Binary to Bushy: Bayesian Hierarchical Clustering with the Beta Coalescent

Yuening Hu[1], **Jordan Boyd-Graber**[2], **Hal Daumè III**[3], **Z. Irene Ying**[4]
1, 3: Computer Science, 2: iSchool and UMIACS, 4: Agricultural Research Service
1, 2, 3: University of Maryland, 4: Department of Agriculture
ynhu@cs.umd.edu, {jbg,hal}@umiacs.umd.edu, zhu.ying@ars.usda.gov

Abstract

Discovering hierarchical regularities in data is a key problem in interacting with large datasets, modeling cognition, and encoding knowledge. A previous Bayesian solution—Kingman's coalescent—provides a probabilistic model for data represented as a *binary* tree. Unfortunately, this is inappropriate for data better described by bushier trees. We generalize an existing belief propagation framework of Kingman's coalescent to the *beta coalescent*, which models a wider range of tree structures. Because of the complex combinatorial search over possible structures, we develop new sampling schemes using sequential Monte Carlo and Dirichlet process mixture models, which render inference efficient and tractable. We present results on synthetic and real data that show the beta coalescent outperforms Kingman's coalescent and is qualitatively better at capturing data in bushy hierarchies.

1 The Need For Bushy Hierarchical Clustering

Hierarchical clustering is a fundamental data analysis problem: given observations, what hierarchical grouping of those observations effectively encodes the similarities between observations? This is a critical task for understanding and describing observations in many domains [1, 2], including natural language processing [3], computer vision [4], and network analysis [5]. In all of these cases, natural and intuitive hierarchies are not binary but are instead **bushy**, with more than two children per parent node. Our goal is to provide efficient algorithms to discover bushy hierarchies.

We review existing nonparametric probabilistic clustering algorithms in Section 2, with particular focus on Kingman's coalescent [6] and its generalization, the beta coalescent [7, 8]. While Kingman's coalescent has attractive properties—it is probabilistic and has edge "lengths" that encode how similar clusters are—it only produces binary trees. The beta coalescent (Section 3) does not have this restriction. However, naïve inference is impractical, because bushy trees are more complex: we need to consider all possible subsets of nodes to construct each internal nodes in the hierarchy.

Our first contribution is a *generalization of the belief propagation framework* [9] for beta coalescent to compute the joint probability of observations and trees (Section 3). After describing sequential Monte Carlo posterior inference for the beta coalescent, we develop efficient inference strategies in Section 4, where we use proposal distributions that draw on the connection between Dirichlet processes—a ubiquitous Bayesian nonparametric tool for non-hierarchical clustering—and hierarchical coalescents to *make inference tractable*. We present results on both synthetic and real data that show the beta coalescent captures bushy hierarchies and outperforms Kingman's coalescent (Section 5).

2 Bayesian Clustering Approaches

Recent hierarchical clustering techniques have been incorporated inside statistical models; this requires formulating clustering as a statistical—often Bayesian—problem. Heller et al. [10] build

binary trees based on the marginal likelihoods, extended by Blundell et al. [11] to trees with arbitrary branching structure. Ryan et al. [12] propose a tree-structured stick-breaking process to generate trees with unbounded width and depth, which supports data observations at leaves *and* internal nodes.[1] However, these models do not distinguish edge lengths, an important property in distinguishing how "tight" the clustering is at particular nodes.

Hierarchical models can be divided into complementary "fragmentation" and "coagulation" frameworks [7]. Both produce hierarchical partitions of a dataset. Fragmentation models start with a single partition and divide it into ever more specific partitions until only singleton partitions remain. Coagulation frameworks repeatedly merge singleton partitions until only one partition remains. Pitman-Yor diffusion trees [13], a generalization of Dirichlet diffusion trees [14], are an example of a bushy fragmentation model, and they model edge lengths and build non-binary trees.

Instead, our focus is on bottom-up coalescent models [8], one of the coagulation models and complementary to diffusion trees, which can also discover hierarchies and edge lengths. In this model, n nodes are observed (we use both *observed* to emphasize that nodes are known and *leaves* to emphasize topology). These observed nodes are generated through some unknown tree with latent edges and unobserved internal nodes. Each node (both observed and latent) has a single parent. The convention in such models is to assume our observed nodes come at time $t = 0$, and at time $-\infty$ all nodes share a common ur-parent through some sequence of intermediate parents.

Consider a set of n individuals observed at the present (time $t = 0$). All individuals start in one of n singleton sets. After time t_i, a set of these nodes coalesce into a new node. Once a set merges, their parent replaces the original nodes. This is called a **coalescent event**. This process repeats until there is only one node left, and a complete tree structure π (Figure 1) is obtained.

Different coalescents are defined by different probabilities of merging a set of nodes. This is called the coalescent **rate**, defined by a general family of coalescents: the lambda coalescent [7, 15]. We represent the rate via the symbol λ_n^k, the rate at which k out of n nodes merge into a parent node. From a collection of n nodes, $k \leq n$ can coalesce at some coalescent event (k can be different for different coalescent events). The rate of a fraction γ of the nodes coalescing is given by $\gamma^{-2}\Lambda(d\gamma)$, where $\Lambda(d\gamma)$ is a finite measure on $[0, 1]$. So k nodes merge at rate

$$\lambda_n^k = \int_0^1 \gamma^{k-2}(1-\gamma)^{n-k}\Lambda(d\gamma) \qquad (2 \leq k \leq n). \tag{1}$$

Choosing different measures yields different coalescents. A degenerate Dirac delta measure at 0 results in Kingman's coalescent [6], where λ_n^k is 1 when $k = 2$ and zero otherwise. Because this gives zero probability to non-binary coalescent events, this only creates binary trees.

Alternatively, using a beta distribution $\text{BETA}(2-\alpha, \alpha)$ as the measure Λ yields the beta coalescent. When α is closer to 1, the tree is bushier; as α approaches 2, it becomes Kingman's coalescent. If we have n_{i-1} nodes at time t_{i-1} in a beta coalescent, the rate $\lambda_{n_{i-1}}^{k_i}$ for a children set of k_i nodes at time t_i and the total rate $\lambda_{n_{i-1}}$ of *any* children set merging—summing over all possible mergers—is

$$\lambda_{n_{i-1}}^{k_i} = \frac{\Gamma(k_i - \alpha)\Gamma(n_{i-1} - k_i + \alpha)}{\Gamma(2-\alpha)\Gamma(\alpha)\Gamma(n_{i-1})} \quad \text{and} \quad \lambda_{n_{i-1}} = \sum_{k_i=2}^{n_{i-1}} \tbinom{n_{i-1}}{k_i} \lambda_{n_{i-1}}^{k_i}. \tag{2}$$

Each coalescent event also has an edge length—**duration**—δ_i. The duration of an event comes from an exponential distribution, $\delta_i \sim \exp(\lambda_{n_{i-1}})$, and the parent node forms at time $t_i = t_{i-1} - \delta_i$. Shorter durations mean that the children more closely resemble their parent (the mathematical basis for similarity is specified by a transition kernel, Section 3).

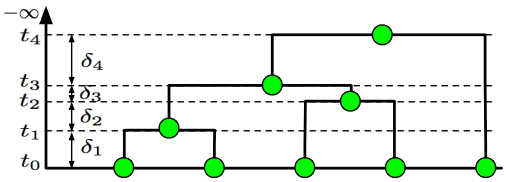
Analogous to Kingman's coalescent, the prior probability of a complete tree π is the product of all of its constituent coalescent events $i = 1, \ldots m$, merging k_i children after duration δ_i,

$$p(\pi) = \prod_{i=1}^{m} \underbrace{p(k_i|n_{i-1})}_{\text{Merge } k_i \text{ nodes}} \cdot \underbrace{p(\delta_i|k_i, n_{i-1})}_{\text{After duration } \delta_i} = \prod_{i=1}^{m} \lambda_{n_{i-1}}^{k_i} \cdot \exp(-\lambda_{n_{i-1}}\delta_i). \tag{3}$$

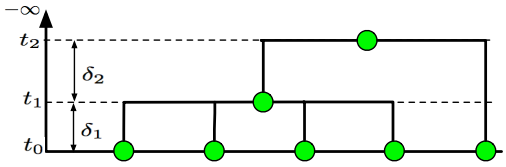
[1] This is appropriate where the entirety of a population is known—both ancestors and descendants. We focus on the case where only the descendants are known. For a concrete example, see Section 5.2.

Algorithm 1 MCMC inference for generating a tree

for Particle $s = 1, 2, \cdots, S$ **do**
  Initialize $n^s = n$, $i = 0$, $t_0^s = 0$, $w_0^s = 1$.
  Initialize the node set $V^s = \{\rho_0, \rho_1, \cdots, \rho_n\}$.
while $\exists s \in \{1 \cdots S\}$ where $n^s > 1$ **do**
  Update $i = i + 1$.
  for Particle $s = 1, 2, \cdots, S$ **do**
    if $n^s == 1$ **then**
      Continue.
    Propose a duration δ_i^s by Equation 10.
    Set coalescent time $t_i^s = t_{i-1}^s - \delta_i^s$.
    Sample partitions p_i^s from DPMM.
    Propose a set $\rho_{\vec{c}_i}^s$ according to Equation 11.
    Update weight w_i^s by Equation 13.
    Update $n^s = n^s - |\rho_{\vec{c}_i}^s| + 1$.
    Remove $\rho_{\vec{c}_i}^s$ from V^s, add ρ_i^s to V^s.
  Compute effective sample size **ESS** [16].
  if ESS $< S/2$ **then**
    Resample particles [17].



(a) Kingman's coalescent



(b) the beta coalescent

Figure 1: The beta coalescent can merge four similar nodes at once, while Kingman's coalescent only merges two each time.

3 Beta Coalescent Belief Propagation

The beta coalescent prior only depends on the topology of the tree. In real clustering applications, we also care about a node's *children* and *features*. In this section, we define the nodes and their features, and then review how we use message passing to compute the probabilities of trees.

An internal node ρ_i is defined as the merger of other nodes. The children **set** of node ρ_i, $\rho_{\vec{c}_i}$, coalesces into a new node $\rho_i \equiv \cup_{b \in \vec{c}_i} \rho_b$. This encodes the identity of the nodes that participate in specific coalescent events; Equation 3, in contrast, only considers the number of nodes involved in an event. In addition, each node is associated with a multidimensional feature vector y_i.

Two terms specify the relationship between nodes' features: an **initial** distribution $p_0(y_i)$ and a **transition kernel** $\kappa_{t_i t_b}(y_i, y_b)$. The initial distribution can be viewed as a prior or regularizer for feature representations. The transition kernel encourages a child's feature y_b (at time t_b) to resemble feature y_i (formed at t_i); shorter durations $t_b - t_i$ increase the resemblance.

Intuitively, the transition kernel can be thought as a similarity score; the more similar the features are, the more likely nodes are. For Brownian diffusion (discussed in Section 4.3), the transition kernel follows a Gaussian distribution centered at a feature. The covariance matrix Σ is decided by the mutation rate μ [18, 9], the probability of a mutation in an individual. Different kernels (e.g., multinomial, tree kernels) can be applied depending on modeling assumptions of the feature representations.

To compute the probability of the beta coalescent tree π and observed data $\mathbf{x}$, we generalize the belief propagation framework used by Teh et al. [9] for Kingman's coalescent; this is a more scalable alternative to other approaches for computing the probability of a Beta coalescent tree [19]. We define a subtree structure $\theta_i = \{\theta_{i-1}, \delta_i, \rho_{\vec{c}_i}\}$, thus the tree θ_m after the final coalescent event m is a complete tree π. The message for node ρ_i marginalizes over the features of the nodes in its children set.[2] The total message for a parent node ρ_i is

$$M_{\rho_i}(y_i) = Z_{\rho_i}^{-1}(\mathbf{x}|\theta_i) \prod_{b \in \vec{c}_i} \int \kappa_{t_i t_b}(y_i, y_b) M_{\rho_b}(y_b) dy_b. \tag{4}$$

where $Z_{\rho_i}(\mathbf{x}|\theta_i)$ is the local normalizer, which can be computed as the combination of initial distribution and messages from a set of children,

$$Z_{\rho_i}(\mathbf{x}|\theta_i) = \int p_0(y_i) \prod_{b \in \vec{c}_i} \left(\int \kappa_{t_i t_b}(y_i, y_b) M_{\rho_b}(y_b) dy_b \right) dy_i. \tag{5}$$

[2]When ρ_b is a leaf, the message $M_{\rho_b}(y_b)$ is a delta function centered on the observation.

Recursively performing this marginalization through message passing provides the joint probability of a complete tree π and the observations $\mathbf{x}$. At the root,

$$Z_{-\infty}(\mathbf{x}|\theta_m) = \int p_0(y_{-\infty})\kappa_{-\infty,t_m}(y_{-\infty}, y_m)M_{\rho_m}(y_m)dy_m dy_{-\infty} \tag{6}$$

where $p_0(y_{-\infty})$ is the initial feature distribution and m is the number of coalescent events. This gives the marginal probability of the whole tree,

$$p(\mathbf{x}|\pi) = Z_{-\infty}(\mathbf{x}|\theta_m)\prod_{i=1}^{m} Z_{\rho_i}(\mathbf{x}|\theta_i), \tag{7}$$

The joint probability of a tree π combines the prior (Equation 3) and likelihood (Equation 7),

$$p(\mathbf{x}, \pi) = Z_{-\infty}(\mathbf{x}|\theta_m)\prod_{i=1}^{m} \lambda_{n_{i-1}}^{k_i} \exp(-\lambda_{n_{i-1}}\delta_i) \cdot Z_{\rho_i}(\mathbf{x}|\theta_i). \tag{8}$$

3.1 Sequential Monte Carlo Inference

Sequential Monte Carlo (SMC)—often called particle filters—estimates a structured sequence of hidden variables based on observations [20]. For coalescent models, this estimates the posterior distribution over tree structures given observations $\mathbf{x}$. Initially ($i = 0$) each observation is in a singleton cluster;[3] in subsequent particles ($i > 0$), points coalesce into more complicated tree structures θ_i^s, where s is the particle index and we add superscript s to all the related notations to distinguish between particles. We use sequential importance resampling [21, SIR] to weight each particle s at time t_i, denoted as w_i^s.

The weights from SIR approximate the posterior. Computing the weights requires a conditional distribution of data given a latent state $p(\mathbf{x}|\theta_i^s)$, a transition distribution between latent states $p(\theta_i^s|\theta_{i-1}^s)$, and a proposal distribution $f(\theta_i^s|\theta_{i-1}^s, \mathbf{x})$. Together, these distributions define weights

$$w_i^s = w_{i-1}^s \frac{p(\mathbf{x} \mid \theta_i^s)p(\theta_i^s \mid \theta_{i-1}^s)}{f(\theta_i^s \mid \theta_{i-1}^s, \mathbf{x})}. \tag{9}$$

Then we can approximate the posterior distribution of the hidden structure using the normalized weights, which become more accurate with more particles.

To apply SIR inference to belief propagation with the beta coalescent prior, we first define the particle space structure. The s^{th} particle represents a subtree θ_{i-1}^s at time t_{i-1}^s, and a transition to a new subtree θ_i^s takes a set of nodes $\rho_{\vec{c}_i}^s$ from θ_{i-1}^s, and merges them at t_i^s, where $t_i^s = t_{i-1}^s - \delta_i^s$ and $\theta_i^s = \{\theta_{i-1}^s, \delta_i^s, \rho_{\vec{c}_i}^s\}$. Our proposal distribution must provide the duration δ_i^s and the children set $\rho_{\vec{c}_i}^s$ to merge based on the previous subtree θ_{i-1}^s.

We propose the duration δ_i^s from the prior exponential distribution and propose a children set from the posterior distribution based on the local normalizers. [4] This is the "priorpost" method in Teh et al. [9].

However, this approach is intractable. Given n_{i-1} nodes at time t_i, we must consider all possible children sets $\binom{n_{i-1}}{2} + \binom{n_{i-1}}{3} + \cdots + \binom{n_{i-1}}{n_{i-1}}$. The computational complexity grows from $O(n_{i-1}^2)$ (Kingman's coalescent) to $O(2^{n_{i-1}})$ (beta coalescent).

4 Efficiently Finding Children Sets with DPMM

We need a more efficient way to consider possible children sets. Even for Kingman's coalescent, which only considers *pairs* of nodes, Gorur et al. [22] do not exhaustively consider all pairs. Instead, they use data structures from computational geometry to select the R closest pairs as their restriction set, reducing inference to $O(n \log n)$. While finding closest pairs is a traditional problem in computational geometry, discovering arbitrary-sized sets is less studied.

[3] The relationship between time and particles is non-intuitive. Time t goes backward with subsequent particles. When we use time-specific adjectives for particles, this is with respect to *inference*.

[4] This is a special case of Section 4.2's algorithm, where the restriction set Ω_i is all possible subsets.

In this section, we describe how we use a Dirichlet process mixture model [23, DPMM] to discover a restriction set Ω, integrating DPMMs into the SMC proposal. We first briefly review what DPMMs are, describe why they are attractive, and then describe how we incorporate DPMMs in SMC inference.

The DPMM is defined by a concentration β and a base distribution G_0. A distribution over mixtures is drawn from a Dirichlet process (DP): $G \sim \text{DP}(\beta, G_0)$. Each observation x_i is assigned to a mixture component μ_i drawn from G. Because the Dirichlet process is a discrete distribution, observations i and j can have the same mixture component ($\mu_i = \mu_j$). When this happens, points are said to be in the same partition. Posterior inference can discover a distribution over partitions. A full derivation of these sampling equations appears in the supplemental material.

4.1 Attractive Properties of DPMMs

DPMMs and Coalescents Berestycki et al. [8] showed that the distribution over partitions in a Dirichlet process is equivalent to the distribution over coalescents' allelic partitions—the set of members that have the same feature representation—when the mutation rate μ of the associated kernel is half of the Dirichlet concentration β (Section 3). For Brownian diffusion, we can connect DPMM with coalescents by setting the kernel covariance $\Sigma = \mu\mathbf{I}$ to $\Sigma = \beta/2\mathbf{I}$.

The base distribution G_0 is also related with nodes' feature. The base distribution G_0 of a Dirichlet process generates the probability measure G for each block, which generates the nodes in a block. As a result, we can select a base distribution which fits the distribution of the samples in coalescent process. For example, if we use Gaussian distribution for the transition kernel and prior, a Gaussian is also appropriate as the DPMM base distribution.

Effectiveness as a Proposal The necessary condition for a valid proposal [24] is that it should have support on a superset of the true posterior. In our case, the distribution over partitions provided by the DPMM considers all possible children sets that could be merged in the coalescent. Thus the new proposal with DPMM satisfies this requirement, and it is a valid proposal.

In addition, Chen [25] gives a set of desirable criteria for a good proposal distribution: accounts for outliers, considers the likelihood, and lies close to the true posterior. The DPMM fulfills these criteria. First, the DPMM provides a distribution over all partitions. Varying the concentration parameter β can control the length of the tail of the distribution over partitions. Second, choosing the base distribution of the DPMM appropriately models the feature likelihood; i.e., ensuring the DPMM places similar nodes together in a partition with high probability. Third, the DPMM qualitatively provides reasonable children sets when compared with exhaustively considering all children sets (Figure 2(c)).

4.2 Incorporating DPMM in SMC Proposals

To address the inference intractability in Section 3.1, we use the DPMM to obtain a distribution over partitions of nodes. Each partition contains clusters of nodes, and we take a union over all partitions to create a restriction set $\Omega_i = \{\omega_{i1}, \omega_{i2}, \cdots\}$, where each ω_{ij} is a subset of the n_{i-1} nodes. A standard Gibbs sampler provides these partitions (see supplemental).

With this restriction set Ω_i, we propose the duration time δ_i^s from the exponential distribution and propose a children set $\rho_{\vec{c}_i}^s$ based on the local normalizers

$$f_i(\delta_i^s) = \lambda_{n_{i-1}}^s \exp(-\lambda_{n_{i-1}}^s \delta_i^s) \quad (10) \qquad f_i(\rho_{\vec{c}_i}^s | \delta_i^s, \theta_{i-1}^s) = \frac{Z_{\rho_i}(\mathbf{x}|\theta_{i-1}^s, \delta_i^s, \rho_{\vec{c}_i}^s)}{Z_0} \cdot \mathbb{I}\left[\rho_{\vec{c}_i}^s \in \Omega_i^s\right], \quad (11)$$

where Ω_i^s restricts the candidate children sets, $\mathbb{I}$ is the indicator, and we replace $Z_{\rho_i}(\mathbf{x}|\theta_i^s)$ with $Z_{\rho_i}(\mathbf{x}|\theta_{i-1}^s, \delta_i^s, \rho_{\vec{c}_i}^s)$ since they are equivalent here. The normalizer is

$$Z_0 = \sum\nolimits_{\rho'_{\vec{c}}} Z_{\rho_i}(\mathbf{x}|\theta_{i-1}^s, \delta_i^s, \rho'_{\vec{c}}) \cdot \mathbb{I}\left[\rho'_{\vec{c}} \in \Omega_i^s\right] = \sum\nolimits_{\rho'_{\vec{c}} \in \Omega_i^s} Z_{\rho_i}(\mathbf{x}|\theta_{i-1}^s, \delta_i^s, \rho'_{\vec{c}}). \quad (12)$$

Applying the true distribution (the i^{th} multiplicand from Equation 8) and the proposal distribution (Equation 10 and Equation 11) to the SIR weight update (Equation 9),

$$w_i^s = w_{i-1}^s \frac{\lambda_{n_{i-1}}^{s\,|\rho_{\vec{c}_i}^s|} \cdot \sum_{\rho'_{\vec{c}} \in \Omega_i^s} Z_{\rho_i}(\mathbf{x}|\theta_{i-1}^s, \delta_i^s, \rho'_{\vec{c}})}{\lambda_{n_{i-1}}^s}, \quad (13)$$

where $|\rho^s_{\vec{c}_i}|$ is the size of children set $\rho^s_{\vec{c}i}$; parameter $\lambda^{|\rho^s_{\vec{c}_i}|}_{n_{i-1}}$ is the rate of the children set $\rho^s_{\vec{c}_i}$ (Equation 2); and $\lambda^s_{n_{i-1}}$ is the rate of all possible sets given a total number of nodes n_{i-1} (Equation 2).

We can view this new proposal as a coarse-to-fine process: DPMM proposes candidate children sets; SMC selects a children set from DPMM to coalesce. Since the coarse step is faster and filters "bad" children sets, the slower finer step considers fewer children sets, saving computation time (Algorithm 1). If Ω_i has all children sets, it recovers exhaustive SMC. We estimate the effective sample size [16] and resample [17] when needed. For smaller sets, the DPMM is sometimes impractical (and only provides singleton clusters). In such cases it is simpler to enumerate all children sets.

4.3 Example Transition Kernel: Brownian Diffusion

This section uses Brownian diffusion as an example for message passing framework. The initial distribution $p_0(y)$ of each node is $\mathcal{N}(0, \infty)$; the transition kernel $\kappa_{t_i t_b}(y, \cdot)$ is a Gaussian centered at y with variance $(t_i - t_b)\Sigma$, where $\Sigma = \mu\mathbf{I}$, $\mu = \beta/2$, β is the concentration parameter of DPMM. Then the local normalizer $Z_{\rho_i}(\mathbf{x}|\theta_i)$ is

$$Z_{\rho_i}(\mathbf{x}|\theta_i) = \int \mathcal{N}(y_i; 0, \infty) \prod\nolimits_{b \in \vec{c}_i} \mathcal{N}(y_i; \hat{y}_b, \Sigma(v_{\rho_b} + t_b - t_i)) dy_i, \tag{14}$$

and the node message $M_{\rho_i}(y_i)$ is normally distributed $M_{\rho_i}(y_i) \sim \mathcal{N}(y_i; \hat{y}_{\rho_i}, \Sigma v_{\rho_i})$, where

$$v_{\rho_i} = \left(\sum\nolimits_{b \in \vec{c}_i} (v_{\rho_b} + t_b - t_i)^{-1} \right)^{-1}, \qquad \hat{y}_{\rho_i} = \left(\sum\nolimits_{b \in \vec{c}_i} \frac{\hat{y}_{\rho_b}}{v_{\rho_b} + t_b - t_i} \right) v_{\rho_i}.$$

5 Experiments: Finding Bushy Trees

In this section, we compare trees built by the beta coalescent (**beta**) against those built by Kingman's coalescent (**kingman**) and hierarchical agglomerative clustering [26, **hac**] on both synthetic and real data. We show **beta** performs best and can capture data in more interpretable, bushier trees.

Setup The parameter α for the beta coalescent is between 1 and 2. The closer α is to 1, bushier the tree is, and we set $\alpha = 1.2$.[5] We set the mutation rate as 1, thus the DPMM parameter is initialized as $\beta = 2$, and updated using slice sampling [27]. All experiments use 100 initial iterations of DPMM inference with 30 more iterations after each coalescent event (forming a new particle).

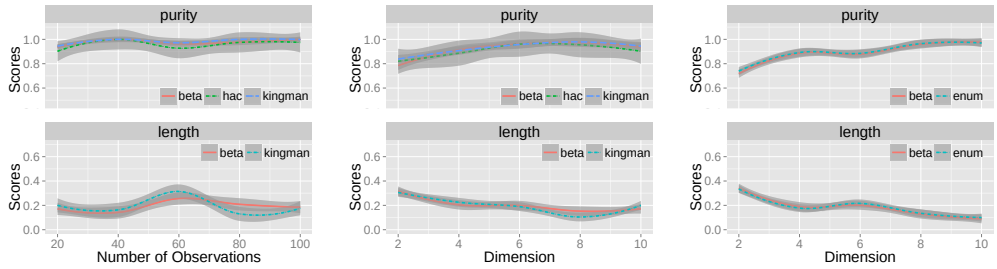
Metrics We use three metrics to evaluate the quality of the trees discovered by our algorithm: purity, subtree and path length. The dendrogram **purity** score [28, 10] measures how well the leaves in a subtree belong to the same class. For any two leaf nodes, we find the least common subsumer node s and—for the subtree rooted at s—measure the fraction of leaves with same class labels. The **subtree** score [9] is the ratio between the number of internal nodes with all children in the same class and the total number of internal nodes. The path **length** score is the average difference—over all pairs—of the lowest common subsumer distance between the true tree and the generated tree, where the lowest common subsumer distance is the distance between the root and the lowest common subsumer of two nodes. For **purity** and **subtree**, higher is better, while for **length**, lower is better. Scores are in expectation over particles and averaged across chains.

5.1 Synthetic Hierarchies

To test our inference method, we generated synthetic data with edge length (full details available in the supplemental material); we also assume each child of the root has a unique label and the descendants also have the same label as their parent node (except the root node).

We compared **beta** against **kingman** and **hac** by varying the number of observations (Figure 2(a)) and feature dimensions (Figure 2(b)). In both cases, **beta** is comparable to **kingman** and **hac** (no edge **length**). While increasing the feature dimension improves both scores, more observations do not: for synthetic data, a small number of observations suffice to construct a good tree.

[5] With DPMM proposals, α has a negligible effect, so we elide further analysis for different α values.



(a) Increasing observations (b) Increasing dimension (c) **beta** v.s. **enum**

Figure 2: Figure 2(a) and 2(b) show the effect of changing the underlying data size or number of dimension. Figure 2(c) shows that our DPMM proposal for children sets is comparable to an exhaustive enumeration of all possible children sets (**enum**).

To evaluate the effectiveness of using our DPMM as a proposal distribution, we compare exhaustively enumerating all children set candidates (**enum**) while keeping the SMC otherwise unchanged; this experiment uses ten data points (**enum** is completely intractable on larger data). **Beta** uses the DPMM and achieved similar accuracy (Figure 2(c)) while greatly improving efficiency.

5.2 Human Tissue Development

Our first real dataset is based on the developmental biology of human tissues. As a human develops, tissues specialize, starting from three embryonic germ layers: the endoderm, ectoderm, and mesoderm. These eventually form all human tissues. For example, one developmental pathway is *ectoderm* $\rightarrow$ *neural crest* $\rightarrow$ *cranial neural crest* $\rightarrow$ *optic vesicle* $\rightarrow$ *cornea*. Because each germ layer specializes into many different types of cells at specific times, it is inappropriate to model this development as a binary tree, or with clustering models lacking path lengths.

Historically, uncovering these specialization pathways is a painstaking process, requiring inspection of embryos at many stages of development; however, massively parallel sequencing data make it possible to efficiently form developmental hypotheses based on similar patterns of gene expression. To investigate this question we use the transcriptome of 27 tissues with known, unambiguous, time-specific lineages [29]. We reduce the original 182727 dimensions via principle component analysis [30, PCA]. We use five chains with five particles per chain.

Using reference developmental trees, **beta** performs better on all three scores (Table 1) because **beta** builds up a bushy hierarchy more similar to the true tree. The tree recovered by **beta** (Figure 3) reflects human development. The first major differentiation is the division of embryonic cells into three layers of tissue: endoderm, mesoderm, and ectoderm. These go on to form almost all adult organs and cells. The placenta (magenta), however, forms from a fourth cell type, the trophoblast; this is placed in its own cluster at the root of the tree. It also successfully captures ectodermal tissue lineage. However, mesodermic and endodermic tissues, which are highly diverse, do not cluster as well. Tissues known to secrete endocrine hormones (dashed borders) cluster together.

5.3 Clustering 20-newsgroups Data

Following Heller et al. [10], we also compare the three models on 20-newsgroups,[6] a multilevel hierarchy first dividing into general areas (rec, space, and religion) before specializing into areas such as baseball or hockey.[7] This true hierarchy is inset in the bottom right of Figure 4, and we assume each edge has the same length. We apply latent Dirichlet allocation [31] with 50 topics to this corpus, and use the topic distribution for each document as the document feature. We use five chains with eighty particles per chain.

[6] `http://qwone.com/~jason/20Newsgroups/`

[7] We use "rec.autos", "rec.sport.baseball", "rec.sport.hockey", "sci.space" newsgroups but also—in contrast to Heller et al. [10]—added "soc.religion.christian".

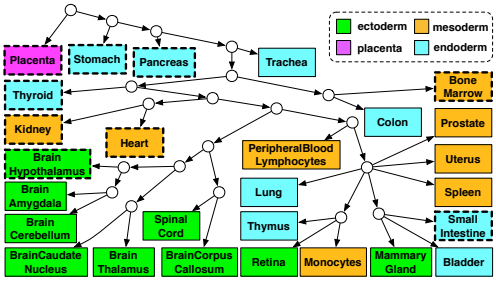


Figure 3: One sample hierarchy of human tissue from **beta**. Color indicates germ layer origin of tissue. Dashed border indicates secretory function. While neural tissues from the ectoderm were clustered correctly, some mesoderm and endoderm tissues were commingled. The cluster also preferred placing secretory tissues together and higher in the tree.

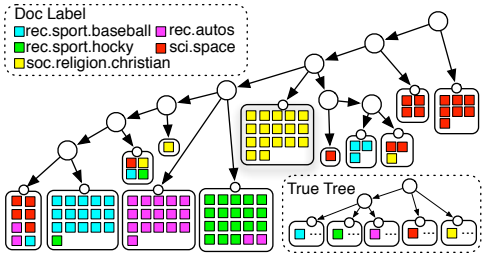


Figure 4: One sample hierarchy of the 20newsgroups from **beta**. Each small square is a document colored by its class label. Large rectangles represent a subtree with all the enclosed documents as leaf nodes. Most of the documents from the same group are clustered together; the three "rec" groups are merged together first, and then merged with the religion and space groups.

	Biological Data			20-newsgroups Data		
	hac	**kingman**	**beta**	**hac**	**kingman**	**beta**
purity ↑	0.453	0.474 ± 0.029	$\mathbf{0.492} \pm 0.028$	0.465	0.510 ± 0.047	$\mathbf{0.565} \pm 0.081$
subtree ↑	0.240	0.302 ± 0.033	$\mathbf{0.331} \pm 0.050$	0.571	0.651 ± 0.013	$\mathbf{0.720} \pm 0.013$
length ↓	–	0.654 ± 0.041	$\mathbf{0.586} \pm 0.051$	–	0.477 ± 0.027	$\mathbf{0.333} \pm 0.047$

Table 1: Comparing the three models: **beta** performs best on all three scores.

As with the biological data, **beta** performs best on all scores for 20-newsgroups. Figure 4 shows a bushy tree built by **beta**, which mostly recovered the true hierarchy. Documents within a newsgroup merge first, then the three "rec" groups, followed by "space" and "religion" groups. We only use topic distribution as features, so better results could be possible with more comprehensive features.

6 Conclusion

This paper generalizes Bayesian hierarchical clustering, moving from Kingman's coalescent to the beta coalescent. Our novel inference scheme based on SMC and DPMM make this generalization practical and efficient. This new model provides a bushier tree, often a more realistic view of data.

While we only consider real-valued vectors, which we model through the ubiquitous Gaussian, other likelihoods might be better suited to other applications. For example, for discrete data such as in natural language processing, a multinomial likelihood may be more appropriate. This is a straightforward extension of our model via other transition kernels and DPMM base distributions.

Recent work uses the coalescent as a means of producing a clustering in tandem with a downstream task such as classification [32]. Hierarchies are often taken *a priori* in natural language processing. Particularly for linguistic tasks, a fully statistical model like the beta coalescent that jointly learns the hierarchy and a downstream task could improve performance in dependency parsing [33] (clustering parts of speech), multilingual sentiment [34] (finding sentiment-correlated words across languages), or topic modeling [35] (finding coherent words that should co-occur in a topic).

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments, and thank Héctor Corrada Bravo for pointing us to human tissue data. This research was supported by NSF grant #1018625. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsor.

References

[1] Kaufman, L., P. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.

[2] Jain, A. K. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[3] Brown, P. F., V. J. D. Pietra, P. V. deSouza, et al. Class-based n-gram models of natural language. *Computational Linguistics*, 18:18–4, 1990.

[4] Bergen, J., P. Anandan, K. Hanna, et al. Hierarchical model-based motion estimation. In *ECCV*. 1992.

[5] Girvan, M., M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99:7821–7826, 2002.

[6] Kingman, J. F. C. On the genealogy of large populations. *Journal of Applied Probability*, 19:27–43, 1982.

[7] Pitman, J. Coalescents with multiple collisions. *The Annals of Probability*, 27:1870–1902, 1999.

[8] Berestycki, N. Recent progress in coalescent theory. In *Ensaios Matematicos*, vol. 16. 2009.

[9] Teh, Y. W., H. Daumé III, D. M. Roy. Bayesian agglomerative clustering with coalescents. In *NIPS*. 2008.

[10] Heller, K. A., Z. Ghahramani. Bayesian hierarchical clustering. In *ICML*. 2005.

[11] Blundell, C., Y. W. Teh, K. A. Heller. Bayesian rose trees. In *UAI*. 2010.

[12] Adams, R., Z. Ghahramani, M. Jordan. Tree-structured stick breaking for hierarchical data. In *NIPS*. 2010.

[13] Knowles, D., Z. Ghahramani. Pitman-Yor diffusion trees. In *UAI*. 2011.

[14] Neal, R. M. Density modeling and clustering using Dirichlet diffusion trees. *Bayesian Statistics*, 7:619–629, 2003.

[15] Sagitov, S. The general coalescent with asynchronous mergers of ancestral lines. *Journal of Applied Probability*, 36:1116–1125, 1999.

[16] Neal, R. M. Annealed importance sampling. *Technical report 9805, University of Toronto*, 1998.

[17] Fearhhead, P. Sequential Monte Carlo method in filter theory. *PhD thesis, University of Oxford*, 1998.

[18] Felsenstein, J. Maximum-likelihood estimation of evolutionary trees from continuous characters. *Am J Hum Genet*, 25(5):471–492, 1973.

[19] Birkner, M., J. Blath, M. Steinrucken. Importance sampling for lambda-coalescents in the infinitely many sites model. *Theoretical population biology*, 79(4):155–73, 2011.

[20] Doucet, A., N. De Freitas, N. Gordon, eds. *Sequential Monte Carlo methods in practice*. 2001.

[21] Gordon, N., D. Salmond, A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings F, Radar and Signal Processing*, 140(2):107–113, 1993.

[22] Görür, D., L. Boyles, M. Welling. Scalable inference on Kingman's coalescent using pair similarity. *JMLR*, 22:440–448, 2012.

[23] Antoniak, C. E. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

[24] Cappe, O., S. Godsill, E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *PROCEEDINGS-IEEE*, 95(5):899, 2007.

[25] Chen, Z. Bayesian filtering: From kalman filters to particle filters, and beyond. *McMaster*, [Online], 2003.

[26] Eads, D. Hierarchical clustering (scipy.cluster.hierarchy). *SciPy*, 2007.

[27] Neal, R. M. Slice sampling. *Annals of Statistics*, 31:705–767, 2003.

[28] Powers, D. M. W. Unsupervised learning of linguistic structure an empirical evaluation. *International Journal of Corpus Linguistics*, 2:91–131, 1997.

[29] Jongeneel, C., M. Delorenzi, C. Iseli, et al. An atlas of human gene expression from massively parallel signature sequencing (mpss). *Genome Res*, 15:1007–1014, 2005.

[30] Shlens, J. A tutorial on principal component analysis. In *Systems Neurobiology Laboratory, Salk Institute for Biological Studies*. 2005.

[31] Blei, D. M., A. Ng, M. Jordan. Latent Dirichlet allocation. *JMLR*, 2003.

[32] Rai, P., H. Daumé III. The infinite hierarchical factor regression model. In *NIPS*. 2008.

[33] Koo, T., X. Carreras, M. Collins. Simple semi-supervised dependency parsing. In *ACL*. 2008.

[34] Boyd-Graber, J., P. Resnik. Holistic sentiment analysis across languages: Multilingual supervised latent Dirichlet allocation. In *EMNLP*. 2010.

[35] Andrzejewski, D., X. Zhu, M. Craven. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *ICML*. 2009.