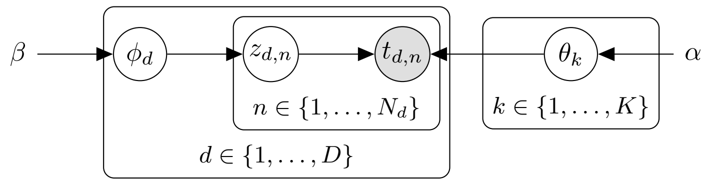

Figure 2: Plate diagram for the LDA model

The LDA model has two main advantages over the PLSA. First, by placing a prior on the topic mixtures, it specifies how new documents can be generated. Hence, it is possible to apply Bayesian techniques to go beyond a maximum-likelihood point estimate; instead, we can estimate the entire posterior distribution of the parameters $\theta$ and $\phi$ given the observed data. (We will discuss Bayesian inference algorithms further in the next section.) Second, the Dirichlet priors effectively act as regularizers, which allow us to quantitatively control the bias-variance tradeoff of the model. In particular, a stronger (more peaked) prior reduces the flexibility of the model, which in turns reduces overfitting at the cost of producing a more biased model.

Several important limitations of the LDA model are:

1. Unlike PLSA, the LDA model is not amenable to using the EM algorithm, since the EM update rules for LDA involve sums with prohibitively many terms. Instead, we must use alternative inference algorithms, which we will explore in the next section.

2. The number of topics must be specified in advance. Setting this parameter using standard model-selection techniques such as $k$-fold cross-validation can be time consuming. We will see that this can be addressed via Bayesian non-parametric techniques, in which one or more latent variables are drawn from an infinite-dimensional space.

3. A "flat" collection of topics has no built-in notion of levels of abstraction, and so the topics discovered by LDA may be difficult to interpret. This is because such models may mix abstract and concrete terms within a single topic. Nested topics allow us to represent containment relationships between relatively abstract topics, such as "sports in general" or "basketball" and relatively concrete topics, such as "Michael Jordan" or "the 1996 Chicago Bulls." The intention is to create more easily interpretable topics, each of which can be associated with a sharply distinguished level of abstraction.

4. A related issue involves extremely common words, such as “the,” “or,” and “this.” These words can lead to noisy results because they are roughly equally likely to appear within any topic. Consequently, they are often explicitly omitted during pre-processing of the dataset using a manually-curated list of so-called *stopwords*. In the context of nested topic hierarchies, we will see that this issue can be handled automatically, since nodes at the top of the hierarchy will naturally collect common stopwords; then, there is no need to explicitly filter out stopwords during pre-processing.
5. The bag-of-words approach fails to account for the ordering of words, even though this may significantly impact the meaning of a document; for example, in the bag-of-words approach, “man bites dog” is considered exactly equivalent to “dog bites man.” Solutions to this limitation are outside the scope of this report, although as mentioned above this is an active area of research.

## 4 Bayesian Inference Algorithms for Latent-Variable Models

Before we generalize the LDA model to accomodate hierarchies of topics, we will first discuss several inference algorithms which are relevant to both the LDA model and to hierarchical topic models.

In Bayesian inference for latent-variable models (such as LDA), our goal is generally to compute a posterior distribution over the latent variables, conditioned on the observed data and any fixed model parameters. That is, we wish to compute

$$\text{Posterior} = p(\text{latent variables} \mid \text{data}, \text{fixed parameters}). \tag{15}$$

This, in turn, allows us to compute relevant statistics on the posterior distribution, such as the expected value and variance of the latent variables.

A probabilistic model typically defines both the *likelihood* of the data, given by

$$\text{Likelihood} = p(\text{data} \mid \text{latent variables}, \text{fixed parameters}), \tag{16}$$

and the *prior distribution* of the latent variables, given by

$$\text{Prior} = p(\text{latent variables} \mid \text{fixed parameters}). \tag{17}$$

Therefore, in abstract terms, computing the posterior distribution is a matter of applying Bayes’ rule:

$$\begin{aligned}
&\text{Posterior} && (18)\\
&= p(\text{latent variables} \mid \text{data}, \text{fixed parameters}) && (19)\\
&= \frac{p(\text{data} \mid \text{latent variables}, \text{fixed parameters}) \times p(\text{latent variables} \mid \text{fixed parameters})}{p(\text{data} \mid \text{fixed parameters})} && (20)\\
&= \frac{\text{Likelihood} \times \text{Prior}}{p(\text{data} \mid \text{fixed parameters})} && (21)
\end{aligned}$$

Since the denominator in equation (21) does not depend on the value of the latent variables, we can think of it as merely a normalization constant, and we can compute it by marginalizing the numerator over all combinations of values for the latent variables. However, this marginalization is often difficult to compute—typically because it involves either an intractable integral or a sum with an unmanageable number of terms. Because of this, we must use approximate inference algorithms to estimate the posterior distribution.

The main Bayesian inference algorithms that have been proposed for the LDA model include Gibbs sampling and variational inference. We will see that similar algorithms can be applied to hierarchical topic models, although our choice of algorithms is constrained somewhat by the increased complexity of such models.

### 4.1 Gibbs Sampling

Gibbs sampling belongs to a particular class of stochastic sampling techniques known as Monte Carlo Markov Chain (MCMC) algorithms. It is a “Monte Carlo” technique because it involves estimating a quantity by repeatedly drawing samples from a probability distribution; furthermore, it is a “Markov

Chain" technique because it involves a sequence of stochastic updates to some state vector in which the transition probabilities at each step depend only upon the previous state.

To implement Gibbs sampling, it is necessary to compute the conditional probability distribution of each latent variable given all other latent variables, plus the full set of observed variables. Let $\vec{x}$ denote the observed data, let $\vec{z} = (z_1, \ldots, z_m)$ denote the latent variables in the model, and let $\vec{z}_{-k}$ denote the vector of all latent variables *except* $z_k$. Then, we must be able to compute the following quantity for each $k \in \{1, \ldots, m\}$:

$$p(z_k \mid \vec{z}_{-k}, \vec{x}) \tag{22}$$

These quantities are sometimes referred to as the *complete conditionals* of the model.

Gibbs sampling maintains a state vector in which each component corresponds to a latent variable. We initialize the state randomly and repeatedly update it according to stochastic rules. A single Markov-chain update involves iterating over the latent variables (optionally, this can be done in a random order). For each latent variable, we temporarily discard its current value, compute the conditional probability distribution of the current latent variable as given in equation (22), and sample an updated value from that conditional distribution. Then, there are theoretical results which guarantee that the long-term equilibrium distribution of the state vector will correspond to the true posterior (for example, see [19]). After reaching equilibrium, we can estimate posterior-distribution statistics by collecting a sample of state vectors, either by running multiple independent Markov chains in parallel, or by taking a sequence of regularly-spaced samples from a single Markov chain. Together, these sampled state vectors will constitute an empirical approximation of the posterior distribution. This procedure is summarized in Algorithm 1.

**Algorithm 1** Gibbs Sampling

- **Input:**
  - $\vec{x}$: an observed dataset
  - $C \geq 1$: number of independent Markov chains to simulate
  - $S \geq 1$: number of samples to save per Markov chain
  - $B \geq 1$: number of steps to discard before saving any samples ("burn-in period")
  - $L \geq 1$: number of steps to simulate between saved samples ("lag")
- **Output:**
  - $\hat{Z}$: a sequence of $C \times S$ samples which collectively approximate the posterior $p(\vec{z} \mid \vec{x})$
- **Procedure:**

```
Initialize $\hat{Z}$ as an empty list
for $i \in \{1, \ldots, C\}$ do
    Initialize $\vec{z}$ to a random vector compatible with the domains of the latent variables
    for $j \in \{1, \ldots, B + LS\}$ do
        Compute the conditional distribution $p(z_k \mid \vec{z}_{-k}, \vec{x})$
        Update $z_k$ by sampling from this conditional distribution
        if $j \geq B$ and $(j - B) \bmod L = 0$ then
            Append $\vec{z}$ to $\hat{Z}$
        end if
    end for
end for
return $\hat{Z}$
```

A closely related variant known as *collapsed Gibbs sampling* integrates out some of the latent variables, leaving us with a modified form of the complete conditional that depends only on the observed data and a subset of the latent variables. Then, we use Gibbs sampling on these remaining latent variables. The result will be an estimate of the posterior distribution over the remaining latent variables, conditioned on the observed data.

The main disadvantage of Gibbs sampling is that it can be difficult to determine when the Markov chain has reached equilibrium. Furthermore, compared to stochastic variational inference, which we discuss below, Gibbs sampling can require significantly more iterations through the full dataset.

Its main advantage is that, given enough time, the equilibrium distribution of the Markov chain will exactly match the posterior distribution.

### 4.2 Coordinate-Ascent Variational Inference

As outlined in [4], variational inference is an optimization technique that can be used to find an approximation to the posterior distribution. This technique involves defining a restricted set of functions and searching for the member of that set which most closely approximates the true posterior distribution. We define a set $Q$ of approximating distributions to be the set of all functions which can be written in the following form:

$$q(\vec{z}) = \prod_{k=1}^{m} q_k(z_k) \tag{23}$$

where $\vec{z} = (z_1, \ldots, z_m)$ represents the full set of latent variables in the model, and where each function $q_k$ is a member of a parameterized family of univariate distributions. If we let $\nu_k$ represent the parameters associated with the $k^{\text{th}}$ latent variable, then the function $q$ is fully specified by selecting a value for $(\nu_1, \ldots, \nu_m)$. By assuming that the posterior can be approximated as a product of univariate factors, we use what is known as the "mean-field approximation."

For probabilistic inference, we typically use *reversed Kullback-Leibler divergence* to measure the distance between a candidate distribution and the true posterior. In mathematical notation, if $p(\cdot \mid x)$ represents the true posterior distribution, we wish to find $q$ such that

$$\mathcal{L} = \mathrm{KL}(q \parallel p(\cdot \mid x)) \tag{24}$$

is minimized. By expanding the definition of $\mathcal{L}$, we can write

$$\mathcal{L} = \log p(x) - \mathrm{ELBO}, \tag{25}$$

where ELBO (short for "evidence lower bound") is defined as

$$\mathrm{ELBO} = E_q[\log p(\vec{z}, \vec{x})] - E_q[\log q(\vec{z})] \tag{26}$$

Then, since $\log p(x)$ does not depend on $q$, minimizing $\mathcal{L}$ is equivalent to maximizing ELBO. This fact will be useful later when we wish to monitor the convergence of our algorithm, because ELBO is signficantly easier to compute than the actual objective function $\mathcal{L}$.

To aid in optimizing the objective function, the model is typically constructed such that its conditional distributions are members of the *exponential family*, and conjugate priors are used wherever possible. By design, this is true of the LDA model and the hierarchical topic models discussed below. The exponential family [23] is a collection of parameterized sets of probability distributions which can be written in the following form, where $x$ represents the domain variable, and $\theta$ represents the distribution's parameters:

$$p(x \mid \theta) = h(x) \exp\left(\eta(\theta) \cdot T(x) - A(\theta)\right). \tag{27}$$

Here, $\eta(\theta)$ is a vector which contains the *natural parameters* of the model, and $T(x)$ is a vector containing the *sufficient statistics* of the model. To write a distribution in exponential-family form, we can attempt to compute the logarithm of the probability density; then, we try to derive the functional form of $\eta$ and $T$ by examining any terms which depend on both $x$ and $\theta$. Any terms which only depend on $x$ or only on $\theta$ can be folded into the functions $h(x)$ and $A(\theta)$. For example, the categorical distribution can be expressed as follows:

$$\begin{aligned} \text{Categorical}(x \mid \theta) &= \prod_{k=1}^{K} \theta_k^{\mathbb{1}[x=k]} && (28) \\ &= \exp\left( \sum_{k=1}^{K} \mathbb{1}[x=k] \log \theta_k \right). && (29) \end{aligned}$$

Thus, categorical distributions belong to the exponential family, with

$$\begin{aligned} \eta(\theta) &= (\log \theta_k)_{k \in 1:K} && \text{("natural parameters")}, && (30) \\ T(x) &= (\mathbb{1}[x=k])_{k \in 1:K} && \text{("sufficient statistics")}, && (31) \\ h(x) &= 1, && && (32) \\ A(\theta) &= 0. && && (33) \end{aligned}$$

At this point, it is also worth noting that we will also make use of the *inverse natural parameter transform*, which is the inverse of $\eta(\theta)$. For example, with the categorical distribution we can extract the discrete probabilities $\theta$ from $\eta$ by applying the softmax function:

$$\theta_k = \frac{\exp(\eta_k)}{\sum_{\ell=1}^{K} \eta_\ell}. \tag{34}$$

For variational inference, in addition to selecting an exponential-family model, the variational distributions $q_k$ are assumed to belong to the same family as the corresponding model distributions $p(z_k \mid \text{parents}(z_k))$.

The update rules are derived using coordinate ascent; that is, we repeatedly iterate over the latent variables and update them one at a time. In particular, we derive the updates by computing the functional derivative of the objective function with respect to one of the latent variables and solving for the value of the corresponding variational parameter which makes this functional derivative equal to zero. To obtain concrete update formulas, we can calculate the logarithm of the model's joint probability distribution and then perform the following steps for each latent variable $z_k$:

- Drop any constant terms in the log joint probability (that is, terms that do not depend on $z_k$).
- Express the remaining terms as a dot product between the sufficient statistics of the distribution $p(z_k \mid \text{parents}(z_k))$, which are functions of $z_k$, and a vector of natural parameters, denoted $\eta_k$, which do not depend on $z_k$.
- Compute the expectation of $\eta_k$ with respect to the variational distributions for the latent variables. That is, compute

$$E_q[\eta_k] = E_{z_1 \sim q_1}\left[\cdots E_{z_m \sim q_m}\left[\eta_k\right]\cdots\right] \tag{35}$$

- Compute the inverse of the natural parameter transformation corresponding to the variational distribution $q_k$, and apply it to the vector $E_q[\eta_k]$. This yields the concrete values with which to update the variational parameters corresponding to $z_k$.

We then repeat the above steps while continually monitoring the value of the ELBO until it appears to have converged. This procedure is summarized in Algorithm 2.

**Algorithm 2** Coordinate-Ascent Variational Inference

```
• Input:
    ▪ $\vec{x}$: an observed dataset
• Output:
    ▪ $(\nu_1, \dots, \nu_m)$: vector of optimized variational parameters corresponding to the latent variables
• Procedure:
  Initialize $(\nu_1, \dots, \nu_m)$ to a "reasonable" guess based on the input data. (This can be done using model-specific heuristics, combined with random noise for symmetry-breaking.)
  Throughout this algorithm, $q$ will be determined by the current value of $(\nu_1, \dots, \nu_m)$:
      $q(\vec{z}) = \prod_{k=1}^{m} q_k(z_k; \nu_k)$
  while ELBO has not converged do
      for each latent variable $z_k$, with $k \in \{1, \dots, m\}$ do
          Update $\nu_k \leftarrow \theta_k(E_{z\sim q}[\eta_k])$.
          Here, $\theta_k(\cdot)$ is the inverse parameter transform corresponding to $z_k$. Moreover, $\eta_k$ is the vector of natural parameters associated with the complete conditional $p(z_k \mid \vec{z}_{-k}, \vec{x})$.
      end for
      Compute ELBO according equation (26).
  end while
  return $(\nu_1, \dots, \nu_m)$
```

## 4.3 Stochastic Variational Inference

Stochastic variational inference (which was introduced in [14]) allows variational inference to scale to larger datasets. It does this by dividing the latent variables into "local" (i.e., per-document) and "global" (i.e., corpus-wide) latent variables. In this section, we will use the generic model illustrated in Figure 3 as a prototype, with $\beta$ denoting the global latent variables and $z_{1:N}$ denoting the local latent variables. Given a model of this structure, we can see the limitations of coordinate-ascent variational inference by considering how to compute the complete conditionals:

$$\text{For local variables:} \quad p(z_n \mid \alpha, \beta, z_{-n}, x_{1:N}) = p(z_n \mid \alpha, \beta, x_n) \tag{36}$$

$$\text{For global variables:} \quad p(\beta \mid \alpha, z_{1:N}, x_{1:N}) \tag{37}$$

In particular, note that if $\alpha$, $\beta$, and $x_n$ are given, then $z_n$ is conditionally independent of all other variables. Thus, the coordinate-ascent computations for local latent variables can be done efficiently, using one data point at a time. In contrast, a single update to the global latent variable $\beta$ requires computations involving the entire training dataset.

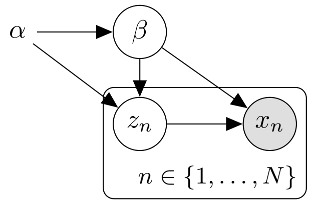

Figure 3: An illustration of the distinction between local and global latent variables in stochastic variational inference. Here, $\beta$ denotes the global latent variables, and $z_{1:N}$ denotes the local latent variables.

To address this inefficiency, stochastic variational inference uses repeated sampling of mini-batches to update the global variational parameters. The update rules for the local latent variables are the same as in coordinate-ascent variational inference, but the global latent variables are updated using the *natural gradient* of the objective function, estimated on small samples of the data.

Traditional optimization routines use a standard gradient; this make sense in cases where Euclidean distance suffices for measuring similarities between candidate maximizers. However, in the case of parameterized probability distributions, the Euclidean distance between pairs of parameter vectors may not accurately reflect the "true" difference between the corresponding distributions. Hoffman [14] gives an example involving a normal distribution parameterized by its mean and variance. If we start with a mean of $\mu = 0$ and a variance of $\sigma^2 = 0.01$, then adding 1 to the mean will have a relatively large effect, since $\text{Normal}(0, 0.01)$ has very little overlap with $\text{Normal}(1, 0.01)$. The situation is reversed if we start with $\mu = 0$ and $\sigma^2 = 1000$, since $\text{Normal}(0, 1000)$ is nearly identical to $\text{Normal}(1, 1000)$. Thus, relying on gradients defined in terms of Euclidean distance can lead to unnecessarily slow convergence, since some steps will essentially be wasted by moving in an inefficient direction.

To address this, we define a matrix $G$ which encodes information about how the "true" distance function varies within a local neighborhood. This matrix $G$ is also known as a *metric tensor*. When dealing with parameterized probability distributions, we can define $G$ in terms of a symmetric variant of KL divergence—in particular, we can use the arithmetic mean of KL divergence and reversed KL divergence. We omit the details of the calculation, but $G$ turns out to be equal to a matrix known as the *Fisher information matrix* of the variational distribution. Then, rather than simply following the gradient of the objective function in Euclidean coordinates (i.e., $\nabla\mathcal{L}$), the natural gradient takes steps in the direction of $G^{-1}\nabla\mathcal{L}$. For exponential-family models, the natural gradient update rule has a particularly simple form, since the inverse metric tensor $G^{-1}$ cancels out entirely in such cases. In particular, the natural gradient for the global parameters is

$$G^{-1}\nabla\mathcal{L} = E_q[\eta^{\text{global}}] - \mu. \tag{38}$$

Here, $\mu$ represents the current value of the variational parameters corresponding to the global variable $\beta$. Moreover, $\eta^{\text{global}}$ represents the exponential-family natural parameters corresponding to the complete conditional of the global variable, $p(\beta \mid \alpha, z_{1:N}, x_{1:N})$.

Now, instead of computing the natural gradient of the objective function based on the entire dataset, we can use stochastic optimization techniques to compute updates based on random subsets of the data called *mini-batches*. For each observation in each mini-batch, the local per-observation parameters are updated according to the traditional coordinate-ascent formulas. For the global parameters, only a small modification is needed: we compute natural gradient using a virtual dataset containing multiple copies of the current mini-batch. In particular, if the full dataset has $N$ observations, and a mini-batch $b$ contains $S$ observations, we perform updates for the global parameters using a virtual dataset consisting of $\lceil N/S \rceil$ copies of the mini-batch. As long as we base our updates on estimates which agree with the natural gradient *in expectation*, we will still achieve reasonable convergence; this same principle is also used in stochastic gradient descent. The update rule for the global parameters is then

$$\mu \leftarrow \mu + \rho_t G^{-1} \nabla \mathcal{L} \tag{39}$$

$$= (1 - \rho_t)\mu + \rho_t E_q[\eta_b^{\text{global}}] \tag{40}$$

Here $\{\rho_t\}_{t \geq 1}$ is a sequence of step-sizes satisfying $\sum_{t \geq 1} \rho_t = \infty$ and $\sum_{t \geq 1} \rho_t^2 < \infty$; for example, Hoffman [14] uses $\rho_t = (t + \tau)^{-\kappa}$, with $\tau \geq 0$ and $\kappa \in (0.5, 1]$. In addition, $\eta_b^{\text{global}}$ is essentially the same as $\eta^{\text{global}}$, except it is computed based on the virtual dataset derived from $\lceil N/S \rceil$ repetitions of the mini-batch $b$. The overall procedure for stochastic variational inference is outlined in Algorithm 3.

**Algorithm 3** Stochastic Variational Inference

- **Input:**
  - $\vec{x}$: dataset containing $N$ observations
- **Output:**
  - $\mu$: an optimized parameter corresponding to the global latent variables
  - $\nu$: a length-$N$ vector of optimized parameters corresponding to the local latent variables
- **Procedure:**

```
Initialize $\mu$ and $\nu$ to "reasonable" guesses based on the input data.
Let $\{\rho_t\}_{t \geq 1}$ be a sequence of step-sizes satisfying $\sum_{t \geq 1} \rho_t = \infty$ and $\sum_{t \geq 1} \rho_t^2 < \infty$.
Set $t \leftarrow 0$
while ELBO has not converged do
    Update $t \leftarrow t + 1$
    Randomly sample a mini-batch $b$ containing $S$ observations
    for each observation $x_i$ in $b$ do
        Update $\nu_i \leftarrow E_q[\eta_i^{\text{local}}]$
    end for
    Update $\mu \leftarrow (1 - \rho_t)\mu + \rho_t E_q[\eta_b^{\text{global}}]$
    Compute ELBO according equation (26).
end while
return $\mu$, $\nu$
```

## 5 Hierarchical Topic Models Based on Dirichlet Processes

Next, we discuss a line of research in which random variables known as Dirichlet Processes are used to learn a tree-structured hierarchy of topics from a corpus. The first such technique is known as the Nested Chinese Restaurant Process, and a subsequent enhancement to this technique is called the Nested Hierarchical Dirichlet Process.

### 5.1 The Nested Chinese Restaurant Process Topic Model

The NCRP topic model was defined in Griffiths et al [11] and Blei et al [3] and is capable of learning a hierarchy of topics. There is some ambiguity in the terminology used in the literature, as the term "NCRP" can refer to a probability distribution over paths in a tree, but it has also been used to refer to the particular nested topic model discussed in this section. To disambiguate these terms, we will typically use the term "NCRP distribution" to refer to the former, and "NCRP topic model" to refer to the latter.

Both the NCRP distribution and the NCRP topic model are based on a probability distribution known as the Chinese Restaurant Process (CRP), which we will discuss first.

### 5.1.1 The Chinese Restaurant Process: A Distribution Over Partitions

The CRP is a stochastic process which allows us to put a prior distribution on a potentially infinite partition of a dataset. Here, the term "stochastic process" refers to an infinite set of random variables which are indexed by some infinite set. With nonzero probability, the CRP can assign observations to arbitrarily large numbers of partitions, although for a finite dataset the size of the partition is of course bounded by the number of observations.

The CRP is based on a thought experiment involving a restaurant containing an infinite number of tables, each of which have an infinite capacity. The tables are indexed starting from 1, and an arbitrary number of customers sequentially arrive at the restaurant. With probability 1, the first customer is guaranteed to sit at the first table. All subsequent customers select a table according to the following rules:

- If $n$ is the total number of previously-seated customers at the restaurant, the $(n+1)^{\text{th}}$ customer will sit at the next empty table with probability $\frac{\alpha}{n+\alpha}$.
- If the first $k$ tables are occupied, with the $i^{\text{th}}$ table containing $m_i$ customers, the $(n+1)^{\text{th}}$ customer will sit at table $i$ with probability $\frac{m_i}{n+\alpha}$.

Here, $\alpha$ is a positive-valued parameter that affects whether there will be relatively few tables containing a large number of customers, or relatively many tables which each contain only a few customers. In the limit $\alpha \to 0$, there will be a single table with all of the customers. On the other hand, as $\alpha \to \infty$, the number of occupied tables increases.

Although this thought experiment gives us an intuitive sense for how to generate a random partition, it involves a sequence of random trials in which each trial is dependent on the outcomes of all previous trials. Thus, for the purpose of deriving an implementation, it is preferable to use a reformulation that involves independent random variables. One convenient method is to use a *stick-breaking construction*. We define an infinite sequence of beta-distributed random variables:

$$V_k \sim \text{Beta}(1, \alpha) \qquad \text{for all } k \geq 1. \tag{41}$$

Then, we define an infinite sequence of probabilities:

$$\pi_k = V_k \prod_{j=1}^{k-1} (1 - V_j). \tag{42}$$

We can view the infinite sequence $\{\pi_k\}_{k \geq 1}$ as defining a probability distribution over all positive integers. In the limit where the number of observations approaches infinity, the distribution over partitions defined by the CRP and by this stick-breaking construction are equivalent. Since the random variables $V_k$ are i.i.d., this formulation makes it easier to derive joint and conditional probability distributions.

### 5.1.2 The Dirichlet Process: A Distribution Over Grouped Data

In the context of topic models, the CRP can be used as a distribution over indices into an infinite i.i.d. set of topic vectors. In terms of the restaurant analogy, this corresponds to associating each table with an independent draw from some "base distribution" (in topic models, these draws are from a Dirichlet distribution); thus, selecting a table corresponds to selecting a topic vector. When the CRP is used in this way—specifically, as a distribution over indices into some countably infinite set of i.i.d. random variables—we may alternatively view it as a *Dirichlet process*. The Dirichlet process is a probability distribution over probability distributions. It is parameterized by a positive scalar parameter $\alpha$ and a base distribution, $G_0$. As for notation, we can write

$$X \sim \text{DP}(\alpha, G_0) \tag{43}$$

as shorthand for

$$\theta_k \sim G_0 \qquad \text{for } k \geq 1 \tag{44}$$

$$V_k \sim \text{Beta}(1, \alpha) \qquad \text{for } k \geq 1 \tag{45}$$

$$\pi_k = V_k \prod_{j=1}^{k-1} (1 - V_j) \qquad \text{for } k \geq 1 \tag{46}$$

$$X = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \tag{47}$$

Here, the notation $X = \sum_k \pi_k \delta_{\theta_k}$ means that $X$ is a discrete distribution over the values $\{\theta_k : k \geq 1\}$. In particular, $R \sim X$ implies that $R = \theta_k$ with probability $\sum_{\ell:\theta_k=\theta_\ell} \pi_\ell$, for each $k \geq 1$. Note that each $\theta_k$ is a member of the domain of the distribution $G_0$, and $X$ is a probability distribution whose domain is a subset of the domain of $G_0$. Furthermore, $V_k$ and $\pi_k$ are scalars between 0 and 1, with the sequence $\{\pi_k\}$ satisfying $\sum_k \pi_k = 1$. This formulation will be useful in later sections, when we explore the Hierarchical Dirichlet Process (HDP) and the Nested Hierarchical Dirichlet Process (NHDP).

The name "Dirichlet process" comes from the fact that, for any finite partition $(A_1, \ldots, A_n)$ of the domain of $G_0$, the vector $(X(A_1), \ldots, X(A_n))$ is disributed according to a Dirichlet distribution with parameters $(\alpha G_0(A_1), \ldots, \alpha G_0(A_n))$. Here, the notation $X(A_i)$ or $G_0(A_i)$ represents the probability mass which $X$ (or $G_0$, respectively) assigns to the set $A_i$. The fact that such a distribution exists was proved in Ferguson [8], although this result was merely an existential proof. Sethuraman [20] later provided the explicit stick-breaking construction for the Dirichlet process described above.

The preceding formulation can be modified slightly if we express it in terms of a so-called GEM distribution (GEM stands for Griffiths, Engen, and McCloskey). A GEM distribution is a distribution over distributions over positive integers, defined such that

$$\vec{\pi} \sim \text{GEM}_2(\gamma_1, \gamma_2) \tag{48}$$

is equivalent to

$$V_k \sim \text{Beta}(\gamma_1, \gamma_2) \qquad \text{for } k \geq 1 \tag{49}$$

$$\pi_k = V_k \prod_{j=1}^{k-1} (1 - V_j) \qquad \text{for } k \geq 1 \tag{50}$$

Note in this context that $\vec{\pi}$ is an infinite sequence; that is, $\vec{\pi} = \{\pi_k\}_{k \geq 1}$. Furthermore, this sequence sums to 1 and can therefore be viewed as a discrete distribution over the set of positive integers. The subscript "2" indicates the this is a two-parameter variant of the GEM distribution. The single-parameter variant is equivalent to setting the first parameter to 1. That is, $\text{GEM}(\alpha) = \text{GEM}_2(1, \alpha)$. With these definitions, a Dirichlet process $X \sim \text{DP}(\alpha, G_0)$ is equivalent to

$$\theta_k \sim G_0 \qquad \text{for } k \geq 1 \tag{51}$$

$$\vec{\pi} \sim \text{GEM}(\alpha) \tag{52}$$

$$X = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \tag{53}$$

### 5.1.3 The Nested Chinese Restaurant Process: A Distribution Over Paths

Next, we define the Nested Chinese Restaurant Process (NCRP) in terms of the CRP. Whereas the CRP is a distribution over partitions, the NCRP is a distribution over paths in a tree. Continuing the restaurant analogy, suppose we have an infinite number of restaurants, each with an infinite number of infinite-capacity tables. There is one restaurant which is singled out as the "root" restaurant, and every customer begins by selecting a restaurant from this table according to the rules of the (non-nested) CRP. Each table has a card containing the address of another restaurant. After all customers at the current restaurant have been seated, the customers at each table simultaneously get up and move to the restaurant which is marked on the card at their table. Then, this process repeats at a new set of restaurants. No restaurant appears on a card more than once, which means that there is one and only one way to reach each restaurant.

We can imagine each restaurant as a node in an infinitely wide, infinitely deep tree. Each node can be indexed by a tuple representing the sequence of table selections required to reach that restaurant. For example, the empty sequence corresponds to the restaurant at the root node, and the sequence $(3, 1)$ corresponds to the first descendant of the third descendant of the root node. That is, $(3, 1)$ corresponds to the restaurant we would reach if we picked the third table at the root-node restaurant, and then the first table at the next restaurant.

In this formulation, a single draw from the NCRP corresponds to a distribution over infinite paths in this tree. Equivalently, we can think of a single NCRP draw as a probability distribution over infinite sequences of positive integers. Drawing an infinite sequence $\{a_i\}_{i\geq 1}$ corresponds to choosing table $a_1$ from the first restaurant, entering the restaurant corresponding to $a_1$, choosing table $a_2$ from that restaurant, entering the restaurant corresponding to $a_2$, choosing table $a_3$ from that restaurant, and so on.

Like the CRP, the NCRP can also be defined in terms of a stick-breaking construction. To do this, we use an infinite collection of beta-distributed random variables, indexed by the set of finite-length paths in an infinite tree:

$$V_r \sim \text{Beta}(1, \alpha), \tag{54}$$

where $r$ is any non-empty, finite-length tuple of positive integers. As a special case, the random variable associated with the root node, $V_{()}$, is always equal to $1$, rather than being beta-distributed. Then, we recursively define a set of variables as follows:

$$\pi_{()} = V_{()} = 1 \tag{55}$$

$$\pi_{r[1:\ell]} = \pi_{r[1:\ell-1]} \cdot \left( V_{r[1:\ell]} \prod_{j=1}^{r[\ell]-1} (1 - V_{r[1:\ell-1],j}) \right) \tag{56}$$

Expanding this recurrence relation yields:

$$\begin{aligned} \pi_r &= \prod_{\ell=1}^{\text{len}(r)} \left( V_{r[1:\ell]} \prod_{j=1}^{r[\ell]-1} \left(1 - V_{r[1:\ell-1],j}\right) \right) & (57) \\ &= \left( \prod_{\ell=1}^{\text{len}(r)} V_{r[1:\ell]} \right) \left( \prod_{\ell=1}^{\text{len}(r)} \prod_{j=1}^{r[\ell]-1} \left(1 - V_{r[1:\ell-1],j}\right) \right). & (58) \end{aligned}$$

If $r$ is a finite-length tuple, then $\pi_r$ represents the probability of drawing a path which contains the prefix $r$. Fixing $V_{()} = \pi_{()} = 1$ corresponds to the fact that every path contains the empty tuple as a prefix. Furthermore, we note the similarity between this formulation and equation (42); in the NCRP, we are essentially repeating the CRP at each node in an infinite tree.

The preceding discussion uses a tree of infinite depth. However, it is also possible to use only a finite number of layers in the tree (that is, to truncate the tree at a finite depth). Specifically, if we truncate the tree at depth $L$, then each full-length path in the tree—equivalently, each leaf node in the tree—corresponds to a tuple of positive integers of length $L$. Note that even if we truncate the tree in terms of its depth, it still contains an infinite number of paths, since the width of the tree is infinite.

### 5.1.4 The NCRP Topic Model

Now we have the necessary ingredients to describe how the NCRP topic model is defined in terms of the NCRP distribution. In the NCRP topic model, a single draw from the NCRP distribution is used to define a global probability distribution over paths. Moreover, for each node in the infinite tree, a discrete distribution over the vocabulary is drawn from a Dirichlet distribution. To each document we assign a single path through the tree, as well as a distribution over "levels of abstraction" (i.e., a distribution over depths within the tree). Then, within each document, each word is drawn using a two step process: first, a depth $z$ is selected from the distribution over depths specific to the current document; next, a vocabulary word is drawn from the $z^{\text{th}}$ topic along the sampled path corresponding to the current document.

Thus, the generative process for sampling documents according to the NCRP topic model is as follows:

- For each node in the infinite tree, generate a "topic," which consists of a distribution over the vocabulary, by drawing from a Dirichlet. To avoid actually computing an infinite number of topics, this step can be performed via lazy evaluation. That is, we can generate and store a new topic only when a subsequent step requires it.
- Draw a distribution over paths from the NCRP. In concrete terms, this means we must draw a beta-distributed random variable for each node in the infinite tree. Again, this can be done via lazy evaluation.
- For each document:
  - Draw a distribution over depths, using a 2-parameter GEM distribution. (Alternatively, in the case where the depth has been truncated to some finite value $L$, the distribution over depths can be drawn from a $(L+1)$-component Dirichlet.)
  - Draw a path through the global tree, using the global distribution over paths.
  - For each word in the document:
    - Draw a depth $z$ from the distribution-over-depths associated with this document.
    - Draw a vocabulary word from the topic associated with the $z^{\text{th}}$ element of the path associated with this document.

Here are the conditional probability distributions for this model:

$$\theta_r \sim \text{Dirichlet}(\alpha^{(\theta)}) \tag{59}$$

$$V_r \sim \text{Beta}(1, \alpha^{(V)}) \tag{60}$$

$$\phi_d \sim \text{GEM}_2(\alpha_1^{(\phi)}, \alpha_2^{(\phi)}) \tag{61}$$

$$z_{d,n} \sim \text{Categorical}(\phi_d) \tag{62}$$

$$c_d \sim \sum_{\substack{r\text{:path} \\ \text{of length } L}} \pi_r \delta_r \tag{63}$$

$$t_{d,n} \sim \text{Categorical}(\theta[c_d[1:z_{d,n}]]) \tag{64}$$

where

$$\pi_r = \prod_{\ell=1}^{\text{len}(r)} \left( V_{r[1:\ell]} \prod_{j=1}^{r[\ell]-1} (1 - V_{r[1:\ell-1],j}) \right) \tag{65}$$

In the case where the depth of the tree has been truncated to $L$, we draw the distributions-over-depths $\phi_d$ from a Dirichlet with $L+1$ components, instead of from a GEM distribution. We use $L+1$ components because topics can be drawn from anywhere between the root (at depth 0) and the leaves (at depth $L$). Hence, in the finite-depth case, we would substitute the following distribution for $\phi_d$:

$$\phi_d \sim \text{Dirichlet}(\alpha^{(\phi)}) \tag{66}$$

The plate diagram for this graphical model is shown in Figure 4.

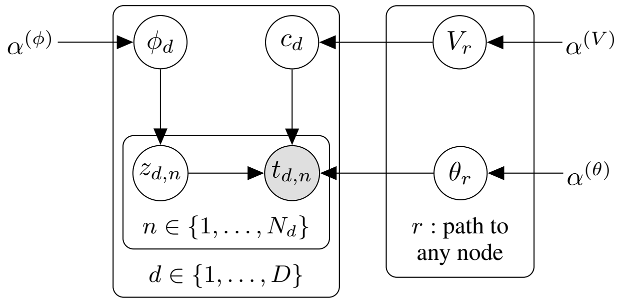

Figure 4: Plate diagram for the NCRP topic model

### 5.1.5 Gibbs Sampling for the NCRP

Papers such as [11] and [3] used NCRP inference algorithms based on Gibbs sampling.

Initially, Griffiths et al [11] used collapsed Gibbs sampling—integrating out the depth proportions $\phi_d$ and the topic vectors $\theta_k$— and a finite-depth tree. Moreover, to avoid explicitly modeling an infinitely wide tree, they made use of the order-dependent "restaurant analogy" formulation of the NCRP. Thus, each Gibbs sampling step had the ability to grow or shrink the width of the tree.

Later, Blei et al [3] focused on the Bayesian non-parametric nature of the NCRP, offering a way to handle an infinite-depth tree. They again used collapsed Gibbs sampling, integrating out $\phi_d$ and $\theta_k$. To handle the infinite-depth tree, they essentially used lazy evaluation. At any given point, only a finite number of layers $L$ were explicitly modeled, but if the deepest layer was ever sampled, a new layer would be added for subsequent iterations.

### 5.1.6 Variational Inference for the NCRP

A variational inference algorithm for the NCRP model was proposed in Wang et al [22]. The procedure described in section 4.2 applies to traditional finite-dimensional models; however, in the context of Bayesian non-parametric models such as the NCRP, the latent variables belong to an infinite-dimensional space. This means that there are an infinite number of variational parameters that would need to be updated in each iteration. Hence, in order to obtain a feasible algorithm, it is necessary to make additional approximations.

The approach taken in [22] is to start with a finite version of the tree which has been truncated in terms of both its depth and its width. Then, any latent variables associated with the nodes that lie outside of this finite truncation are assumed to be equal to the corresponding prior distribution; thus, outside of the finite truncation, there is no dependence on any variational parameters. With the variational distributions assumed to be of this truncated form, it becomes possible to compute a probability distribution over equivalence classes of paths. Paths which reach the deepest layer of the tree each belong to their own equivalence class; since the truncation is finite, there are a finite number of such singleton classes. On the other hand, paths which are not entirely included in the the truncated tree are grouped into infinitely large equivalence classes according to the nodes at which they exit the truncated tree. The authors then use this distribution over equivalence classes to implement greedy heuristics for adding, pruning, and merging paths in the finite tree. For example, if the expected number of observations that fall into one of the truncated equivalence classes grows large enough, the algorithm responds by adding a new path to the finite tree.

## 5.2 The Nested Hierarchical Dirichlet Process Topic Model

The primary limitation of the NCRP has to do with the fact that each document is associated with only a single path through the topic hierarchy. This creates problems when the training corpus contains documents with a variety of "hybrid" topics. From the example mentioned earlier, there may be some documents which contain a mixture of the topics *sports* and *medicine*, and the relative proportions of these topics may vary depending on the specific focus of the document. To accommodate such documents, the NCRP leaves us with limited options:

1. Treat *sports medicine* as a subtopic of either *sports* or *medicine*.
2. Treat *sports medicine* as a top-level topic, distinct from both *sports* and *medicine*.

Option 1 may lead to unmanageably deep topic hierarchies (or, if we are using the finite-depth variant of NCRP, this option may not even be feasible). On the other hand, option 2 may lead to unmanageably wide topic hierarchies, particularly if a large number of documents contain hybrids of two or more topics. In either case, it becomes difficult to represent the topic hierarchy as a relatively small tree.

Rather than face these limited options, we would prefer to augment the model so that it can better handle a variety of hybrid topics. The solution that was proposed in [16] is a topic model called the Nested Hierarchical Dirichlet Process (NHDP). The NHDP topic model shares many features in common with the NCRP topic model:

- Once again, we draw a global distribution over paths from an NCRP distribution.

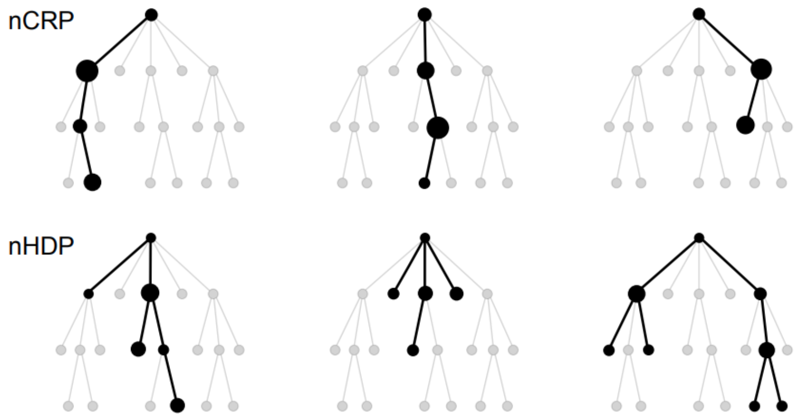

Figure 5: Comparison of the NCRP topic model (top row) and the NHDP topic model (bottom row). Within each row are several examples of document-specific distributions over nodes. (Source: Paisley et al [16])

- As before, there is an infinite set of topic vectors (i.e., distributions over vocabulary words) which are arranged in an infinite tree and are independently drawn from a Dirichlet distribution.
- Furthermore, each document is associated with a "local" distribution over nodes in the infinite tree of topics.

The main difference between these two topic models is the way in which the local distributions over nodes are defined. In the NCRP topic model, each document's local distribution consists of nodes along a single path, but in the NHDP, the local distributions can include multiple branches. This distinction is illustrated in Figure 5.

To see how the local, per-document distributions are defined in the NHDP, we must first discuss the (non-nested) Hierarchical Dirichlet Process (HDP). The HDP was introduced in [21] to model grouped data, in which there are an unbounded (potentially infinite) number of groups, each of which contains multiple observations and is associated with its own vector of parameters. A simple, non-nested HDP model can be formulated as follows:

$$H \sim \text{DP}(\alpha, G_0) \tag{67}$$
$$H_d \sim \text{DP}(\beta, H) \tag{68}$$
$$\mu_{d,n} \sim H_d \tag{69}$$
$$x_{d,n} \sim F(\mu_{d,n}) \tag{70}$$

In this formulation, there is a base distribution $G_0$ (such as a Gaussian or a Dirichlet), and a "global" distribution $H$ is drawn from a Dirichlet process applied to $G_0$. The distribution $H$ is discrete, which means that it can be viewed as a distribution over countably many independent draws from the base distribution. Next, per-group distributions $H_d$ (one for each group $d$ in the data) are drawn from another Dirichlet process, which is applied to the global distribution $H$. For each data point $n$ in group $d$, a parameter vector $\mu_{d,n}$ is drawn from the group-specific distribution $H_d$. Finally, the observed data points $x_{d,n}$ are drawn according to the distribution $F(\mu_{d,n})$, where $F$ is some parameterized distribution, such as a Gaussian or a categorical distribution. In the context of topic models, a "group" is often defined to be a single document, and the individual observations within each group are the words within the corresponding document. The NHDP model uses $G_0 = \text{Dirichlet}(\gamma)$ and $F(\cdot) = \text{Categorical}(\cdot)$.

To define a per-document distribution over nodes, the NHDP uses an HDP defined at each node of the infinite topic tree. However, it also interleaves a document-specific tree of "path propagation" random variables to determine, for each word, whether to extend the current path by drawing the next subtopic or to stop at the current node and draw a specific vocabulary word.

The generative process is as follows:

1. For each node $r$ in the infinite tree, draw a topic vector $\theta_r$, representing a distribution over the vocabulary. Since this involves an infinite number of random draws, this step is performed via lazy evaluation.
2. For each node $r$ and each $j \geq 1$, draw a beta-distributed stick-breaking proportion $V^*_{r,j}$ via lazy evaluation.
3. For each document $d$:
   (a) For each node $r$ in the infinite tree, perform the following steps via lazy evaluation:
      i. For each $j \geq 1$:
         A. Draw an index $z^d_{r,j} \geq 1$ according to a discrete distribution with probabilities given by $\pi_k = V^*_{r,k} \prod_{i=1}^{k-1}(1 - V^*_{r,i})$.
         B. Draw a beta-distributed stick-breaking proportion $V^d_{r,j}$.
      ii. Draw a beta-distributed path-propagation random variable $U^d_r$.
   (b) For each word-slot $n$ in document $d$:
      i. Let $r = ()$, the empty path.
      ii. Repeat the following steps until a value for $c^d_n$ is chosen:
         A. Draw a Bernoulli random variable with probability of success $U^d_r$.
         B. If the outcome is 1, then set $c^d_n = r$ and break out of this loop; otherwise, continue.
         C. For all $j \geq 1$ (via lazy evaluation), find all indices $k$ such that $z^d_{r,k} = j$. Let $S_j$ be the set of such indices, and define $\mu_j = \sum_{k \in S_j} V^d_{r,k} \prod_{i=1}^{k-1}(1 - V^d_{r,i})$.
         D. Draw an index $\tilde{z} \geq 1$ from a discrete probability distribution with probabilities given by $\mu_j$ as defined above.
         E. Extend $r$ by appending $\tilde{z}$ to it; that is, update $r \leftarrow (r, \tilde{z})$.
      iii. Draw a vocabulary word $t^d_n$ from a categorical distribution with probabilities given by $\theta_{c^d_n}$.

In summary, we first draw an infinite, global tree of topics ($\theta_r$) and stick-breaking proportions ($V^*_{r,j}$). Then, for each document $d$, we draw local sequences of indices ($z^d_{r,j}$) based on the global stick-breaking proportions. We also draw an independent set of document-specific stick-breaking proportions ($V^d_{r,j}$). Together, $z^d_{r,j}$ and $V^d_{r,j}$ determine how the topics descended from node $r$ are rearranged and re-weighted in a document-specific fashion. Furthermore, we draw "path-propagation" probabilities $U^d_r$ for each document, and for each node in the infinite tree. By combining $z^d_{r,j}$, $V^d_{r,j}$, and $U^d_r$, we have a per-document distribution over an infinite tree of nodes. For each word $n$ in a given document $d$, we draw a node $c^d_n$ from this document-specific distribution over nodes; this identifies the topic from which the nth word of document $d$ will be drawn. Finally, we draw a vocabulary word $t^d_n$ from Categorical($\theta_{c^d_n}$).

To visualize how the NHDP model is constructed, Figure 6 contains a plate diagram of the Bayesian network underlying this model. Figure 7 shows the corresponding conditional probability distributions.

### 5.2.1 Variational Inference for the NHDP

The inference algorithm for the NHDP topic model is based on stochastic variational inference. Once again, a naive implementation of variational inference is not possible, since there are an infinite number of latent variables. The authors solve this by restricting the model to a tree with both finite depth and finite width. As long as the finite tree is deep and wide enough to accommodate any tree that falls within the anticipated range of sizes, then the algorithm will be able to function as intended.

Further simplifications involve the variational updates for the index variables $z^d_{r,j}$, which determine how branches from the global tree are rearranged in a document-specific way. First, the variables $z^d_{r,j}$ are given Dirac-$\delta$ variational distributions, which corresponds to making a hard assignment for each $z$. (This goes against the typical convention of choosing variational distributions which agree with the corresponding conditional.) Second, for any value of $d$ and $r$, the indices $z^d_{r,j}$ are assumed not to repeat; this corresponds to assuming that the document-specific index assignments only rearrange branches and never duplicate branches. Third, for each document $d$, a greedy algorithm is used

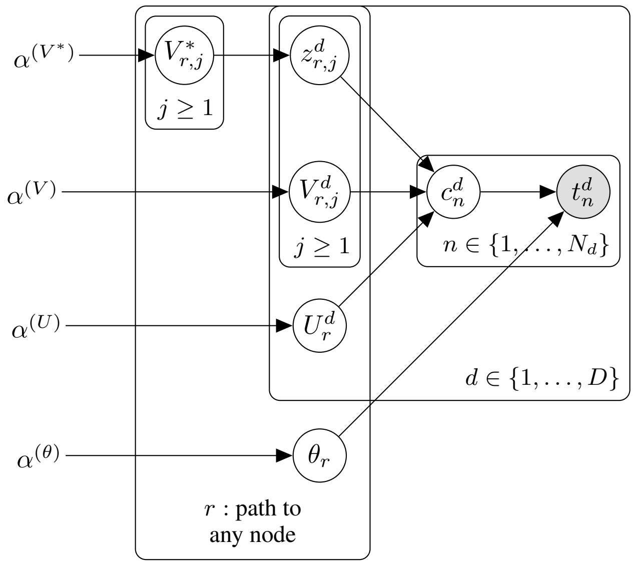

Figure 6: Plate diagram for the NHDP topic model

$$
\begin{aligned}
\theta_r &\sim \text{Dirichlet}(\alpha^{(\theta)}) \\
V^*_{r,j} &\sim \text{Beta}(1, \alpha^{(V^*)}) \\
V^d_{r,j} &\sim \text{Beta}(1, \alpha^{(V)}) \\
U^d_r &\sim \text{Beta}(\alpha_1^{(U)}, \alpha_2^{(U)}) \\
z^d_{r,j} &\sim \sum_{k \geq 1} \left( V^*_{r,k} \prod_{i=1}^{k-1} (1 - V^*_{r,i}) \right) \delta_k \\
c^d_n &\sim \sum_{r:\text{path}} A(r, V^d, z^d)\, B(r, U^d)\, \delta_r \\
A(r, V^d, z^d) &= \prod_{m=0}^{\text{len}(r)-1} \sum_{k \geq 1} \mathbb{1} \left[ z^d_{r[1:m],k} = r[m+1] \right] \left( V^d_{r[1:m],k} \prod_{i=1}^{k-1} \left( 1 - V^d_{r[1:m],i} \right) \right) \\
B(r, U^d) &= U^d_r \prod_{m=0}^{\text{len}(r)-1} \left( 1 - U^d_{r[1:m]} \right) \\
t^d_n &\sim \text{Categorical}(\theta_{c^d_n})
\end{aligned}
$$

Figure 7: Conditional probability distributions for the NHDP topic model

to select only a small number of values for $z^d_{r,j}$; this corresponds to selecting a small subtree for each document. This greedy algorithm starts with just the root node, and adds new nodes only if doing so would increase the objective function (the ELBO) by a predetermined amount. With these simplifications in place, the remainder of the update rules follow the derivations in section 4.3.

## 6 Directions for Future Research

We can identify some directions for future research by considering some of the limitations of the NHDP topic model.

**Scalable algorithms:** The currently available set of inference algorithms force us to make a significant tradeoff between precision and scalability. On the one hand, we have Gibbs sampling, which can theoretically compute the true posterior distribution with arbitrary precision, but which requires requires many passes over the training data and therefore fails to scale well to massive datasets. On the other hand, we have stochastic variational inference, which uses a more efficient mini-batch approach and has deterministic convergence; however, it requires us to make major approximating assumptions, such as the mean-field approximation. To improve the inference algorithm, we could perhaps look for a variant of Gibbs sampling that requires fewer passes over the training data. Alternatively, we might seek a modified version of stochastic variational inference which makes less stringent assumptions on the form of the variational distributions. Yet another approach might be to extend the non-probabilistic, geometric "anchor words" approach of Arora et al [1] in order to handle hierarchies of topics.

**Interpreting models:** The models discussed above generally represent topics as discrete probability distributions over the entire vocabulary. However, a typical vocabulary may contain tens of thousands of words, which makes the vector of topic probabilities unwieldy and difficult to interpret. A common approach is to visualize a topic by displaying its top $K$ most likely vocabulary words. However, within a large topic hierarchy, a small value of $K$ might not provide enough information to disambiguate closely related topics, whereas a large value may lead to information overload. By comparison, human-curated topic hierarchies such as library cataloguing systems often use a very small number of words to describe each node in a large hierarchy. Hence, it might be useful to develop techniques to post-process the results of these models in order to derive a concise yet meaningful description of each topic.

**Incorporating human feedback:** In cases where the model's quality is deemed subpar, it may be useful to be able to incorporate human feedback. This would require not only an investigation of the types of mistakes which these topic models are prone to make in practice, but also an investigation into how best to design a user interface that is capable of gathering useful feedback. It may also be useful to apply semi-supervised or multi-label classification techniques in scenarios where partial or even contradictory labels are available.

**Moving beyond the bag-of-words model:** All of the models discussed above ignore the order in which words appear in a document. The sequence of words in a document can influence its meaning, and therefore its classification within a topic hierarchy. By making use of word-order information, we could potentially improve the quality of the results.

**Frameworks for Bayesian non-parametric inference:** Deriving inference algorithms for Bayesian non-parametric models, such as the ones discussed above, is a laborious process which is difficult to automate. For instance, even though the NCRP and NHDP topic models share many features, their inference algorithms must be derived in a model-specific way. This is made especially difficult by the fact that, in Bayesian non-parametric models, there can be an infinite number of latent variables. This requires us to augment traditional inference algorithms with model-specific heuristics. An approach that combines black-box inference techniques (such as those described in [17] and [18]) with model-agnostic heuristics for handling infinite-dimensional latent variables might make it feasible to efficiently explore and compare many variants of these models.

## References

[1] Sanjeev Arora, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning*, pages 280–288, 2013.

[2] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012. Online version available at `http://www.cs.ucl.ac.uk/staff/d.barber/brml/`; Draft dated 2017-02-02.

[3] David M Blei, Thomas L Griffiths, and Michael I Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.

[4] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, (just-accepted), 2017.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[6] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[7] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[8] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[9] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Springer, 2001.

[10] Samuel J Gershman and David M Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

[11] Thomas L Griffiths, Michael I Jordan, Joshua B Tenenbaum, and David M Blei. Hierarchical topic models and the nested chinese restaurant process. In *Advances in neural information processing systems*, pages 17–24, 2004.

[12] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[13] Gregor Heinrich. Parameter estimation for text analysis. *Technical report*, 2005.

[14] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[15] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[16] John Paisley, Chong Wang, David M Blei, and Michael I Jordan. Nested hierarchical dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270, 2015.

[17] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

[18] Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator variational inference. In *Advances in Neural Information Processing Systems*, pages 496–504, 2016.

[19] Philip Resnik and Eric Hardisty. Gibbs sampling for the uninitiated. Technical report, University of Maryland, College Park, 2010.

[20] Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.

[21] Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392, 2005.

[22] Chong Wang and David M Blei. Variational inference for the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2009.

[23] Wikipedia. Exponential family. `https://en.wikipedia.org/w/index.php?title=Exponential_family&oldid=787816251`. Accessed September 2017.

[24] Wikipedia. Latent dirichlet allocation. `https://en.wikipedia.org/w/index.php?title=Latent_Dirichlet_allocation&oldid=797823717`. Accessed September 2017.

[25] Wikipedia. Latent semantic analysis. `https://en.wikipedia.org/w/index.php?title=Latent_semantic_analysis&oldid=798597246`. Accessed September 2017.

[26] Wikipedia. Probabilistic latent semantic analysis. `https://en.wikipedia.org/w/index.php?title=Probabilistic_latent_semantic_analysis&oldid=783155225`. Accessed September 2017.