Tree-Structured Stick Breaking for Hierarchical Data

Ryan Prescott Adams[*]
Dept. of Computer Science
University of Toronto

Zoubin Ghahramani
Dept. of Engineering
University of Cambridge

Michael I. Jordan
Depts. of EECS and Statistics
University of California, Berkeley

Abstract

Many data are naturally modeled by an unobserved hierarchical structure. In this paper we propose a flexible nonparametric prior over unknown data hierarchies. The approach uses nested stick-breaking processes to allow for trees of unbounded width and depth, where data can live at any node and are infinitely exchangeable. One can view our model as providing infinite mixtures where the components have a dependency structure corresponding to an evolutionary diffusion down a tree. By using a stick-breaking approach, we can apply Markov chain Monte Carlo methods based on slice sampling to perform Bayesian inference and simulate from the posterior distribution on trees. We apply our method to hierarchical clustering of images and topic modeling of text data.

1 Introduction

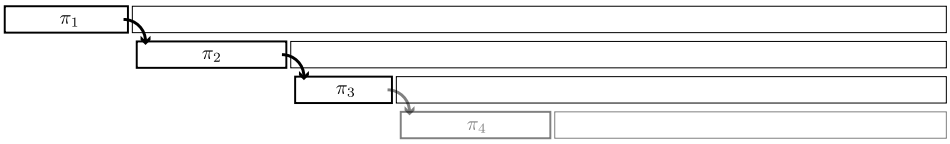
Structural aspects of models are often critical to obtaining flexible, expressive model families. In many cases, however, the structure is unobserved and must be inferred, either as an end in itself or to assist in other estimation and prediction tasks. This paper addresses an important instance of the structure learning problem: the case when the data arise from a latent hierarchy. We take a direct nonparametric Bayesian approach, constructing a prior on tree-structured partitions of data that provides for unbounded width and depth while still allowing tractable posterior inference.

Probabilistic approaches to latent hierarchies have been explored in a variety of domains. Unsupervised learning of densities and nested mixtures has received particular attention via finite-depth trees [1], diffusive branching processes [2] and hierarchical clustering [3, 4]. Bayesian approaches to learning latent hierarchies have also been useful for semi-supervised learning [5], relational learning [6] and multi-task learning [7]. In the vision community, distributions over trees have been useful as priors for figure motion [8] and for discovering visual taxonomies [9].

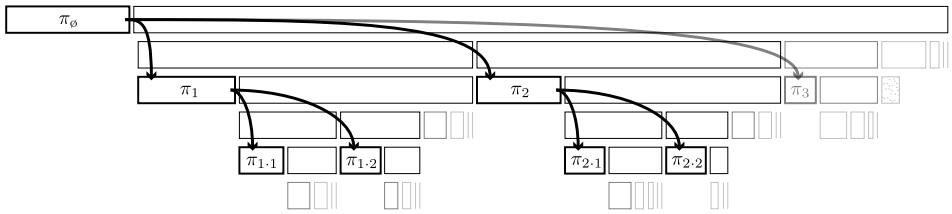
In this paper we develop a distribution over probability measures that imbues them with a natural hierarchy. These hierarchies have unbounded width and depth and the data may live at internal nodes on the tree. As the process is defined in terms of a distribution over probability measures and not as a distribution over data per se, data from this model are infinitely exchangeable; the probability of any set of data is not dependent on its ordering. Unlike other infinitely exchangeable models [2, 4], a pseudo-time process is not required to describe the distribution on trees and it can be understood in terms of other popular Bayesian nonparametric models.

Our new approach allows the components of an infinite mixture model to be interpreted as part of a diffusive evolutionary process. Such a process captures the natural structure of many data. For example, some scientific papers are considered *seminal* — they spawn new areas of research and cause new papers to be written. We might expect that within a text corpus of scientific documents, such papers would be the natural ancestors of more specialized papers that followed on from the new ideas. This motivates two desirable features of a distribution over hierarchies: 1) ancestor data (the

[*] http://www.cs.toronto.edu/~rpa/



(a) Dirichlet process stick breaking



(b) Tree-structured stick breaking

Figure 1: a) Dirichlet process stick-breaking procedure, with a linear partitioning. b) Interleaving two stick-breaking processes yields a tree-structured partition. Rows 1, 3 and 5 are ν-breaks. Rows 2 and 4 are ψ-breaks.

"prototypes") should be able to live at internal nodes in the tree, and 2) as the ancestor/descendant relationships are not known *a priori*, the data should be infinitely exchangeable.

2 A Tree-Structured Stick-Breaking Process

Stick-breaking processes based on the beta distribution have played a prominent role in the development of Bayesian nonparametric methods, most significantly with the constructive approach to the Dirichlet process (DP) due to Sethuraman [10]. A random probability measure G can be drawn from a DP with base measure αH using a sequence of beta variates via:

$$G = \sum_{i=1}^{\infty} \pi_i \, \delta_{\theta_i} \qquad \pi_i = \nu_i \prod_{i'=1}^{i-1} (1 - \nu_{i'}) \qquad \theta_i \sim H \qquad \nu_i \sim \text{Be}(1, \alpha) \qquad \pi_1 = \nu_1. \tag{1}$$

We can view this as taking a stick of unit length and breaking it at a random location. We call the left side of the stick π_1 and then break the right side at a new place, calling the left side of this new break π_2. If we continue this process of "keep the left piece and break the right piece again" as in Fig. 1a, assigning each π_i a random value drawn from H, we can view this is a random probability measure centered on H. The distribution over the sequence $(\pi_1, \pi_2, \cdots)$ is a case of the GEM distribution [11], which also includes the Pitman-Yor process [12]. Note that in Eq. (1) the θ_i are i.i.d. from H; in the current paper these parameters will be drawn according to a hierarchical process.

The GEM construction provides a distribution over infinite partitions of the unit interval, with natural numbers as the index set as in Fig. 1a. In this paper, we extend this idea to create a distribution over infinite partitions that also possess a hierarchical graph topology. To do this, we will use finite-length sequences of natural numbers as our index set on the partitions. Borrowing notation from the Pólya tree (PT) construction [13], let $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \cdots, \epsilon_K)$, denote a length-$K$ sequence of positive integers, i.e., $\epsilon_k \in \mathbb{N}^+$. We denote the zero-length string as $\boldsymbol{\epsilon} = \emptyset$ and use $|\boldsymbol{\epsilon}|$ to indicate the length of $\boldsymbol{\epsilon}$'s sequence. These strings will index the nodes in the tree and $|\boldsymbol{\epsilon}|$ will then be the depth of node $\boldsymbol{\epsilon}$.

We interleave two stick-breaking procedures as in Fig. 1b. The first has beta variates $\nu_{\boldsymbol{\epsilon}} \sim \text{Be}(1, \alpha(|\boldsymbol{\epsilon}|))$ which determine the size of a given node's partition as a function of depth. The second has beta variates $\psi_{\boldsymbol{\epsilon}} \sim \text{Be}(1, \gamma)$, which determine the branching probabilities. Interleaving these processes partitions the unit interval. The size of the partition associated with each $\boldsymbol{\epsilon}$ is given by

$$\pi_{\boldsymbol{\epsilon}} = \nu_{\boldsymbol{\epsilon}} \varphi_{\boldsymbol{\epsilon}} \prod_{\boldsymbol{\epsilon}' \prec \boldsymbol{\epsilon}} \varphi_{\boldsymbol{\epsilon}'} (1 - \nu_{\boldsymbol{\epsilon}'}) \qquad \varphi_{\boldsymbol{\epsilon}\epsilon_i} = \psi_{\boldsymbol{\epsilon}\epsilon_i} \prod_{j=1}^{\epsilon_i - 1} (1 - \psi_{\boldsymbol{\epsilon} j}) \qquad \pi_\emptyset = \nu_\emptyset, \tag{2}$$

where $\boldsymbol{\epsilon}\epsilon_i$ denotes the sequence that results from appending ϵ_i onto the end of $\boldsymbol{\epsilon}$, and $\boldsymbol{\epsilon}' \prec \boldsymbol{\epsilon}$ indicates that $\boldsymbol{\epsilon}$ could be constructed by appending onto $\boldsymbol{\epsilon}'$. When viewing these strings as identifying nodes on a tree, $\{\boldsymbol{\epsilon}\epsilon_i : \epsilon_i \in 1, 2, \cdots\}$ are the children of $\boldsymbol{\epsilon}$ and $\{\boldsymbol{\epsilon}' : \boldsymbol{\epsilon}' \prec \boldsymbol{\epsilon}\}$ are the ancestors of $\boldsymbol{\epsilon}$. The $\{\pi_{\boldsymbol{\epsilon}}\}$ in Eq. (2) can be seen as products of several decisions on how to allocate mass to nodes and branches in the tree: the $\{\varphi_{\boldsymbol{\epsilon}}\}$ determine the probability of a particular sequence of children and the $\nu_{\boldsymbol{\epsilon}}$ and $(1 - \nu_{\boldsymbol{\epsilon}})$ terms determine the proportion of mass allotted to $\boldsymbol{\epsilon}$ versus nodes that are descendants of $\boldsymbol{\epsilon}$.

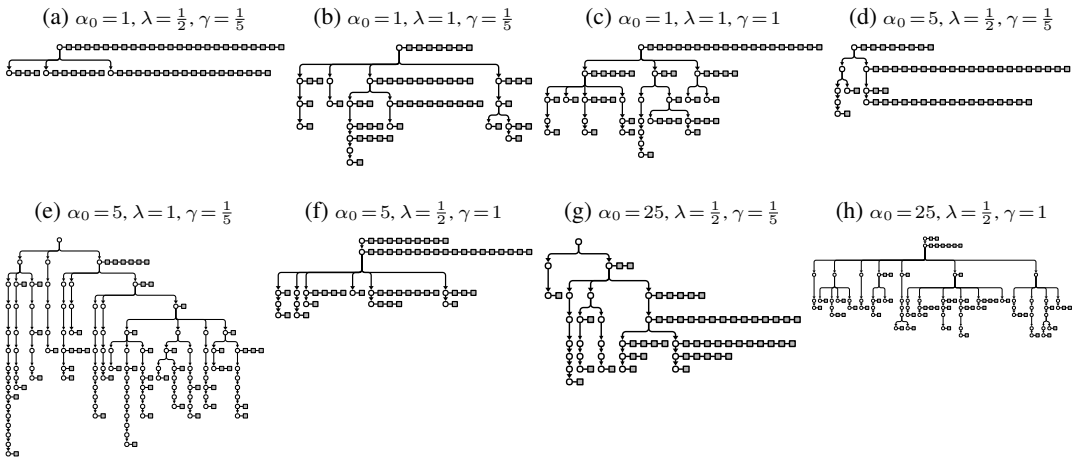


Figure 2: Eight samples of trees over partitions of fifty data, with different hyperparameter settings. The circles are represented nodes, and the squares are the data. Note that some of the sampled trees have represented nodes with no data associated with them and that the branch ordering does not correspond to a size-biased permutation.

We require that the $\{\pi_{\boldsymbol{\epsilon}}\}$ sum to one. The ψ-sticks have no effect upon this, but $\alpha(\cdot) : \mathbb{N} \to \mathbb{R}^+$ (the depth-varying parameter for the ν-sticks) must satisfy $\sum_{j=1}^{\infty} \ln(1+1/\alpha(j-1)) = +\infty$ (see [14]). This is clearly true for $\alpha(j) = \alpha_0 > 0$. A useful function that also satisfies this condition is $\alpha(j) = \lambda^j \alpha_0$ with $\alpha_0 > 0, \lambda \in (0, 1]$. The decay parameter λ allows a distribution over trees with most of the mass at an intermediate depth. This is the $\alpha(\cdot)$ we will assume throughout the remainder of the paper.

An Urn-based View

When a Bayesian nonparametric model induces partitions over data, it is sometimes possible to construct a Blackwell-MacQueen [15] type urn scheme that corresponds to sequentially generating data, while integrating out the underlying random measure. The "Chinese restaurant" metaphor for the Dirichlet process is a popular example. In our model, we can use such an urn scheme to construct a treed partition over a finite set of data.

The urn process can be seen as a path-reinforcing Bernoulli trip down the tree where each datum starts at the root and descends into children until it stops at some node. The first datum lives at the root node with probability $1/(\alpha(0)+1)$, otherwise it descends and instantiates a new child. It stays at this new child with probability $1/(\alpha(1)+1)$ or descends again and so on. A later datum stays at node $\boldsymbol{\epsilon}$ with probability $(N_{\boldsymbol{\epsilon}}+1)/(N_{\boldsymbol{\epsilon}}+N_{\boldsymbol{\epsilon}\prec\cdot}+\alpha(|\boldsymbol{\epsilon}|)+1)$, where $N_{\boldsymbol{\epsilon}}$ is the number of previous data that stopped at $\boldsymbol{\epsilon}$, and $N_{\boldsymbol{\epsilon}\prec\cdot}$ is the number of previous data that came down this path of the tree but did not stop at $\boldsymbol{\epsilon}$, i.e., a sum over all descendants: $N_{\boldsymbol{\epsilon}\prec\cdot} = \sum_{\boldsymbol{\epsilon}\prec\boldsymbol{\epsilon}'} N_{\boldsymbol{\epsilon}'}$. If a datum descends to $\boldsymbol{\epsilon}$ but does not stop then it chooses which child to descend to according to a Chinese restaurant process where the previous customers are only those data who have also descended to this point. That is, if it has reached node $\boldsymbol{\epsilon}$ but will not stay there, it descends to existing child $\boldsymbol{\epsilon}\epsilon_i$ with probability $(N_{\boldsymbol{\epsilon}\epsilon_i}+N_{\boldsymbol{\epsilon}\epsilon_i\prec\cdot})/(N_{\boldsymbol{\epsilon}\prec\cdot}+\gamma)$ and instantiates a new child with probability $\gamma/(N_{\boldsymbol{\epsilon}\prec\cdot}+\gamma)$. A particular path therefore becomes more likely according to its "popularity" with previous data. Note that a node can be a part of a popular path without having any data of its own. Fig. 2 shows the structures over fifty data drawn from this process with different hyperparameter settings. Note that the branch ordering in a realization of the urn scheme will not necessarily be the same as that of the size-biased ordering [16] of the partitions in Fig. 1b: the former is a tree over a finite set of data and the latter is over a random infinite partition.

The urn view allows us to compare this model to other priors on infinite trees. One contribution of this model is that the data can live at internal nodes in the tree, but are nevertheless infinitely exchangeable. This is in contrast to the model in [8], for example, which is not infinitely exchangeable. The nested Chinese restaurant process (nCRP) [17] provides a distribution over trees of unbounded width and depth, but data correspond to paths of infinite length, requiring an additional distribution over depths that is not path-dependent. The Pólya tree [13] uses a recursive stick-breaking process to specify a distribution over nested partitions in a binary tree, however the data live at infinitely-deep leaf nodes. The marginal distribution on the topology of a Dirichlet diffusion tree [2] (and the clustering variant of Kingman's coalescent [4]) provides path-reinforcement and infinite exchangeability, however it requires a pseudo-time hazard process and data do not live at internal nodes.

3 Hierarchical Priors for Node Parameters

One can view the stick-breaking construction of the Dirichlet process as generating an infinite partition and then labeling each cell i with parameter θ_i drawn i.i.d. from H. In a mixture model, data from the ith component are generated independently according to a distribution $f(x \mid \theta_i)$, where x takes values in a sample space $\mathcal{X}$. In our model, we continue to assume that the data are generated independently given the latent labeling, but to take advantage of the tree-structured partitioning of Section 2 an i.i.d. assumption on the node parameters is inappropriate. Rather, the distribution over the parameters at node $\boldsymbol{\epsilon}$, denoted $\theta_{\boldsymbol{\epsilon}}$, should depend in an interesting way on its ancestors $\{\theta_{\boldsymbol{\epsilon}'} : \boldsymbol{\epsilon}' \prec \boldsymbol{\epsilon}\}$. A natural way to specify such dependency is via a directed graphical model, with the requirement that edges must always point down the tree. An intuitive subclass of such graphical models are those in which a child is conditionally independent of all ancestors, given its parents and any global hyperparameters. This is the case we will focus on here, as it provides a useful view of the parameter-generation process as a "diffusion down the tree" via a Markov transition kernel that can be essentially any distribution with a location parameter. Coupling such a kernel, which we denote $T(\theta_{\boldsymbol{\epsilon}\epsilon_i} \leftarrow \theta_{\boldsymbol{\epsilon}})$, with a root-level prior $p(\theta_\emptyset)$ and the node-wise data distribution $f(x \mid \theta_{\boldsymbol{\epsilon}})$, we have a complete model for infinitely exchangeable tree-structured data on $\mathcal{X}$. We now examine a few specific examples.

Generalized Gaussian Diffusions If our data distribution $f(x \mid \theta)$ is such that the parameters can be specified as a real-valued vector $\theta \in \mathbb{R}^M$, then we can use a Gaussian distribution to describe the parent-to-child transition kernel: $T_{\mathsf{norm}}(\theta_{\boldsymbol{\epsilon}\epsilon_i} \leftarrow \theta_{\boldsymbol{\epsilon}}) = \mathcal{N}(\theta_{\boldsymbol{\epsilon}\epsilon_i} \mid \eta\, \theta_{\boldsymbol{\epsilon}}, \Lambda)$, where $\eta \in [0, 1)$. Such a kernel captures the simple idea that the child's parameters are noisy versions of the parent's, as specified by the covariance matrix Λ, while η ensures that all parameters in the tree have a finite marginal variance. While this will not result in a conjugate model unless the data are themselves Gaussian, it has the simple property that each node's parameter has a Gaussian prior that is specified by its parent. We present an application of this model in Section 5, where we model images as a distribution over binary vectors obtained by transforming a real-valued vector to $(0, 1)$ via the logistic function.

Chained Dirichlet-Multinomial Distributions If each datum is a set of counts over M discrete outcomes, as in many finite topic models, a multinomial model for $f(x \mid \theta)$ may be appropriate. In this case, $\mathcal{X} = \mathbb{N}^M$, and $\theta_{\boldsymbol{\epsilon}}$ takes values in the $(M-1)$-simplex. We can construct a parent-to-child transition kernel via a Dirichlet distribution with concentration parameter κ: $T_{\mathsf{dir}}(\theta_{\boldsymbol{\epsilon}\epsilon_i} \leftarrow \theta_{\boldsymbol{\epsilon}}) = \mathrm{Dir}(\kappa\theta_{\boldsymbol{\epsilon}})$, using a symmetric Dirichlet for the root node, i.e., $\theta_\emptyset \sim \mathrm{Dir}(\kappa \mathbf{1})$.

Hierarchical Dirichlet Processes A very general way to specify the distribution over data is to say that it is drawn from a random probability measure with a Dirichlet process prior. In our case, one flexible approach would be to model the data at node $\boldsymbol{\epsilon}$ with a distribution $G_{\boldsymbol{\epsilon}}$ as in Eq. (1). This means that $\theta_{\boldsymbol{\epsilon}} \sim G_{\boldsymbol{\epsilon}}$ where $G_{\boldsymbol{\epsilon}}$ now corresponds to an infinite set of parameters. The hierarchical Dirichlet process (HDP) [18] provides a natural parent-to-child transition kernel for the tree-structured model, again with concentration parameter κ: $T_{\mathsf{hdp}}(G_{\boldsymbol{\epsilon}\epsilon_i} \leftarrow G_{\boldsymbol{\epsilon}}) = \mathrm{DP}(\kappa G_{\boldsymbol{\epsilon}})$. At the top level, we specify a global base measure H for the root node, i.e., $G_\emptyset \sim H$. One negative aspect of this transition kernel is that the $G_{\boldsymbol{\epsilon}}$ will have a tendency to collapse down onto a single atom. One remedy is to smooth the kernel with η as in the Gaussian case, i.e., $T_{\mathsf{hdp}}(G_{\boldsymbol{\epsilon}\epsilon_i} \leftarrow G_{\boldsymbol{\epsilon}}) = \mathrm{DP}(\kappa\,(\eta\, G_{\boldsymbol{\epsilon}} + (1-\eta)\, H))$.

4 Inference via Markov chain Monte Carlo

We have so far defined a model for data that are generated from the parameters associated with the nodes of a random tree. Having seen N data and assuming a model $f(x \mid \theta_{\boldsymbol{\epsilon}})$ as in the previous section, we wish to infer possible trees and model parameters. As in most complex probabilistic models, closed form inference is impossible and we instead generate posterior samples via Markov chain Monte Carlo (MCMC). To operate efficiently over a variety of regimes without tuning, we use slice sampling [19] extensively. This allows us to sample from the true posterior distribution over the finite quantities of interest despite our model containing an infinite number of parameters. The primary data structure in our Markov chain is the set of N strings describing the current assignments of data to nodes, which we denote $\{\boldsymbol{\epsilon}_n\}_{n=1}^N$. We represent the ν-sticks and parameters $\theta_{\boldsymbol{\epsilon}}$ for all nodes that are traversed by the data in its current assignments, i.e., $\{\nu_{\boldsymbol{\epsilon}}, \theta_{\boldsymbol{\epsilon}} : \exists n, \boldsymbol{\epsilon} \preceq \boldsymbol{\epsilon}_n\}$. We also represent all ψ-sticks in the "hull" of the tree that contains the data: if at some node $\boldsymbol{\epsilon}$ one of the N data paths passes through child $\boldsymbol{\epsilon}\epsilon_i$, then we represent all the ψ-sticks in the set $\bigcup_{\boldsymbol{\epsilon}_n} \bigcup_{\boldsymbol{\epsilon}\epsilon_i \preceq \boldsymbol{\epsilon}_n} \{\psi_{\boldsymbol{\epsilon}\epsilon_j} : \epsilon_j \leq \epsilon_i\}$.

function SAMP-ASSIGNMENT(n)
  $p_{\mathsf{slice}} \sim \mathrm{Uni}(0, f(x_n \mid \theta_{\boldsymbol{\epsilon}_n}))$
  $u_{\mathsf{min}} \leftarrow 0$, $u_{\mathsf{max}} \leftarrow 1$
  loop
    $u \sim \mathrm{Uni}(u_{\mathsf{min}}, u_{\mathsf{max}})$
    $\boldsymbol{\epsilon} \leftarrow$ FIND-NODE(u, $\emptyset$)
    $p \leftarrow f(x_n \mid \theta_{\boldsymbol{\epsilon}})$
    if $p > p_{\mathsf{slice}}$ **then return** $\boldsymbol{\epsilon}$
    else if $\boldsymbol{\epsilon} < \boldsymbol{\epsilon}_n$ **then** $u_{\mathsf{min}} \leftarrow u$
    else $u_{\mathsf{max}} \leftarrow u$

function FIND-NODE(u, $\boldsymbol{\epsilon}$)
  if $u < \nu_{\boldsymbol{\epsilon}}$ **then return** $\boldsymbol{\epsilon}$
  else
    $u \leftarrow (u - \nu_{\boldsymbol{\epsilon}})/(1 - \nu_{\boldsymbol{\epsilon}})$
    while $u < 1 - \prod_j (1 - \psi_{\boldsymbol{\epsilon}\epsilon_j})$ **do**
      Draw a new ψ-stick
    $\boldsymbol{e} \leftarrow$ edges from ψ-sticks
    $i \leftarrow$ bin index for u from edges
    Draw $\theta_{\boldsymbol{\epsilon}\epsilon_i}$ and $\nu_{\boldsymbol{\epsilon}\epsilon_i}$ if necessary
    $u \leftarrow (u - e_i)/(e_{i+1} - e_i)$
    return FIND-NODE(u, $\boldsymbol{\epsilon}\epsilon_i$)

function SIZE-BIASED-PERM($\boldsymbol{\epsilon}$)
  $\rho \leftarrow \emptyset$
  while represented children **do**
    $\boldsymbol{w} \leftarrow$ weights from $\{\psi_{\boldsymbol{\epsilon}\epsilon_i}\}$
    $\boldsymbol{w} \leftarrow \boldsymbol{w} \backslash \rho$
    $j \sim \boldsymbol{w}$
    $\rho \leftarrow$ append j
  return ρ

Slice Sampling Data Assignments The primary challenge in inference with Bayesian nonparametric mixture models is often sampling from the posterior distribution over assignments, as it is frequently difficult to integrate over the infinity of unrepresented components. To avoid this difficulty, we use a slice sampling approach that can be viewed as a combination of the Dirichlet slice sampler of Walker [20] and the retrospective sampler of Papaspiliopolous and Roberts [21].

Section 2 described a path-reinforcing process for generating data from the model. An alternative method is to draw a uniform variate u on $(0, 1)$ and break sticks until we know what $\pi_{\boldsymbol{\epsilon}}$ the u fell into. One can imagine throwing a dart at the top of Fig. 1b and considering which $\pi_{\boldsymbol{\epsilon}}$ it hits. We would draw the sticks and parameters from the prior, as needed, conditioning on the state instantiated from any previous draws and with parent-to-child transitions enforcing the prior downwards in the tree. The pseudocode function FIND-NODE(u, $\boldsymbol{\epsilon}$) with $u \sim \mathrm{Uni}(0, 1)$ and $\boldsymbol{\epsilon} = \emptyset$ draws such a sample. This representation leads to a slice sampling scheme on u that does not require any tuning parameters.

To slice sample the assignment of the nth datum, currently assigned to $\boldsymbol{\epsilon}_n$, we initialize our slice sampling bounds to $(0, 1)$. We draw a new u from the bounds and use the FIND-NODE function to determine the associated $\boldsymbol{\epsilon}$ from the currently-represented state, plus any additional state that must be drawn from the prior. We do a *lexical* comparison ("string-like") of the new $\boldsymbol{\epsilon}$ and our current state $\boldsymbol{\epsilon}_n$, to determine whether this new path corresponds to a u that is "above" or "below" our current state. This lexical comparison prevents us from having to represent the initial u_n. We shrink the slice sampling bounds appropriately, depending on the comparison, until we find a u that satisfies the slice. This procedure is given in pseudocode as SAMP-ASSIGNMENT(n). After performing this procedure, we can discard any state that is not in the previously-mentioned hull of representation.

Gibbs Sampling Stick Lengths Given the represented sticks and the current assignments of nodes to data, it is straightforward to resample the lengths of the sticks from the posterior beta distributions

$$\nu_{\boldsymbol{\epsilon}} \mid \text{data} \sim \mathrm{Be}(N_{\boldsymbol{\epsilon}} + 1, N_{\boldsymbol{\epsilon}\prec\cdot} + \alpha(|\boldsymbol{\epsilon}|)) \qquad \psi_{\boldsymbol{\epsilon}\epsilon_i} \mid \text{data} \sim \mathrm{Be}(N_{\boldsymbol{\epsilon}\epsilon_i\prec\cdot} + 1, \gamma + \textstyle\sum_{j>i} N_{\boldsymbol{\epsilon}\epsilon_j\prec\cdot}),$$

where $N_{\boldsymbol{\epsilon}}$ and $N_{\boldsymbol{\epsilon}\prec\cdot}$ are the path-based counts as described in Section 2.

Gibbs Sampling the Ordering of the ψ-Sticks When using the stick-breaking representation of the Dirichlet process, it is crucial for mixing to sample over possible orderings of the sticks. In our model, we include such moves on the ψ-sticks. We iterate over each instantiated node $\boldsymbol{\epsilon}$ and perform a Gibbs update of the ordering of its immediate children using its invariance under size-biased permutation (SBP) [16]. For a given node, the ψ-sticks provide a "local" set of weights that sum to one. We repeatedly draw without replacement from the discrete distribution implied by the weights and keep the ordering that results. Pitman [16] showed that distributions over sequences such as our ψ-sticks are invariant under such permutations and we can view the SIZE-BIASED-PERM($\boldsymbol{\epsilon}$) procedure as a Metropolis–Hastings proposal with an acceptance ratio that is always one.

Slice Sampling Stick-Breaking Hyperparameters Given all of the instantiated sticks, we slice sample from the conditional posterior distribution over the hyperparameters α_0, λ and γ:

$$p(\alpha_0, \lambda \mid \{\nu_{\boldsymbol{\epsilon}}\}) \propto \mathbb{I}(\alpha_0^{\mathsf{min}} < \alpha_0 < \alpha_0^{\mathsf{max}}) \mathbb{I}(\lambda^{\mathsf{min}} < \lambda < \lambda^{\mathsf{max}}) \prod_{\boldsymbol{\epsilon}} \mathrm{Be}(\nu_{\boldsymbol{\epsilon}} \mid 1, \lambda^{|\boldsymbol{\epsilon}|}\alpha_0)$$

$$p(\gamma \mid \{\psi_{\boldsymbol{\epsilon}}\}) \propto \mathbb{I}(\gamma^{\mathsf{min}} < \gamma < \gamma^{\mathsf{max}}) \prod_{\boldsymbol{\epsilon}} \mathrm{Be}(\psi_{\boldsymbol{\epsilon}} \mid 1, \gamma),$$

where the products are over nodes in the aforementioned hull. We initialize the bounds of the slice sampler with the bounds of the top-hat prior.

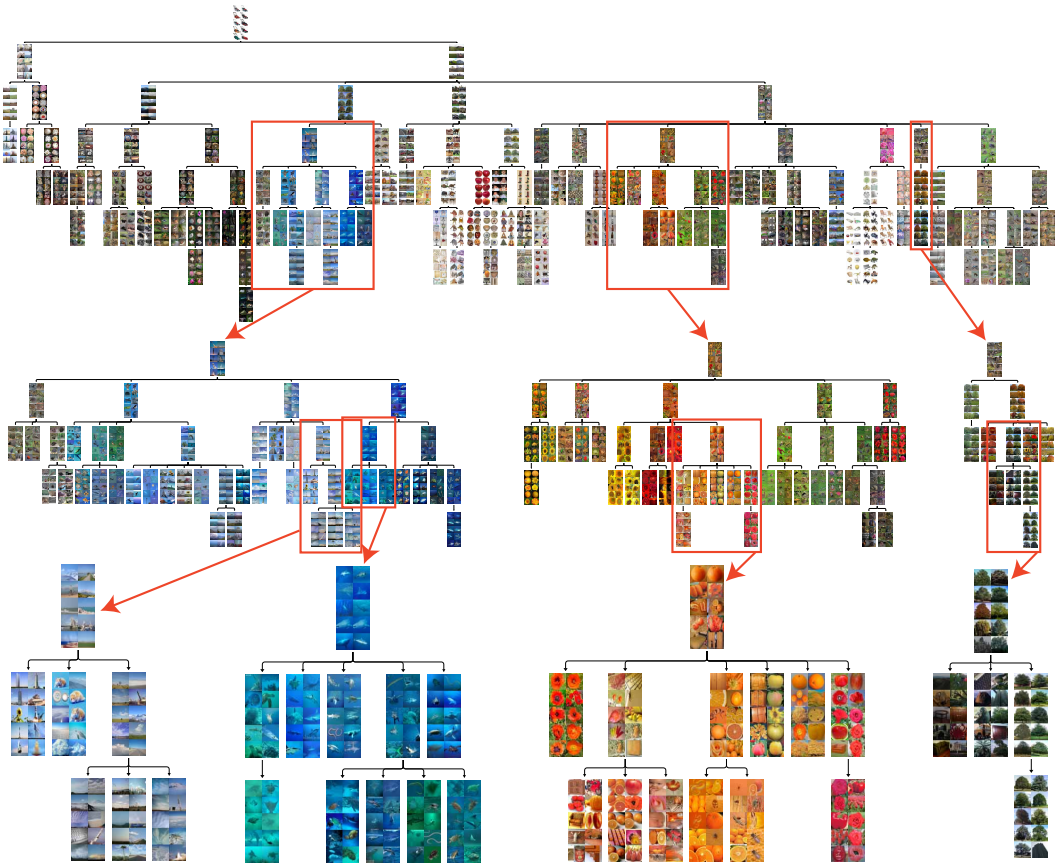


Figure 3: These figures show a subset of the tree learned from the 50,000 CIFAR-100 images. The top tree only shows nodes for which there were at least 250 images. The ten shown at each node are those with the highest probability under the node's distribution. The second row shows three expanded views of subtrees, with nodes that have at least 50 images. Detailed views of portions of these subtrees are shown in the third row.

Selecting a Single Tree We have so far described a procedure for generating posterior samples from the tree structures and associated stick-breaking processes. If our objective is to find a single tree, however, samples from the posterior distribution are unsatisfying. Following [17], we report a best single tree structure over the data by choosing the sample from our Markov chain that has the highest complete-data likelihood $p(\{x_n, \boldsymbol{\epsilon}_n\}_{n=1}^N \mid \{\nu_{\boldsymbol{\epsilon}}\}, \{\psi_{\boldsymbol{\epsilon}}\}, \alpha_0, \lambda, \gamma)$.

5 Hierarchical Clustering of Images

We applied our model and MCMC inference to the problem of hierarchically clustering the CIFAR-100 image data set [1]. These data are a labeled subset of the *80 million tiny images* data [22] with 50,000 32×32 color images. We did not use the labels in our clustering. We modeled the images via 256-dimensional binary features that had been previously extracted from each image (i.e., $x_n \in \{0,1\}^{256}$) using a deep neural network that had been trained for an image retrieval task [23]. We used a factored Bernoulli likelihood at each node, parameterized by a latent 256-dimensional real vector (i.e., $\theta_{\boldsymbol{\epsilon}} \in \mathbb{R}^{256}$) that was transformed component-wise via the logistic function:

$$f(x_n \mid \theta_{\boldsymbol{\epsilon}}) = \prod_{d=1}^{256} \left(1 + \exp\{-\theta_{\boldsymbol{\epsilon}}^{(d)}\}\right)^{-x_n^{(d)}} \left(1 + \exp\{\theta_{\boldsymbol{\epsilon}}^{(d)}\}\right)^{1-x_n^{(d)}}.$$

The prior over the parameters of a child node was Gaussian with its parent's value as the mean. The covariance of the prior (Λ in Section 3) was diagonal and inferred as part of the Markov chain. We placed independent $\mathrm{Uni}(0.01, 1)$ priors on the elements of the diagonal. To efficiently learn the node parameters, we used Hamiltonian (hybrid) Monte Carlo (HMC) [24], taking 25 leapfrog HMC steps, with a randomized step size. We occasionally interleaved a slice sampling move for robustness.

[1] `http://www.cs.utoronto.ca/~kriz/cifar.html`

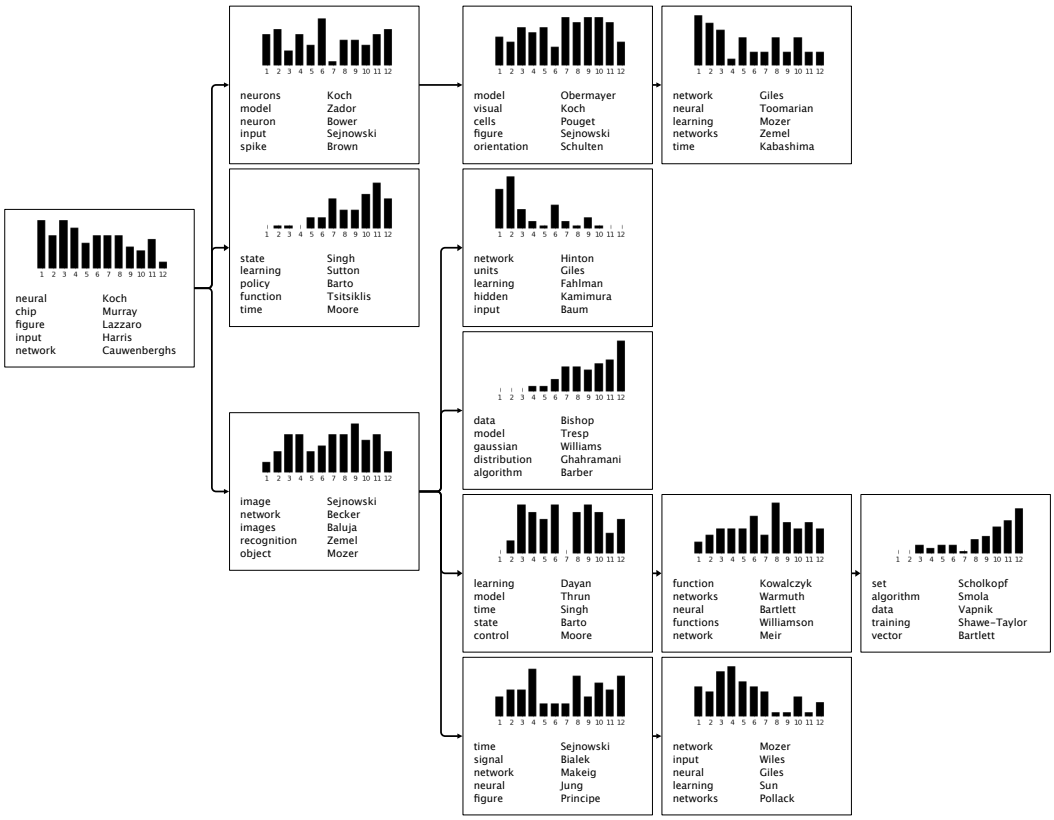


Figure 4: A subtree of documents from NIPS 1-12, inferred using 20 topics. Only nodes with at least 50 documents are shown. Each node shows three aggregated statistics at that node: the five most common author names, the five most common words and a histogram over the years of proceedings.

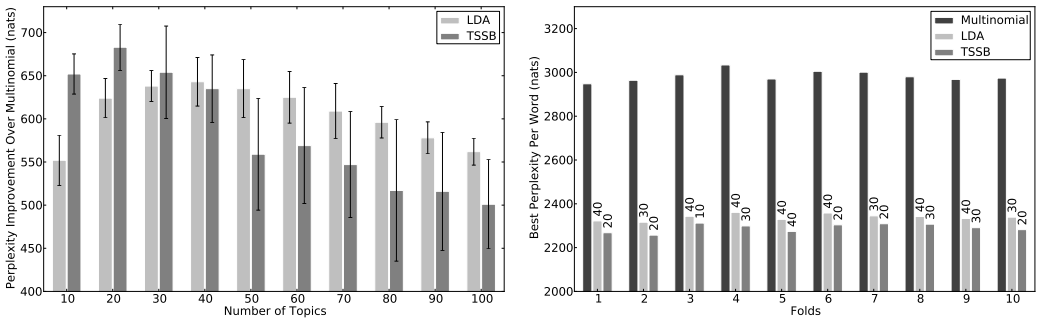
For the stick-breaking processes, we used $\alpha_0 \sim \text{Uni}(10, 50)$, $\lambda \sim \text{Uni}(0.05, 0.8)$, and $\gamma \sim \text{Uni}(1, 10)$. Using Python on a single core of a modern workstation each MCMC iteration of the entire model (including slice sampled reassignment of all 50,000 images) requires approximately three minutes. Fig. 3 represents a part of the tree with the best complete-data log likelihood after 4000 such iterations. The tree provides a useful visualization of the data set, capturing broad variations in color at the higher levels of the tree, with lower branches varying in texture and shape. A larger version of this tree is provided in the supplementary material.

6 Hierarchical Modeling of Document Topics

We also used our approach in a bag-of-words topic model, applying it to 1740 papers from NIPS 1–12 [2]. As in latent Dirichlet allocation (LDA) [25], we consider a topic to be a distribution over words and each document to be described by a distribution over topics. In LDA, each document has a unique topic distribution. In our model, however, each document lives at a node and that *node* has a unique topic distribution. Thus multiple documents share a distribution over topics if they inhabit the same node. Each node's topic distribution is from a chained Dirichlet-multinomial as described in Section 3. The topics each have symmetric Dirichlet priors over their word distributions. This results in a different kind of topic model than that provided by the nested Chinese restaurant process. In the nCRP, each node corresponds to a topic and documents are spread across infinitely-long paths down the tree. Each word is drawn from a distribution over depths that is given by a GEM distribution. In the nCRP, it is not the *documents* that have the hierarchy, but the *topics*.

We did two kinds of analyses. The first is a visualization as with the image data of the previous section, using all 1740 documents. The subtree in Fig. 4 shows the nodes that had at least fifty documents, along with the most common authors and words at that node. The normalized histogram in each box shows which of the twelve years are represented among the documents in that node. An

[2] `http://cs.nyu.edu/~roweis/data.html`



(a) Improvement versus multinomial, by number of topics

(b) Best perplexity per word, by folds

Figure 5: Results of predictive performance comparison between latent Dirichlet allocation (LDA) and tree-structured stick breaking (TSSB). a) Mean improvement in perplexity per word over Laplace-smoothed multinomial, as a function of topics (larger is better). The error bars show the standard deviation of the improvement across the ten folds. b) Best predictive perplexity per word for each fold (smaller is better). The numbers above the LDA and TSSB bars show how many topics were used to achieve this.

expanded version of this tree is provided in the supplementary material. Secondly, we quantitatively assessed the predictive performance of the model. We created ten random partitions of the NIPS corpus into 1200 training and 540 test documents. We then performed inference with different numbers of topics $(10, 20, \ldots, 100)$ and evaluated the predictive perplexity of the held-out data using an empirical likelihood estimate taken from a mixture of multinomials (pseudo-documents of infinite length, see, e.g. [26]) with 100,000 components. As Fig. 5a shows, our model improves in performance over standard LDA for smaller numbers of topics. This improvement appears to be due to the constraints on possible topic distributions that are imposed by the diffusion. For larger numbers of topics, however, it may be that these constraints become a hindrance and the model may be allocating predictive mass to regions where it is not warranted. In absolute terms, more topics did not appear to improve predictive performance for LDA or the tree-structured model. Both models performed best with fewer than fifty topics and the best tree model outperformed the best LDA model on all folds, as shown in Fig. 5b.

The MCMC inference procedure we used to train our model was as follows: first, we ran Gibbs sampling of a standard LDA topic model for 1000 iterations. We then burned in the tree inference for 500 iterations with fixed word-topic associations. We then allowed the word-topic associations to vary and burned in for an additional 500 iterations, before drawing 5000 samples from the full posterior. For the comparison, we burned in LDA for 1000 iterations and then drew 5000 samples from the posterior [27]. For both models we thinned the samples by a factor of 50. The mixing of the topic model seems to be somewhat sensitive to the initialization of the κ parameter in the chained Dirichlet-multinomial and we initialized this parameter to be the same as the number of topics.

7 Discussion

We have presented a model for a distribution over random measures that also constructs a hierarchy, with the goal of constructing a general-purpose prior on tree-structured data. Our approach is novel in that it combines infinite exchangeability with a representation that allows data to live at internal nodes on the tree, without a hazard rate process. We have developed a practical inference approach based on Markov chain Monte Carlo and demonstrated it on two real-world data sets in different domains.

The imposition of structure on the parameters of an infinite mixture model is an increasingly important topic. In this light, our notion of evolutionary diffusion down a tree sits within the larger class of models that construct dependencies between distributions on random measures [28, 29, 18].

Acknowledgements

The authors wish to thank Alex Krizhevsky for providing the image feature data. We also thank Kurt Miller, Iain Murray, Hanna Wallach, and Sinead Williamson for valuable discussions, and Yee Whye Teh for suggesting Gibbs moves based on size-biased permutation. RPA is a Junior Fellow of the Canadian Institute for Advanced Reserch.

References

[1] Christopher K. I. Williams. A MCMC approach to hierarchical mixture modelling. In *Advances in Neural Information Processing Systems 12*, pages 680–686. 2000.

[2] Radford M. Neal. Density modeling and clustering using Dirichlet diffusion trees. In *Bayesian Statistics 7*, pages 619–629, 2003.

[3] Katherine A. Heller and Zoubin Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

[4] Yee Whye Teh, Hal Daumé III, and Daniel Roy. Bayesian agglomerative clustering with coalescents. In *Advances in Neural Information Processing Systems 20*, 2007.

[5] Charles Kemp, Thomas L. Griffiths, Sean Stromsten, and Joshua B. Tenenbaum. Semi-supervised learning with trees. In *Advances in Neural Information Processing Systems 16*. 2004.

[6] Daniel M. Roy, Charles Kemp, Vikash K. Mansinghka, and Joshua B. Tenenbaum. Learning annotated hierarchies from relational data. In *Advances in Neural Information Processing Systems 19*, 2007.

[7] Hal Daumé III. Bayesian multitask learning with latent hierarchies. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009.

[8] Edward Meeds, David A. Ross, Richard S. Zemel, and Sam T. Roweis. Learning stick-figure models using nonparametric Bayesian priors over trees. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[9] Evgeniy Bart, Ian Porteous, Pietro Perona, and Max Welling. Unsupervised learning of visual taxonomies. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[10] Jayaram Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[11] Jim Pitman. Poisson–Dirichlet and GEM invariant distributions for split-and-merge transformation of an interval partition. *Combinatorics, Probability and Computing*, 11:501–514, 2002.

[12] Jim Pitman and Marc Yor. The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, 1997.

[13] R. Daniel Mauldin, William D. Sudderth, and S. C. Williams. Pólya trees and random distributions. *The Annals of Statistics*, 20(3):1203–1221, September 1992.

[14] Hemant Ishwaran and Lancelot F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, March 2001.

[15] David Blackwell and James B. MacQueen. Ferguson distributions via Pólya urn schemes. *Annals of Statistics*, 1(2):353–355, 1973.

[16] Jim Pitman. Random discrete distributions invariant under size-biased permutation. *Advances in Applied Probability*, 28(2):525–539, 1996.

[17] 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*, 57(2):1–30, 2010.

[18] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[19] Radford M. Neal. Slice sampling (with discussion). *The Annals of Statistics*, 31(3):705–767, 2003.

[20] Stephen G. Walker. Sampling the Dirichlet mixture model with slices. *Communications in Statistics*, 36:45–54, 2007.

[21] Omiros Papaspiliopoulos and Gareth O. Roberts. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, 95(1):169–186, 2008.

[22] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009.

[24] Radford M. Neal. MCMC using Hamiltonian dynamics. In *Handbook of Markov chain Monte Carlo*. Chapman and Hall / CRC Press.

[25] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

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

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

[28] Steven N. MacEachern. Dependent nonparametric processes. In *Proceedings of the Section on Bayesian Statistical Science*, 1999.

[29] Steven N. MacEachern, Athanasios Kottas, and Alan E. Gelfand. Spatial nonparametric Bayesian models. Technical Report 01-10, Institute of Statistics and Decision Sciences, Duke University, 2001.