Efficient Methods for Inferring Large Sparse Topic Hierarchies

Doug Downey, Chandra Sekhar Bhagavatula, Yi Yang
Electrical Engineering and Computer Science
Northwestern University
ddowney@eecs.northwestern.edu,{csb,yiyang}@u.northwestern.edu

Abstract

Latent variable topic models such as Latent Dirichlet Allocation (LDA) can discover topics from text in an unsupervised fashion. However, scaling the models up to the many distinct topics exhibited in modern corpora is challenging. "Flat" topic models like LDA have difficulty modeling sparsely expressed topics, and richer hierarchical models become computationally intractable as the number of topics increases.

In this paper, we introduce efficient methods for inferring large topic hierarchies. Our approach is built upon the Sparse Backoff Tree (SBT), a new prior for latent topic distributions that organizes the latent topics as leaves in a tree. We show how a document model based on SBTs can effectively infer accurate topic spaces of over a million topics. We introduce a collapsed sampler for the model that exploits sparsity and the tree structure in order to make inference efficient. In experiments with multiple data sets, we show that scaling to large topic spaces results in much more accurate models, and that SBT document models make use of large topic spaces more effectively than flat LDA.

1 Introduction

Latent variable topic models, such as Latent Dirichlet Allocation (Blei et al., 2003), are popular approaches for automatically discovering topics in document collections. However, learning models that capture the large numbers of distinct topics expressed in today's corpora is challenging. While efficient methods for learning large topic models have been developed (Li et al., 2014; Yao et al., 2009; Porteous et al., 2008), these methods have focused on "flat" topic models such as LDA. Flat topic models over large topic spaces are prone to overfitting: even in a Web-scale corpus, some words are expressed rarely, and many documents are brief. Inferring a large topic distribution for each word and document given such sparse data is challenging. As a result, LDA models in practice tend to consider a few thousand topics at most, even when training on billions of words (Mimno et al., 2012).

A promising alternative to flat topic models is found in recent *hierarchical* topic models (Paisley et al., 2015; Blei et al., 2010; Li and McCallum, 2006; Wang et al., 2013; Kim et al., 2013; Ahmed et al., 2013). Topics of words and documents can be naturally arranged into hierarchies. For example, an article on the topic of the *Chicago Bulls* is also relevant to the more general topics of *NBA*, *Basketball*, and *Sports*. Hierarchies can combat data sparsity: if data is too sparse to place the term "Pau Gasol" within the Chicago Bulls topic, perhaps it can be appropriately modeled at somewhat less precision within the Basketball topic. A hierarchical model can make fine-grained distinctions where data is plentiful, and back-off to more coarse-grained distinctions where data is sparse. However, current hierarchical models are hindered by computational complexity. The existing inference methods for the models have runtimes that increase at least linearly with the number of topics, making them intractable on large corpora with large numbers of topics.

In this paper, we present a hierarchical topic model that can scale to large numbers of distinct topics. Our approach is built upon a new prior for latent topic distributions called a Sparse Backoff Tree (SBT). SBTs organize the latent topics as leaves in a tree, and smooth the distributions for each topic with those of similar topics nearby in the tree. SBT priors use absolute discounting and learned backoff distributions for

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 774–784,
Beijing, China, July 26-31, 2015.

smoothing sparse observation counts, rather than the fixed additive discounting utilized in Dirichlet and Chinese Restaurant Process models. We show how the SBT's characteristics enable a novel collapsed sampler that exploits the tree structure for efficiency, allowing SBT-based document models (SBTDMs) that scale to hierarchies of over a million topics.

We perform experiments in text modeling and hyperlink prediction, and find that SBTDM is more accurate compared to LDA and the recent nested Hierarchical Dirichlet Process (nHDP) (Paisley et al., 2015). For example, SBTDMs with a hundred thousand topics achieve perplexities 28-52% lower when compared with a standard LDA configuration using 1,000 topics. We verify that the empirical time complexity of inference in SBTDM increases sub-linearly in the number of topics, and show that for large topic spaces SBTDM is more than an order of magnitude more efficient than the hierarchical Pachinko Allocation Model (Mimno et al., 2007) and nHDP. Lastly, we release an implementation of SBTDM as open-source software.[1]

2 Previous Work

The intuition in SBTDM that topics are naturally arranged in hierarchies also underlies several other models from previous work. Paisley et al. (2015) introduce the nested Hierarchical Dirichlet Process (nHDP), which is a tree-structured generative model of text that generalizes the nested Chinese Restaurant Process (nCRP) (Blei et al., 2010). Both the nCRP and nHDP model the tree structure as a random variable, defined over a flexible (potentially infinite in number) topic space. However, in practice the infinite models are truncated to a maximal size. We show in our experiments that SBTDM can scale to larger topic spaces and achieve greater accuracy than nHDP. To our knowledge, our work is the first to demonstrate a hierarchical topic model that scales to more than one million topics, and to show that the larger models are often much more accurate than smaller models. Similarly, compared to other recent hierarchical models of text and other data (Petinot et al., 2011; Wang et al., 2013; Kim et al., 2013; Ahmed et al., 2013; Ho et al., 2010), our focus is on scaling to larger data sets and topic spaces.

[1] http://websail.cs.northwestern.edu/projects/sbts/

The Pachinko Allocation Model (PAM) introduced by Li & McCallum (Li and McCallum, 2006) is a general approach for modeling correlations among topic variables in latent variable models. Hierarchical organizations of topics, as in SBT, can be considered as a special case of a PAM, in which inference is particularly efficient. We show that our model is much more efficient than an existing PAM topic modeling implementation in Section 5.

Hu and Boyd-Graber (2012) present a method for augmenting a topic model with known hierarchical correlations between words (taken from e.g. WordNet synsets). By contrast, our focus is on automatically learning a hierarchical organization of topics from data, and we demonstrate that this technique improves accuracy over LDA. Lastly, SparseLDA (Yao et al., 2009) is a method that improves the efficiency of inference in LDA by only generating portions of the sampling distribution when necessary. Our collapsed sampler for SBTDM utilizes a related intuition at each level of the tree in order to enable fast inference.

3 Sparse Backoff Trees

In this section, we introduce the Sparse Backoff Tree, which is a prior for a multinomial distribution over a latent variable. We begin with an example showing how an SBT transforms a set of observation counts into a probability distribution. Consider a latent variable topic model of text documents, similar to LDA (Blei et al., 2003) or pLSI (Hofmann, 1999). In the model, each token in a document is generated by first sampling a discrete latent topic variable Z from a document-specific topic distribution, and then sampling the token's word type from a multinomial conditioned on Z.

We will focus on the document's distribution over topics, ignoring the details of the word types for illustration. We consider a model with 12 latent topics, denoted as integers from the set $\{1, \ldots, 12\}$. Assume we have assigned latent topic values to five tokens in the document, specifically the topics $\{1, 4, 4, 5, 12\}$. We indicate the number of times topic value z has been selected as n_z (Figure 1).

Given the five observations, the key question faced by the model is: what is the topic distribution over a sixth topic variable from the same document? In the case of the Dirichlet prior utilized for the topic distribution in LDA, the probability

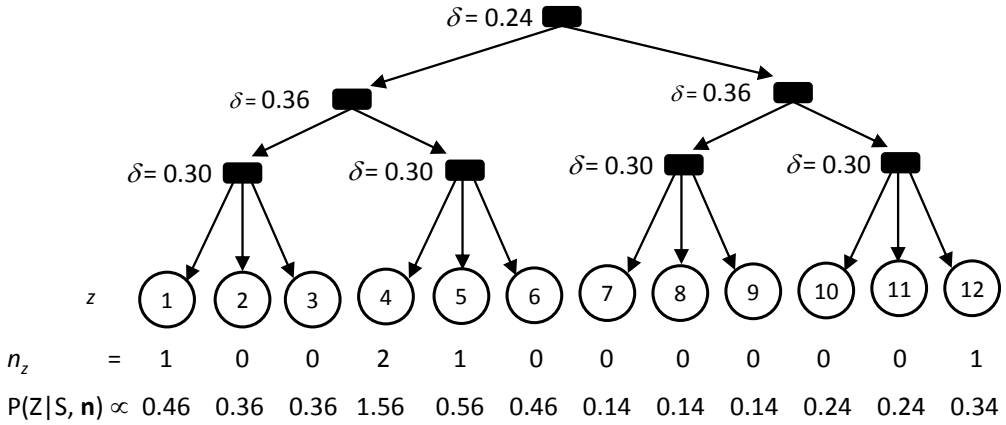

Figure 1: An example Sparse Backoff Tree over 12 latent variable values.

that the sixth topic variable has value z is proportional to $n_z + \alpha$, where α is a hyperparameter of the model.

SBT differs from LDA in that it organizes the topics into a tree structure in which the topics are leaves (see Figure 1). In this paper, we assume the tree structure, like the number of latent topics, is manually selected in advance. With an SBT prior, the estimate of the probability of a topic z is increased when nearby topics in the tree have positive counts. Each interior node a of the SBT has a *discount* δ_a associated with it. The SBT transforms the observation counts n_z into pseudo-counts (shown in the last row in the figure) by subtracting δ_a from each non-zero descendent of each interior node a, and re-distributing the subtracted value uniformly among the descendants of a. For example, the first state has a total of 0.90 subtracted from its initial count $n_1 = 1$, and then receives $0.30/3$ from its parent, $1.08/6$ from its grandparent, and $0.96/12$ from the root node for a total pseudo-count of 0.46. The document's distribution over a sixth topic variable is then proportional to these pseudo-counts.

When each document tends to discuss a set of related topics, the SBT prior will assign a higher likelihood to the data when related topics are located nearby in the tree. Thus, by inferring latent variable values to maximize likelihood, nearby leaves in the tree will come to represent related topics. SBT, unlike LDA, encodes the intuition that a topic becomes more likely in a document that also discusses other, related topics. In the example, the pseudo-count the SBT produces for topic six (which is related to other topics that occur in the document) is almost three times larger than that of topic eight, even though the observation counts are zero in each case. In LDA, topics six and eight would have equal pseudo-counts (proportional to α).

3.1 Definitions

Let Z be a discrete random variable that takes integer values in the set $\{1, \ldots, L\}$. Z is drawn from a multinomial parameterized by a vector θ of length L.

Definition 1 *A Sparse Backoff Tree $SBT(\mathcal{T}, \delta^\theta, Q(z))$ for the discrete random variable Z consists of a rooted tree $\mathcal{T}$ containing L leaves, one for each value of Z; a coefficient $\delta_a > 0$ for each interior node a of $\mathcal{T}$; and a backoff distribution $Q(z)$.*

Figure 1 shows an example SBT. The example includes simplifications we also utilize in our experiments, namely that all nodes at a given depth in the tree have the same number of children and the same δ value. However, the inference techniques we present in Section 4 are applicable to any tree $\mathcal{T}$ and set of coefficients $\{\delta_a\}$.

For a given SBT S, let $\Delta_S(z)$ indicate the sum of all δ_a values for all ancestors a of leaf node z (i.e., all interior nodes on the path from the root to z). For example, in the figure, $\Delta_S(z) = 0.90$ for all z. This amount is the total absolute discount that the SBT applies to the random variable value z.

We define the SBT prior implicitly in terms of the posterior distribution it induces on a random variable Z drawn from a multinomial θ with an SBT prior, after θ is integrated out. Let the vector $\mathbf{n} = [n_1, \ldots, n_L]$ denote the sufficient statistics for any given observations drawn from θ, where n_z is the number of times value z has been observed. Then, the distribution over a subsequent draw of Z

given SBT prior S and observations $\mathbf{n}$ is defined as:

$$P(Z = z|S, \mathbf{n}) \equiv \frac{\max(n_z - \Delta_S(z), 0) + B(S, z, \mathbf{n})Q(z)}{K(S, \sum_i n_i)} \quad (1)$$

where $K(S, \sum_i n_i)$ is a normalizing constant that ensures the distribution sums to one for any fixed number of observations $\sum_i n_i$, and $B(S, z, \mathbf{n})$ and $Q(z)$ are defined as below.

The quantity $B(S, z, \mathbf{n})$ expresses how much of the discounts from all other leaves z' contribute to the probability of z. For an interior node a, let $desc(a)$ indicate the number of leaves that are descendants of a, and let $desc^+(a)$ indicate the number of leaf descendants z of a that have non-zero values n_z. Then the contribution of the discount δ_a of node a to *each* of its descendent leaves is $b(a, \mathbf{n}) = \delta_a desc^+(a)/desc(a)$. We then define $B(S, z, \mathbf{n})$ to be the sum of $b(a, \mathbf{n})$ over all interior nodes a on the path from the root to z.

The function $Q(z)$ is a *backoff distribution*. It allows the portion of the discount probability mass that is allocated to z to vary with a proposed distribution $Q(z)$. This is useful because in practice the SBT is used as a prior for a conditional distribution, for example the distribution $P(Z|w)$ over topic Z given a word w in a topic model of text. In that case, an estimate of $P(Z|w)$ for a rare word w may be improved by "mixing in" the marginal topic distribution $Q(z) = P(Z = z)$, analogous to backoff techniques in language modeling. Our document model described in the following section utilizes two different Q functions, one uniform ($Q(z) = 1/T$) and another related to the marginal topic distribution $P(z)$.

4 The SBT Document Model

We now present the SBT document model, a probabilistic latent variable model of text documents that utilizes SBT priors. We then provide a collapsed sampler for the model that exploits the tree structure to make inference more efficient.

Our document model follows the Latent Dirichlet Allocation (LDA) Model (Blei et al., 2003), illustrated graphically in Figure 2 (left). In LDA, a corpus of documents is generated by sampling a topic distribution θ_d for each document d, and also a distribution over words ϕ_z for each topic. Then, in document d each token w is generated by first sampling a topic z from the multinomial $P(Z|\theta_d)$, and then sampling w from the multinomial $P(W|Z, \phi_z)$.

The SBTDM is the same as LDA, with one significant difference. In LDA, the parameters θ and ϕ are sampled from two Dirichlet priors, with separate hyperparameters α and β. In SBTDM, the parameters are instead sampled from particular SBT priors. Specifically, the generative model is:

$$\begin{aligned} \theta &\sim SBT(\mathcal{T}, \delta^\theta, Q_\theta(z) = 1/T) \\ \phi' &\sim SBT(\mathcal{T}, \delta^\phi, Q_\phi(z) = P^*(z)) \\ \lambda &\sim \text{Dirichlet}(\alpha') \\ Z|\theta &\sim \text{Discrete}(\theta) \\ W|z, \phi', \lambda &\sim \text{Discrete}(\lambda\phi'_{\cdot,z}/P(z|\phi')) \end{aligned}$$

The variable ϕ' represents the distribution of topics given words, $P(Z|W)$. The SBTDM samples a distribution ϕ'_w over topics for each word type w in the vocabulary (of size V). In SBTDM, the random variable ϕ'_w has dimension L, rather than V for ϕ_z as in LDA. We also draw a prior word frequency distribution, $\lambda = \{\lambda_w\}$ for each word w. [2] We then apply Bayes Rule to obtain the conditional distributions $P(W|Z, \phi')$ required for inference. The expression $\lambda\phi'_{\cdot,z}/P(z|\phi')$ denotes the normalized element-wise product of two vectors of length V: the prior distribution λ over words, and the vector of probabilities $P(z|w) = \phi'_{w,z}$ over words w for the given topic z.

The SBT priors for ϕ' and θ share the same tree structure $\mathcal{T}$, which is fixed in advance. The SBTs have different discount factors, denoted by the hyperparameters δ^θ and δ^ϕ. Finally, the backoff distribution for θ is uniform, whereas ϕ's backoff distribution P^* is defined below.

4.1 Backoff distribution $P^*(z)$

SBTDM requires choosing a backoff distribution $P^*(z)$ for ϕ'. As we now show, it is possible to select a natural backoff distribution $P^*(z)$ that enables scalable inference.

Given a set of observations $\mathbf{n}$, we will set $P^*(z)$ proportional to $P(z|S^\phi, \mathbf{n})$. This is a recursive definition, because $P(z|S^\phi, \mathbf{n})$ depends on $P^*(z)$. Thus, we define:

$$P^*(z) \equiv \frac{\sum_w \max(n_z^w - \Delta_S(z), 0)}{C - \sum_w B_w(S^\phi, z, \mathbf{n})} \quad (2)$$

[2] The word frequency distribution does not impact the inferred topics (because words are always observed), and in our experiments we simply use maximum likelihood estimates for λ_w (i.e., setting α' to be negligibly small). Exploring other word frequency distributions is an item of future work.

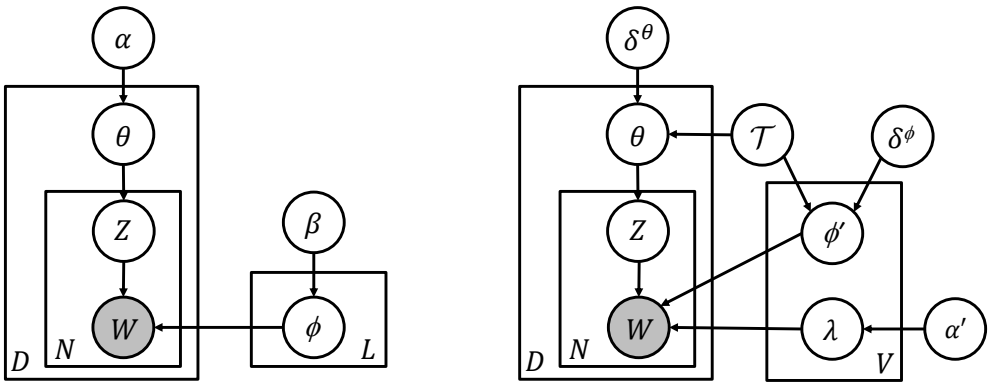

Figure 2: The Latent Dirichlet Allocation Model (left) and our SBT Document Model (right).

where $C > \sum_w B_w(S^\phi, z, \mathbf{n})$ is a hyperparameter, n_z^w is the number of observations of topic z for word w in $\mathbf{n}$, and B_w indicates the function $B(S^\phi, z, \mathbf{n})$ defined in Section 3.1, for the particular word w. That is, $\sum_w B_w(S^\phi, z, \mathbf{n})$ is the total quantity of smoothing distributed to topic z across all words, *before* the backoff distribution $P^*(z)$ is applied.

The form of $P^*(z)$ has two key advantages. The first is that setting $P^*(z)$ proportional to the marginal topic probability allows SBTDM to back-off toward marginal estimates, a successful technique in language modeling (Katz, 1987) (where it has been utilized for word probabilities, rather than topic probabilities). Secondly, setting the backoff distribution in this way allows us to simplify inference, as described below.

Algorithm 1 Compute the sampling distribution for a product of two multinomials with SBT priors with $Q(z) = 1$

```
function INTERSECT(SBT Node a_r, SBT Node a_l)
    if a_r, a_l are leaves then
        τ(i) ← τ(a_r)τ(a_l)
        return i
    end if
    i.r ← a_r
    r(i) ← b(a_l) * τ(a_r)
    i.l ← a_l ; b(i.l) ← 0
    l(i) ← b(a_r) * τ(a_l) − b(a_r)b(a_l)desc(a_r)
    τ(i)+ = r(i) + l(i)
    for all child c non-zero for a_r and a_l do
        i_c ← INTERSECT(a_r.c, a_l.c)
        i.children += i_c
        τ(i) += τ(i_c)
    end for
    return i
end function
```

4.2 Inference with Collapsed Sampling

Given a corpus of documents D, we infer the values of the hidden variables Z using the collapsed Gibbs sampler popular in Latent Dirichlet Allocation models (Griffiths and Steyvers, 2004). Each variable Z_i is sampled given the settings of all other variables (denoted as $\mathbf{n}_{-i}$):

$$P(Z_i = z|\mathbf{n}_{-i}, D) \propto P(z|\mathbf{n}_{-i}, \mathcal{T}, \delta^\theta) \cdot P(w_i|z, \mathbf{n}_{-i}, \mathcal{T}, \delta^\phi) \quad (3)$$

The first term on the right-hand side is given by Equation 1. The second can be rewritten as:

$$P(w_i|z, \mathbf{n}_{-i}, \mathcal{T}, \delta^\phi) = \frac{P(z, w_i|\mathbf{n}_{-i}, \mathcal{T}, \delta^\phi)}{P(z|\mathbf{n}_{-i}, \mathcal{T}, \delta^\phi)} \quad (4)$$

4.3 Efficient Inference Implementation

The primary computational cost when scaling to large topic spaces involves constructing the sampling distribution. Both LDA with collapsed sampling and SBTDM share an advantage in space complexity: the model parameters are specified by a sparse set of non-zero counts denoting how often tokens of each word or document are assigned to each topic. However, in general the sampling distribution for SBTDM has non-uniform probabilities for each of L different latent variable values. Thus, even if many parameters are zero, a naive approach that computes the complete sampling distribution will still take time linear in L.

However, in SBTs the sampling distribution can be constructed efficiently using a simple recursive algorithm that exploits the structure of the tree. The result is an inference algorithm that often requires far less than linear time in L, as we verify in our experiments.

First, we note that $P(z, w_i|\mathbf{n}_{-i}, \mathcal{T}, \delta^\phi)$ is proportional to the sum of two quantities: the discounted count $\max(n_z - \Delta_S, 0)$ and the smoothing probability mass $B(S, z, \mathbf{n})Q(z)$. By choosing $Q(z) = P^*(z)$, we can be ensured that $P^*(z)$ normalizes this sum. Thus, the backoff distri-

bution cancels through the normalization. This means we can normalize the SBT for ϕ' in advance by scaling the non-zero counts by a factor of $1/P^*(z)$, and then at inference time we need only multiply pointwise two multinomials with SBT priors and *uniform* backoff distributions.

The intersection of two multinomials drawn from SBT priors with uniform backoff distributions can be performed efficiently for sparse trees. The algorithm relies on two quantities defined for each node of each tree. The first, $b(a, \mathbf{n})$, was defined in Section 3. It denotes the smoothing that the interior node a distributes (uniformly) to each of its descendent leaves. We denote $b(a, \mathbf{n})$ as $b(a)$ in this section for brevity. The second quantity, $\tau(a)$, expresses the total count mass of all leaf descendants of a, *excluding* the smoothing from ancestors of a.

With the quantities $b(a)$ and $\tau(a)$ for all a, we can efficiently compute the sampling distribution of the product of two SBT-governed multinomials (which we refer to as an SBTI). The method is shown in Algorithm 1. It takes two SBT nodes as arguments; these are corresponding nodes from two SBT priors that share the same tree structure $\mathcal{T}$. It returns an SBTI, a data structure representing the sampling distribution.

The efficiency of Algorithm 1 is reflected in the fact that the algorithm only recurses for child nodes c with non-zero $\tau(c)$ for *both* of the SBT node arguments. Because such cases will be rare for sparse trees, often Algorithm 1 only needs to traverse a small portion of the original SBTs in order to compute the sampling distribution exactly. Our experiments illustrate the efficiency of this algorithm in practice.

Finally, we can efficiently sample from either an SBTI or a single SBT-governed multinomial. The sampling methods are straightforward recursive methods, supplied in Algorithms 2 and 3.

Algorithm 2 Sample(SBT Node a)

```
procedure SAMPLE(SBT Node a)
    if a is a leaf then return a
    end if
    Sample from {b(a)desc(a), τ(a) − b(a)desc(a)}.
    if back-off distribution b(a)desc(a) selected then
        return Uniform[a's descendents]
    else
        Sample a's child c ∼ τ(c)
        return SAMPLE(c)
    end if
end procedure
```

Algorithm 3 Sampling from an SBTI

```
function SAMPLE(SBTI Node i)
    if i is a leaf then return i
    end if
    Sample from {r(i), l(i), τ(i) − r(i) − l(i)}
    if right distribution r(i) selected then
        return SAMPLE(i.r)
    else
        if left distribution l(i) selected then
            return SAMPLE(i.l)
        else
            Sample i's child c ∼ τ(c)
            return SAMPLE(c)
        end if
    end if
end function
```

4.4 Expansion

Much of the computational expense encountered in inference with SBTDM occurs shortly after initialization. After a slow first several sampling passes, the conditional distributions over topics for each word and document become concentrated on a sparse set of paths through the SBT. From that point forward, sampling is faster and requires much less memory.

We utilize the hierarchical organization of the topic space in SBTs to side-step this computational complexity by adding new leaves to the SBTs of a trained SBTDM. This is a "coarse-to-fine" (Petrov and Charniak, 2011) training approach that we refer to as *expansion*. Using expansion, the initial sampling passes of the larger model can be much more time and space efficient, because they leverage the already-sparse structure of the smaller trained SBTDM.

Our expansion method takes as input an inferred sampling distribution $\mathbf{n}$ for a given tree $\mathcal{T}$. The algorithm adds k new branches to each leaf of $\mathcal{T}$ to obtain a larger tree $\mathcal{T}'$. We then transform the sampling state by replacing each $n_i \in \mathbf{n}$ with one of its children in $\mathcal{T}'$. For example, in Figure 1, expanding with $k = 3$ would result in a new tree containing 36 topics, and the single observation of topic 4 in $\mathcal{T}$ would be re-assigned randomly to one of the topics $\{10, 11, 12\}$ in $\mathcal{T}'$.

5 Experiments

We now evaluate the efficiency and accuracy of SBTDM. We evaluate SBTs on two data sets, the RCV1 Reuters corpus of newswire text (Lewis et al., 2004), and a distinct data set of Wikipedia links, WPL. We consider two disjoint subsets of RCV1, a small training set (RCV1s) and a larger

training set (RCV1).

We compare the accuracy and efficiency of SBTDM against flat LDA and two existing hierarchical models, the Pachinko Allocation Model (PAM) and nested Hierarchical Dirichlet Process (nHDP).

To explore how the SBT structure impacts modeling performance, we experiment with two different SBTDM configurations. **SBTDM-wide** is a shallow tree in which the branching factor increases from the root downward in the sequence $3, 6, 6, 9, 9, 12, 12$. Thus, the largest model we consider has $3 \cdot 6 \cdot 6 \cdot 9 \cdot 9 \cdot 12 \cdot 12 = 1{,}259{,}712$ distinct latent topics. **SBTDM-tall** has lower branching factors of either 2 or 3 (so in our evaluation its depth ranges from 3 to 15). As in SBTDM-wide, in SBTDM-tall the lower branching factors occur toward the root of the tree. We vary the number of topics by considering balanced subtrees of each model. For nHDP, we use the same tree structures as in SBT-wide. In preliminary experiments, using the tall structure in nHDP yielded similar accuracy but was somewhat slower.

Similar to our LDA implementation, SBTDM optimizes hyperparameter settings as sampling proceeds. We use local beam search to choose new hyperparameters that maximize leave-one-out likelihood for the distributions $P(Z|d)$ and $P(Z|w)$ on the training data, evaluating the parameters against the current state of the sampler.

We trained all models by performing 100 sampling passes through the full training corpus (i.e., approximately 10 billion samples for RCV1, and 8 billion samples for WPL). We evaluate performance on held-out test sets of 998 documents for RCV1 (122,646 tokens), and 200 documents for WPL (84,610 tokens). We use the left-to-right algorithm (Wallach et al., 2009) over a randomized word order with 20 particles to compute perplexity. We re-optimize the LDA hyperparameters at regular intervals during sampling.

5.1 Small Corpus Experiments

We begin with experiments over a small corpus to highlight the efficiency advantages of SBTDM.

Data Set	Tokens	Vocabulary	Documents
RCV1s	2,669,093	46,130	22,149
RCV1	101,184,494	283,911	781,262
WPL	82,154,551	1,141,670	199,000

Table 1: Statistics of the three training corpora.

As argued above, existing hierarchical models require inference that becomes expensive as the topic space increases in size. We illustrate this by comparing our model with PAM and nHDP. We also compare against a fast "flat" LDA implementation, SparseLDA, from the MALLET software package (McCallum, 2002).

For SBTDM we utilize a parallel inference approach, sampling all variables using a fixed estimate of the counts $\mathbf{n}$, and then updating the counts after each full sampling pass (as in (Wang et al., 2009)). The SparseLDA and nHDP implementations are also parallel. All parallel methods use 15 threads. The PAM implementation provided in MALLET is single-threaded.

Our efficiency measurements are shown in Figure 3. We plot the mean wall-clock time to perform 100 sampling passes over the RCV1s corpus, starting from randomly initialized models (i.e. *without* expansion in SBTDM). For the largest plotted topic sizes for PAM and nHDP, we estimate total runtime using a small number of iterations. The results show that SBTDM's time to sample increases well below linear in the number of topics. Both SBTDM methods have runtimes that increase at a rate substantially below that of the square root of the number of topics (plotted as a blue line in the figure for reference). For the largest numbers of topics in the plot, when we increase the number of topics by a factor of 12, the time to sample increases by less than a factor of 1.7 for both SBT configurations.

We also evaluate the accuracy of the models on the small corpus. We do not compare against PAM, as the MALLET implementation lacks a method for computing perplexity for a PAM model. The results are shown in Table 3. The SBT models tend to achieve lower perplexity than LDA, and SBTDM-tall performs slightly better than SBTDM-wide for most topic sizes. The best model, SBT-wide with 8,748 topics, achieves perplexity 14% lower than the best LDA model and 2% lower than the best SBTDM-tall model. The LDA model overfits for the largest topic configuration, whereas at that size both SBT models remain at least as accurate as any of the LDA models in Table 3.

We also evaluated using the topic coherence measure from (Mimno et al., 2011), which reflects how well the learned topics reflect word co-occurrence statistics in the training data. Follow-

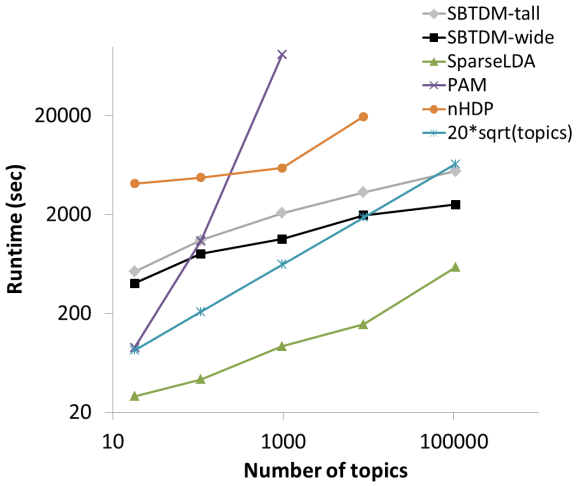

Figure 3: Time (in seconds) to perform a sampling pass over the RCV1s corpus as number of topics varies, plotted on a log-log scale. The SBT models scale sub-linearly in the number of topics.

ing recent experiments with the measure (Stevens et al., 2012), we use $\epsilon = 10^{-12}$ pseudo-co-occurrences of each word pair and we evaluate the average coherence using the top 10 words for each topic. Table 2 shows the results. PAM, LDA, and nHDP have better coherence at smaller topic sizes, but SBT maintains higher coherence as the number of topics increases.

Topics	LDA	PAM	nHDP	SBTDM -wide	SBTDM -tall
18	-420.8	-421.2	-422.9	-444.3	-440.2
108	-434.8	-430.9	-554.3	-445.4	-445.8
972	-451.2	-	-548.1	-443.3	-443.8
8748	-615.3	-	-	-444.3	-444.1

Table 2: Average topic coherence on the small RCV1s corpus.

5.1.1 Evaluating Expansion

The results discussed above do not utilize expansion in SBTDM. To evaluate expansion, we performed separate experiments in which we expanded a 972-topic model trained on RCV1s to initialize a 8,748-topic model. Compared to random initialization, expansion improved efficiency and accuracy. Inference in the expanded model executed approximately 30% faster and used 70% less memory, and the final 8,748-topic models had approximately 10% lower perplexity.

5.2 Large Corpus Results

Our large corpus experiments are reported in Table 4. Here, we compare the test set perplexity of a single model for each topic size and model type. We focus on SBTDM-tall for the large corpora. We utilize expansion (see Section 4.4) for SBTDM-tall models with more than a thousand topics on each data set. The results show that on both data sets, SBTDM-tall utilizes larger numbers of topics more effectively. On RCV1, LDA improves only marginally between 972 and 8,748 topics, whereas SBTDM-tall improves dramatically. For 8,748 topics, SBTDM-tall achieves a perplexity score 17% lower than LDA model on RCV1, and 29% lower on WPL. SBT improves even further in larger topic configurations. Training and testing LDA with our implementation using over a hundred thousand topics was not tractable on our data sets due to space complexity (the MALLET implementation exceeded our maximum 250G of heap space). As discussed above, expansion enables SBTDM to dramatically reduce space complexity for large topic spaces.

The results highlight the accuracy improvements found from utilizing larger numbers of topics than are typically used in practice. For example, an SBTDM with 104,976 topics achieves perplexity 28-52% lower when compared with a standard LDA configuration using only 1,000 topics.

	RCV1		WPL	
# Topics	LDA	SBTDM-tall	LDA	SBTDM-tall
108	**1,121**	1,148	**7,049**	7,750
972	**820**	841	2,598	**2,095**
8,748	772	**637**	1,730	**1,236**
104,976	-	593	-	1,242
1,259,712	-	626	-	-

Table 4: Model accuracy on large corpora (corpus perplexity measure). The SBT model utilizes larger numbers of topics more effectively.

5.3 Exploring the Learned Topics

Lastly, we qualitatively examine whether the SBTDM's learned topics reflect meaningful hierarchical relationships. From an SBTDM of 104,976 topics trained on the Wikipedia links data set, we examined the first 108 leaves (these are contained in a single subtree of depth 5). 760 unique terms (i.e. Wikipedia pages) had positive counts for the topics, and 500 of these terms were related to radio stations.

The leaves appear to encode fine-grained subcategorizations of the terms. In Figure 4, we provide examples from one subtree of six topics (topics 13-18). For each topic t, we list the top three

	Number of Topics				
Model	18	108	972	8,748	104,976
LDA	**1420** (16.3)	**1016** (9.8)	844 (1.8)	845 (3.3)	1058 (4.1)
nHDP	1433 (19.6)	1446 (53.3)	1583 (157.7)	-	-
SBTDM-wide	1510 (31.5)	1091 (31.8)	797 (3.5)	**723** (18.2)	844 (60.1)
SBTDM-tall	1480 (13.5)	1051 (9.1)	**787** (10.5)	736 (3.2)	**776** (14.1)

Table 3: Small training corpus (RCV1s) performance. Shown is perplexity averaged over three runs for each method and number of topics, with standard deviation in parens. Both SBTDM models achieve lower perplexity than LDA and nHDP for the larger numbers of topics.

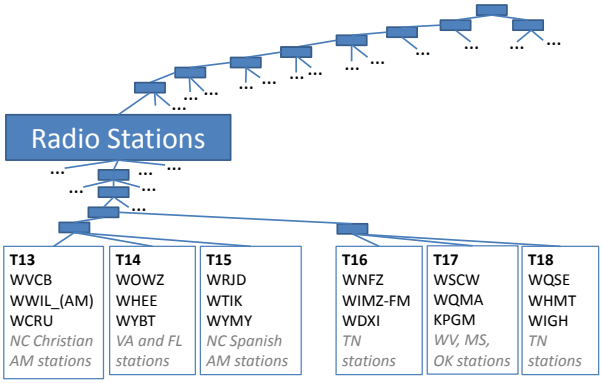

Figure 4: An example of topics from a 104,976-topic SBTDM defined over Wikipedia pages.

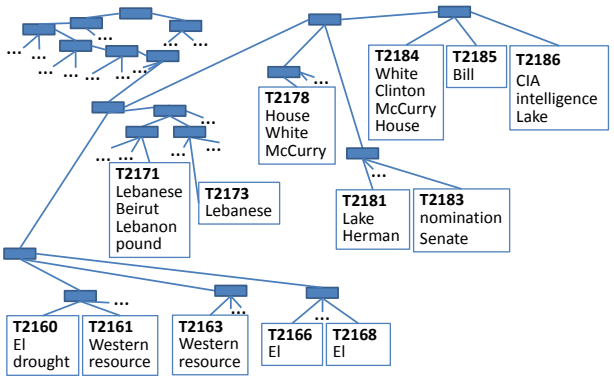

Figure 5: An example of topics from an 8,748-topic SBTDM defined over the RCV1 corpus.

terms w (ranked by symmetric conditional probability, $P(w|t)P(t|w)$), and a specific categorization that applies to the three terms. Interestingly, as shown in the figure, the top terms for the six topics we examined were all four-character "call letters" for particular radio stations. Stations with similar content or in nearby locations tend to cluster together in the tree. For example, the two topics focused on radio stations in Tennessee (TN) share the same parent, as do the topics focused on North Carolina (NC) AM stations. More generally, all six topics focus on radio stations in the southern US.

Figure 5 shows a different example, from a model trained on the RCV1 corpus. In this example, we first select only those terms that occur at least 2,000 times in the corpus and have a statistical association with their topic that exceeds a threshold, and we again rank terms by symmetric conditional probability. The 27-topic subtree detailed in the figure appears to focus on terms from major storylines in United States politics in early 1997, including El Niño, Lebanon, White House Press Secretary Mike McCarry, and the Senate confirmation hearings of CIA Director nominee Tony Lake.

6 Conclusion and Future Work

We introduced the Sparse Backoff Tree (SBT), a prior for latent topic distributions that organizes the latent topics as leaves in a tree. We presented and experimentally analyzed a document model based on the SBT, called an SBTDM. The SBTDM was shown to utilize large topic spaces more effectively than previous techniques.

There are several directions of future work. One limitation of the current work is that the SBT is defined only implicitly. We plan to investigate explicit representations of the SBT prior or related variants. Other directions include developing methods to learn the SBT structure from data, as well as applying the SBT prior to other tasks, such as sequential language modeling.

Acknowledgments

This research was supported in part by NSF grants IIS-1065397 and IIS-1351029, DARPA contract D11AP00268, and the Allen Institute for Artificial Intelligence. We thank the anonymous reviews for their helpful comments.

References

[Ahmed et al.2013] Amr Ahmed, Liangjie Hong, and Alexander Smola. 2013. Nested chinese restaurant franchise process: Applications to user tracking and document modeling. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1426–1434.

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

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

[Griffiths and Steyvers2004] Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235.

[Ho et al.2010] Qirong Ho, Ankur P Parikh, Le Song, and Eric P Xing. 2010. Infinite hierarchical mmsb model for nested communities/groups in social networks. *arXiv preprint arXiv:1010.1868*.

[Hofmann1999] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM.

[Hu and Boyd-Graber2012] Yuening Hu and Jordan Boyd-Graber. 2012. Efficient tree-based topic modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 275–279. Association for Computational Linguistics.

[Katz1987] Slava Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 35(3):400–401.

[Kim et al.2013] Suin Kim, Jianwen Zhang, Zheng Chen, Alice Oh, and Shixia Liu. 2013. A hierarchical aspect-sentiment model for online reviews. In *Proceedings of AAAI*.

[Lewis et al.2004] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397.

[Li and McCallum2006] Wei Li and Andrew McCallum. 2006. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 577–584, New York, NY, USA. ACM.

[Li et al.2014] Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. 2014. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 891–900. ACM.

[McCallum2002] Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

[Mimno et al.2007] David Mimno, Wei Li, and Andrew McCallum. 2007. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th international conference on Machine learning*, pages 633–640. ACM.

[Mimno et al.2011] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics.

[Mimno et al.2012] David Mimno, Matt Hoffman, and David Blei. 2012. Sparse stochastic inference for latent dirichlet allocation. *arXiv preprint arXiv:1206.6425*.

[Paisley et al.2015] J. Paisley, C. Wang, D.M. Blei, and M.I. Jordan. 2015. Nested hierarchical dirichlet processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(2):256–270, Feb.

[Petinot et al.2011] Yves Petinot, Kathleen McKeown, and Kapil Thadani. 2011. A hierarchical model of web summaries. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 670–675. Association for Computational Linguistics.

[Petrov and Charniak2011] Slav Petrov and Eugene Charniak. 2011. *Coarse-to-fine natural language processing*. Springer Science & Business Media.

[Porteous et al.2008] Ian Porteous, Arthur Asuncion, David Newman, Padhraic Smyth, Alexander Ihler, and Max Welling. 2008. Fast collapsed gibbs sampling for latent dirichlet allocation. In *In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 569–577.

[Stevens et al.2012] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961. Association for Computational Linguistics.

[Wallach et al.2009] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1105–1112. ACM.

[Wang et al.2009] Yi Wang, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y Chang. 2009. Plda: Parallel latent dirichlet allocation for large-scale applications. In *Algorithmic Aspects in Information and Management*, pages 301–314. Springer.

[Wang et al.2013] Chi Wang, Marina Danilevsky, Nihit Desai, Yinan Zhang, Phuong Nguyen, Thrivikrama Taula, and Jiawei Han. 2013. A phrase mining framework for recursive construction of a topical hierarchy. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 437–445, New York, NY, USA. ACM.

[Yao et al.2009] Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 937–946. ACM.