Bayesian Agglomerative Clustering with Coalescents

Yee Whye Teh
Gatsby Unit
University College London
ywteh@gatsby.ucl.ac.uk

Hal Daumé III
School of Computing
University of Utah
me@hal3.name

Daniel Roy
CSAIL
MIT
droy@mit.edu

Abstract

We introduce a new Bayesian model for hierarchical clustering based on a prior over trees called Kingman's coalescent. We develop novel greedy and sequential Monte Carlo inferences which operate in a bottom-up agglomerative fashion. We show experimentally the superiority of our algorithms over others, and demonstrate our approach in document clustering and phylolinguistics.

1 Introduction

Hierarchically structured data abound across a wide variety of domains. It is thus not surprising that hierarchical clustering is a traditional mainstay of machine learning [1]. The dominant approach to hierarchical clustering is agglomerative: start with one cluster per datum, and greedily merge pairs until a single cluster remains. Such algorithms are efficient and easy to implement. Their primary limitations—a lack of predictive semantics and a coherent mechanism to deal with missing data—can be addressed by probabilistic models that handle partially observed data, quantify goodness-of-fit, predict on new data, and integrate within more complex models, all in a principled fashion.

Currently there are two main approaches to probabilistic models for hierarchical clustering. The first takes a direct Bayesian approach by defining a prior over trees followed by a distribution over data points conditioned on a tree [2, 3, 4, 5]. MCMC sampling is then used to obtain trees from their posterior distribution given observations. This approach has the advantages and disadvantages of most Bayesian models: averaging over sampled trees can improve predictive capabilities, give confidence estimates for conclusions drawn from the hierarchy, and share statistical strength across the model; but it is also computationally demanding and complex to implement. As a result such models have not found widespread use. [2] has the additional advantage that the distribution induced on the data points is exchangeable, so the model can be coherently extended to new data. The second approach uses a flat mixture model as the underlying probabilistic model and structures the posterior hierarchically [6, 7]. This approach uses an agglomerative procedure to find the tree giving the best posterior approximation, mirroring traditional agglomerative clustering techniques closely and giving efficient and easy to implement algorithms. However because the underlying model has no hierarchical structure, there is no sharing of information across the tree.

We propose a novel class of Bayesian hierarchical clustering models and associated inference algorithms combining the advantages of both probabilistic approaches above. 1) We define a prior and compute the posterior over trees, thus reaping the benefits of a fully Bayesian approach; 2) the distribution over data is hierarchically structured allowing for sharing of statistical strength; 3) we have efficient and easy to implement inference algorithms that construct trees agglomeratively; and 4) the induced distribution over data points is exchangeable. Our model is based on an exchangeable distribution over trees called Kingman's coalescent [8, 9]. Kingman's coalescent is a standard model from population genetics for the genealogy of a set of individuals. It is obtained by tracing the genealogy backwards in time, noting when lineages *coalesce* together. We review Kingman's coalescent in Section 2. Our own contribution is in using it as a prior over trees in a hierarchical clustering model (Section 3) and in developing novel inference procedures for this model (Section 4).

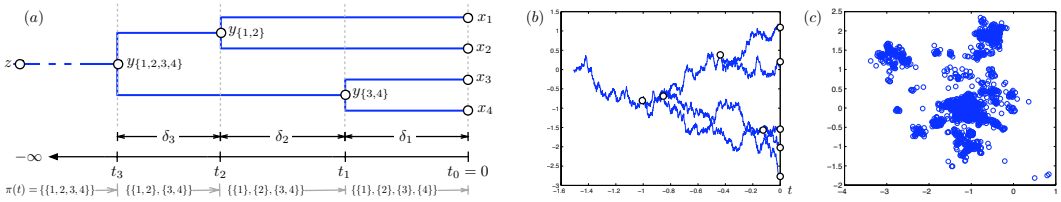


Figure 1: (a) Variables describing the n-coalescent. (b) Sample path from a Brownian diffusion coalescent process in 1D, circles are coalescent points. (c) Sample observed points from same in 2D, notice the hierarchically clustered nature of the points.

2 Kingman's coalescent

Kingman's coalescent is a standard model in population genetics describing the common genealogy (ancestral tree) of a set of individuals [8, 9]. In its full form it is a distribution over the genealogy of a countably infinite set of individuals. Like other nonparametric models (e.g. Gaussian and Dirichlet processes), Kingman's coalescent is most easily described and understood in terms of its finite dimensional marginal distributions over the genealogies of n individuals, called n-coalescents. We obtain Kingman's coalescent as $n \to \infty$.

Consider the genealogy of n individuals alive at the present time $t = 0$. We can trace their ancestry backwards in time to the distant past $t = -\infty$. Assume each individual has one parent (in genetics, *haploid* organisms), and therefore genealogies of $[n] = \{1, \ldots, n\}$ form a *directed forest*. In general, at time $t \leq 0$, there are m ($1 \leq m \leq n$) ancestors alive. Identify these ancestors with their corresponding sets $\rho_1, \ldots, \rho_m$ of descendants (we will make this identification throughout the paper). Note that $\pi(t) = \{\rho_1, \ldots, \rho_m\}$ form a *partition* of $[n]$, and interpret $t \mapsto \pi(t)$ as a function from $(-\infty, 0]$ to the set of partitions of $[n]$. This function is piecewise constant, left-continuous, monotonic ($s \leq t$ implies that $\pi(t)$ is a refinement of $\pi(s)$), and $\pi(0) = \{\{1\}, \ldots, \{n\}\}$ (see Figure 1a). Further, π *completely and succinctly* characterizes the genealogy; we shall henceforth refer to π as *the genealogy* of $[n]$.

Kingman's n-coalescent is simply a distribution over genealogies of $[n]$, or equivalently, over the space of partition-valued functions like π. More specifically, the n-coalescent is a continuous-time, partition-valued, Markov process, which starts at $\{\{1\}, \ldots, \{n\}\}$ at present time $t = 0$, and evolves *backwards in time*, merging (coalescing) lineages until only one is left. To describe the Markov process in its entirety, it is sufficient to describe the jump process (i.e. the embedded, discrete-time, Markov chain over partitions) and the distribution over coalescent times. Both are straightforward and their simplicity is part of the appeal of Kingman's coalescent. Let ρ_{li}, ρ_{ri} be the ith pair of lineages to coalesce, $t_{n-1} < \cdots < t_1 < t_0 = 0$ be the coalescent times and $\delta_i = t_{i-1} - t_i > 0$ be the duration between adjacent events (see Figure 1a). Under the n-coalescent, every pair of lineages merges independently with rate 1. Thus the first pair amongst m lineages merge with rate $\binom{m}{2} = \frac{m(m-1)}{2}$. Therefore $\delta_i \sim \text{Exp}\left(\binom{n-i+1}{2}\right)$ independently, the pair ρ_{li}, ρ_{ri} is chosen from among those right after time t_i, and with probability one a random draw from the n-coalescent is a binary tree with a single root at $t = -\infty$ and the n individuals at time $t = 0$. The genealogy is given as:

$$\pi(t) = \begin{cases} \{\{1\}, \ldots, \{n\}\} & \text{if } t = 0; \\ \pi_{t_{i-1}} - \rho_{li} - \rho_{ri} + (\rho_{li} \cup \rho_{ri}) & \text{if } t = t_i; \\ \pi_{t_i} & \text{if } t_{i+1} < t < t_i. \end{cases} \tag{1}$$

Combining the probabilities of the durations and choices of lineages, the probability of π is simply:

$$p(\pi) = \prod_{i=1}^{n-1} \binom{n-i+1}{2} \exp\left(-\binom{n-i+1}{2}\delta_i\right) / \binom{n-i+1}{2} = \prod_{i=1}^{n-1} \exp\left(-\binom{n-i+1}{2}\delta_i\right) \tag{2}$$

The n-coalescent has some interesting statistical properties [8, 9]. The marginal distribution over tree topologies is uniform and independent of the coalescent times. Secondly, it is infinitely exchangeable: given a genealogy drawn from an n-coalescent, the genealogy of any m contemporary individuals alive at time $t \leq 0$ embedded within the genealogy is a draw from the m-coalescent. Thus, taking $n \to \infty$, there is a distribution over genealogies of a countably infinite population for which the marginal distribution of the genealogy of any n individuals gives the n-coalescent. Kingman called this *the coalescent*.

3 Hierarchical clustering with coalescents

We take a Bayesian approach to hierarchical clustering, placing a coalescent prior on the latent tree and modeling observed data with a Markov process evolving *forward in time* along the tree. We will alter our terminology from genealogy to tree, from n individuals at present time to n observed data points, and from individuals on the genealogy to latent variables on the tree-structured distribution. Let $x_1, \ldots, x_n$ be n observed data at the leaves of a tree π drawn from the n-coalescent. π has $n-1$ coalescent points, the ith occuring when ρ_{li} and ρ_{ri} merge at time t_i to form $\rho_i = \rho_{li} \cup \rho_{ri}$. Let t_{li} and t_{ri} be the times at which ρ_{li} and ρ_{ri} are themselves formed.

We construct a continuous-time Markov process evolving along the tree from the past to the present, branching independently at each coalescent point until we reach time 0, where the n Markov processes induce a distribution over the n data points. The joint distribution respects the conditional independences implied by the structure of the directed tree. Let y_{ρ_i} be a latent variable that takes on the value of the Markov process at ρ_i just before it branches (see Figure 1a). Let $y_{\{i\}} = x_i$ at leaf i.

To complete the description of the likelihood model, let $q(z)$ be the initial distribution of the Markov process at time $t = -\infty$, and $k_{st}(x, y)$ be the transition probability from state x at time s to state y at time t. This Markov process need be neither stationary nor ergodic. Marginalizing over paths of the Markov process, the joint probability over the latent variables and the observations is:

$$p(\mathbf{x}, \mathbf{y}, z|\pi) = q(z) k_{-\infty\, t_{n-1}}(z, y_{\rho_{n-1}}) \prod_{i=1}^{n-1} k_{t_i t_{li}}(y_{\rho_i}, y_{\rho_{li}}) k_{t_i t_{ri}}(y_{\rho_i}, y_{\rho_{ri}}) \tag{3}$$

Notice that the marginal distributions at each observation $p(x_i|\pi)$ are identical and given by the Markov process at time 0. However, they are not independent: they share the same sample path down the Markov process until they split. In fact the amount of dependence between two observations is a function of the time at which the observations coalesce in the past. A more recent coalescent time implies larger dependence. The overall distribution induced on the observations $p(\mathbf{x})$ inherits the infinite exchangeability of the n-coalescent. We considered a brownian diffusion (see Figures 1(b,c)) and a simple independent sites mutation process on multinomial vectors (Section 4.3).

4 Agglomerative sequential Monte Carlo and greedy inference

We develop two classes of efficient and easily implementable inference algorithms for our hierarchical clustering model based on sequential Monte Carlo (SMC) and greedy schemes respectively. In both classes, the latent variables are integrated out, and the trees are constructed in a bottom-up fashion. The full tree π can be expressed as a series of $n-1$ coalescent events, ordered backwards in time. The ith coalescent event involves the merging of the two subtrees with leaves ρ_{li} and ρ_{ri} and occurs at a time δ_i before the previous coalescent event. Let $\theta_i = \{\delta_j, \rho_{lj}, \rho_{rj} \text{ for } j \leq i\}$ denote the first i coalescent events. θ_{n-1} is equivalent to π and we shall use them interchangeably.

We assume that the form of the Markov process is such that the latent variables $\{y_{\rho_i}\}_{i=1}^{n-1}$ and z can be efficiently integrated out using an upward pass of belief propagation on the tree. Let $M_{\rho_i}(y)$ be the message passed from y_{ρ_i} to its parent; $M_{\{i\}}(y) = \delta_{x_i}(y)$ is point mass at x_i for leaf i. $M_{\rho_i}(y)$ is proportional to the likelihood of the observations at the leaves below coalescent event i, given that $y_{\rho_i} = y$. Belief propagation computes the messages recursively up the tree; for $i = 1, \ldots, n-1$:

$$M_{\rho_i}(y) = Z_{\rho_i}^{-1}(\mathbf{x}, \theta_i) \prod_{b=l,r} \int k_{t_i t_{bi}}(y, y_b) M_{\rho_{bi}}(y_b)\, dy_b \tag{4}$$

$Z_{\rho_i}(\mathbf{x}, \theta_i)$ is a normalization constant introduced to avoid numerical problems. The choice of Z does not affect the probability of $\mathbf{x}$, but does impact the accuracy and efficiency of our inference algorithms. We found that $Z_{\rho_i}(\mathbf{x}, \theta_i) = \int q(y) M_{\rho_i}(y)\, dy$ worked well. At the root, we have:

$$Z_{-\infty}(\mathbf{x}, \theta_{n-1}) = \int q(z) k_{-\infty\, t_{n-1}}(z, y) M_{\rho_{n-1}}(y)\, dy\, dz \tag{5}$$

The marginal probability $p(\mathbf{x}|\pi)$ is now given by the product of normalization constants:

$$p(\mathbf{x}|\pi) = Z_{-\infty}(\mathbf{x}, \theta_{n-1}) \prod_{i=1}^{n-1} Z_{\rho_i}(\mathbf{x}, \theta_i) \tag{6}$$

Multiplying in the prior (2) over π, we get the joint probability for the tree π and observations $\mathbf{x}$:

$$p(\mathbf{x}, \pi) = Z_{-\infty}(\mathbf{x}, \theta_{n-1}) \prod_{i=1}^{n-1} \exp\left(-\tbinom{n-i+1}{2} \delta_i\right) Z_{\rho_i}(\mathbf{x}, \theta_i) \tag{7}$$

Our inference algorithms are based upon (7). Note that each term $Z_{\rho_i}(\mathbf{x}, \theta_i)$ can be interpreted as a local likelihood term for coalescing the pair ρ_{li}, ρ_{ri}[1]. In general, for each i, we choose a duration δ_i and a pair of subtrees ρ_{li}, ρ_{ri} to coalesce. This choice is based upon the ith term in (7), interpreted as the product of a local prior and a local likelihood for choosing δ_i, ρ_{li} and ρ_{ri} given θ_{i-1}.

4.1 Sequential Monte Carlo algorithms

Sequential Monte Carlo algorithms (aka particle filters), approximate the posterior using a weighted sum of point masses [10]. These point masses are constructed iteratively. At iteration $i-1$, particle s consists of $\theta^s_{i-1} = \{\delta^s_j, \rho^s_{lj}, \rho^s_{rj} \text{ for } j < i\}$, and has weight w^s_{i-1}. At iteration i, s is extended by sampling δ^s_i, ρ^s_{li} and ρ^s_{ri} from a proposal distribution $f_i(\delta^s_i, \rho^s_{li}, \rho^s_{ri}|\theta^s_{i-1})$, with weights:

$$w^s_i = w^s_{i-1} \exp\left(-\tbinom{n-i+1}{2}\delta^s_i\right) Z_{\rho_i}(\mathbf{x}, \theta^s_i) / f_i(\delta^s_i, \rho^s_{li}, \rho^s_{ri}|\theta^s_{i-1}) \tag{8}$$

After $n-1$ iterations, we obtain a set of trees θ^s_{n-1} and weights w^s_{n-1}. The joint distribution is approximated by: $p(\pi, \mathbf{x}) \approx \sum_s w^s_{n-1} \delta_{\theta^s_{n-1}}(\pi)$, while the posterior is approximated with the weights normalized. An important aspect of SMC is resampling, which places more particles in high probability regions and prunes particles stuck in low probability regions. We resample as in Algorithm 5.1 of [11] when the effective sample size ratio as estimated in [12] falls below one half.

SMC-PriorPrior. The simplest proposal distribution is to sample δ^s_i, ρ^s_{li} and ρ^s_{ri} from the local prior. δ^s_i is drawn from an exponential with rate $\binom{n-i+1}{2}$ and ρ^s_{li}, ρ^s_{ri} are drawn uniformly from all available pairs. The weight updates (8) reduce to multiplying by $Z_{\rho_i}(\mathbf{x}, \theta^s_i)$. This approach is computationally very efficient, but performs badly with many objects due to the uniform draws over pairs. **SMC-PriorPost**. The second approach addresses the suboptimal choice of pairs to coalesce. We first draw δ^s_i from its local prior, then draw ρ^s_{li}, ρ^s_{ri} from the local posterior:

$$f_i(\rho^s_{li}, \rho^s_{ri}|\delta^s_i, \theta^s_{i-1}) \propto Z_{\rho_i}(\mathbf{x}, \theta^s_{i-1}, \delta^s_i, \rho^s_{li}, \rho^s_{ri}); \quad w^s_i = w^s_{i-1} \textstyle\sum_{\rho'_l, \rho'_r} Z_{\rho_i}(\mathbf{x}, \theta^s_{i-1}, \delta^s_i, \rho'_l, \rho'_r) \tag{9}$$

This approach is more computationally demanding since we need to evaluate the local likelihood of every pair. It also performs significantly better than SMC-PriorPrior. We have found that it works reasonably well for small data sets but fails in larger ones for which the local posterior for δ_i is highly peaked. **SMC-PostPost**. The third approach is to draw all of δ^s_i, ρ^s_{li} and ρ^s_{ri} from their posterior:

$$\begin{aligned} f_i(\delta^s_i, \rho^s_{li}, \rho^s_{ri}|\theta^s_{i-1}) &\propto \exp\left(-\tbinom{n-i+1}{2}\delta^s_i\right) Z_{\rho_i}(\mathbf{x}, \theta^s_{i-1}, \delta^s_i, \rho^s_{li}, \rho^s_{ri}) \\ w^s_i &= w^s_{i-1} \textstyle\sum_{\rho'_l, \rho'_r} \int \exp\left(-\tbinom{n-i+1}{2}\delta'\right) Z_{\rho_i}(\mathbf{x}, \theta^s_{i-1}, \delta', \rho'_l, \rho'_r)\, d\delta' \end{aligned} \tag{10}$$

This approach requires the fewest particles, but is the most computationally expensive due to the integral for each pair. Fortunately, for the case of Brownian diffusion process described below, these integrals are tractable and related to generalized inverse Gaussian distributions.

4.2 Greedy algorithms

SMC algorithms are attractive because they produce an arbitrarily accurate approximation to the full posterior. However in many applications a single good tree is often times sufficient. We describe a few greedy algorithms to construct a good tree.

Greedy-MaxProb: the obvious greedy algorithm is to pick δ_i, ρ_{li} and ρ_{ri} maximizing the ith term in (7). We do so by computing the optimal δ_i for each pair of ρ_{li}, ρ_{ri}, and then picking the pair maximizing the ith term at its optimal δ_i. **Greedy-MinDuration**: simply pick the pair to coalesce whose optimal duration is minimum. Both algorithms require recomputing the optimal duration for each pair at each iteration, since the exponential rate $\binom{n-i+1}{2}$ on the duration varies with the iteration i. The total computational cost is thus $O(n^3)$. We can avoid this by using the alternative view of the n-coalesent as a Markov process where each pair of lineages coalesces at rate 1. **Greedy-Rate1**: for each pair ρ_{li} and ρ_{ri} we determine the optimal δ_i, but replacing the $\binom{n-i+1}{2}$ prior rate with 1. We coalesce the pair with most recent time (as in Greedy-MinDuration). This reduces the complexity to $O(n^2)$. We found that all three perform about equally well.

[1] If the Markov process is stationary with equilibrium $q(y)$, $Z_{\rho_i}(\mathbf{x}, \theta_i)$ is a likelihood ratio between two models with observations $\mathbf{x}_{\rho_i}$: (1) a single tree with leaves ρ_i; (2) two independent trees with leaves ρ_{li} and ρ_{ri} respectively. This is similar to [6, 7] and is used later in our NIPS experiment to determine coherent clusters.

4.3 Examples

Brownian diffusion. Consider the case of continuous data evolving via Brownian diffusion. The transition kernel $k_{st}(y,\cdot)$ is a Gaussian centred at y with variance $(t-s)\Lambda$, where Λ is a symmetric p.d. covariance matrix. Because the joint distribution (3) over $\mathbf{x}$, $\mathbf{y}$ and z is Gaussian, we can express each message $M_{\rho_i}(y)$ as a Gaussian with mean $\widehat{y}_{\rho_i}$ and variance Λv_{ρ_i}. The local likelihood is:

$$Z_{\rho_i}(\mathbf{x},\theta_i) = |2\pi\widehat{\Lambda}_i|^{-\frac{1}{2}} \exp\big(-\tfrac{1}{2}\,||\widehat{y}_{\rho_{li}} - \widehat{y}_{\rho_{ri}}||^2_{\widehat{\Lambda}_i}\big); \qquad \widehat{\Lambda}_i = \Lambda(v_{\rho_{li}} + v_{\rho_{ri}} + t_{li} + t_{ri} - 2t_i) \tag{11}$$

where $\|x\|_\Psi = x^\top \Psi^{-1} x$ is the Mahanalobis norm. The optimal duration δ_i can also be solved for,

$$\delta_i = \frac{1}{4\binom{n-i+1}{2}}\left(\sqrt{4\binom{n-i+1}{2}\,||\widehat{y}_{\rho_{li}} - \widehat{y}_{\rho_{ri}}||^2_\Lambda + D^2} - D\right) - \tfrac{1}{2}(v_{\rho_{li}} + v_{\rho_{ri}} + t_{li} + t_{ri} - 2t_{i-1}) \tag{12}$$

where D is the dimensionality. The message at the newly coalesced point has mean and covariance:

$$v_{\rho_i} = \big((v_{\rho_{li}} + t_{li} - t_i)^{-1} + (v_{\rho_{ri}} + t_{ri} - t_i)^{-1}\big)^{-1}; \widehat{y}_{\rho_i} = \big(\tfrac{\widehat{y}_{\rho_{li}}}{v_{\rho_{li}} + t_{li} - t_i} + \tfrac{\widehat{y}_{\rho_{ri}}}{v_{\rho_{ri}} + t_{ri} - t_i}\big) v_{\rho_i} \tag{13}$$

Multinomial vectors. Consider a Markov process acting on multinomial vectors with each entry taking one of K values and evolving independently. Entry d evolves at rate λ_d and has equilibrium distribution vector q_d. The transition rate matrix is $Q_d = \lambda_d(q_h^\top \mathbf{1}_K - I_k)$ where $\mathbf{1}_K$ is a vector of K ones and I_K is identity matrix of size K, while the transition probability matrix for entry d in a time interval of length t is $e^{Q_d t} = e^{-\lambda_d t} I_K + (1 - e^{-\lambda_d t}) q_d^\top \mathbf{1}_K$. Representing the message for entry d from ρ_i to its parent as a vector $M^d_{\rho_i} = [M^{d1}_{\rho_i}, \ldots, M^{dK}_{\rho_i}]^\top$, normalized so that $q_d \cdot M^d_{\rho_i} = 1$, the local likelihood terms and messages are computed as,

$$Z^d_{\rho_i}(\mathbf{x},\theta_i) = 1 - e^{\lambda_h(2t_i - t_{li} - t_{ri})}\big(1 - \textstyle\sum_{k=1}^K q_{dk} M^{dk}_{\rho_{li}} M^{dk}_{\rho_{ri}}\big) \tag{14}$$

$$M^d_{\rho_i} = (1 - e^{\lambda_d(t_i - t_{li})}(1 - M^d_{\rho_{li}}))(1 - e^{\lambda_d(t_i - t_{ri})}(1 - M^d_{\rho_{ri}}))/Z^d_{\rho_i}(\mathbf{x},\theta_i) \tag{15}$$

Unfortunately the optimal δ_i cannot be solved analytically and we use Newton steps to compute it.

4.4 Hyperparameter estimation and predictive density

We perform hyperparameter estimation by iterating between estimating a geneology, then re-estimating the hyperparamters conditioned on this tree. Space precludes a detailed discussion of the algorithms we use; they can be found in the supplemental material. In the Brownian case, we place an inverse Wishart prior on Λ and the MAP posterior $\hat{\Lambda}$ is available in a standard closed form. In the multinomial case, the updates are not available analytically and must be solved iteratively.

Given a tree and a new individual y' we wish to know: (a) where y' might coalescent and (b) what the density is at y'. In the supplemental material, we show that the probability that y' merges at time t with a given sibling is available in closed form for the Brownian motion case. To obtain the density, we sum over all possible siblings and integrate out t by drawing equally spaced samples.

5 Experiments

Synthetic Data Sets In Figure 2 we compare the various SMC algorithms and Greedy-Rate1[2] on a range of synthetic data sets drawn from the Brownian diffusion coalescent process itself ($\Lambda = I_D$) to investigate the effects of various parameters on the efficacy of the algorithms. Generally SMC-PostPost performed best, followed by SMC-PriorPost, SMC-PriorPrior and Greedy-Rate1. With increasing D the amount of data given to the algorithms increases and all algorithms do better, especially Greedy-Rate1. This is because the posterior becomes concentrated and the Greedy-Rate1 approximation corresponds well with the posterior. As n increases, the amount of data increases as well and all algorithms perform better[3]. However, the posterior space also increases and SMC-PriorPrior which simply samples from the prior over genealogies does not improve as much. We see this effect as well when S is small. As S increases all SMC algorithms improve. Finally, the algorithms were surprisingly robust when there is mismatch between the generated data sets' λ and the λ used by the model. We expected all models to perform worse with SMC-PostPost best able to maintain its performance (though this is possibly due to our experimental setup).

[2]We found in unreported experiments that the greedy algorithms worked about equally well.

[3]Each panel was generated from independent runs. Data set variance affected all algorithms, varying overall performance across panels. However, trends in each panel are still valid, as they are based on the same data.

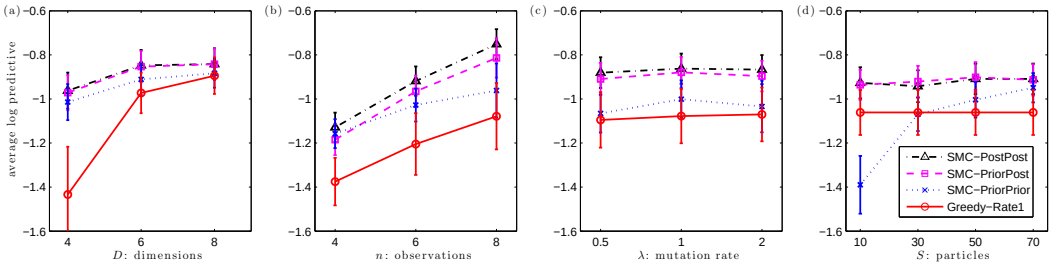


Figure 2: Predictive performance of algorithms as we vary (a) the numbers of dimensions D, (b) observations n, (c) the mutation rate λ ($\Lambda = \lambda I_D$), and (d) number of samples S. In each panel other parameters are fixed to their middle values (we used $S = 50$) in other panels, and we report log predictive probabilities on one unobserved entry, averaged over 100 runs.

	MNIST			SPAMBASE		
	Avg-link	**BHC**	**Coalescent**	**Avg-link**	**BHC**	**Coalescent**
Purity	.363±.004	.392±.006	**.412±.006**	.616±.007	**.711±.010**	.689±.008
Subtree	.581±.005	.579±.005	**.610±.005**	.607±.011	.549±.015	**.661±.012**
LOO-acc	.755±.005	.763±.005	**.773±.005**	.846±.010	.832±.010	**.861±.008**

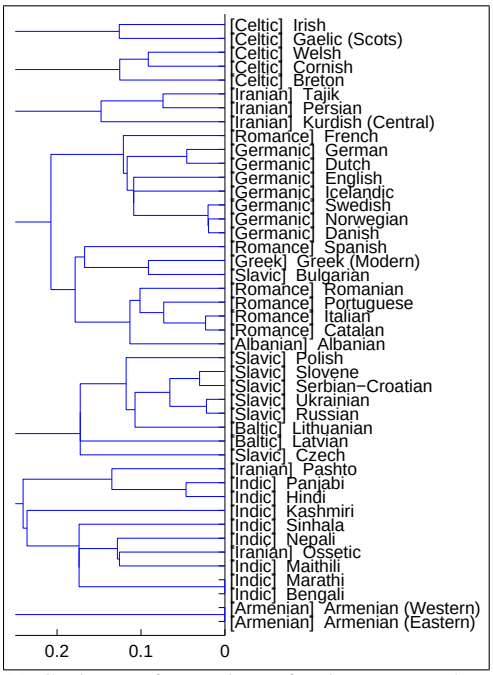
Table 1: Comparative results. Numbers are averages and standard errors over 50 and 20 repeats.

MNIST and SPAMBASE We compare the performance of our approach (Greedy-Rate1 with 10 iterations of hyperparameter update) to two other hierarchical clustering algorithms: average-link agglomerative clustering and Bayesian hierarchical clustering [6]. In MNIST, We use 10 digits from the MNIST data set, 20 examplars for each digit and 20 dimensions (reduced via PCA), repeating the experiment 50 times. In SPAMBASE, we use 100 examples of 57 attributes each from 2 classes, repeating 20 times. We present purity scores [6], subtree scores ($\#\{$interior nodes with all leaves of same class$\}/(n - \#$classes)) and leave-one-out accuracies (all scores between 0 and 1, higher better). The results are in Table 1; as we can see, except for purity on SPAMBASE, ours gives the best performance. Experiments not presented here show that all greedy algorithms perform about the same and that performance improves with hyperparameter updates.

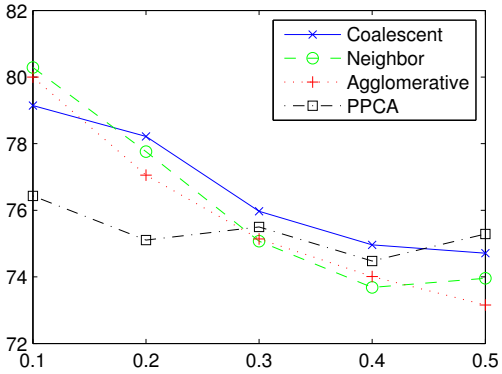
Phylolinguistics We apply our approach (Greedy-Rate1) to a phylolinguistic problem: language evolution. Unlike previous research [13] which studies only phonological data, we use a full typological database of 139 binary features over 2150 languages: the *World Atlas of Language Structures* (henceforth, "WALS") [14]. The data is *sparse*: about 84% of the entries are unknown. We use the same version of the database as extracted by [15]. Based on the Indo-European subset of this data for which at most 30 features are unknown (48 language total), we recover the coalescent tree shown in Figure 3(a). Each language is shown with its genus, allowing us to observe that it teases apart Germanic and Romance languages, but makes a few errors with respect to Iranian and Greek. (In the supplemental material, we report results applied to a wider range of languages.)

Next, we compare predictive abilities to other algorithms. We take a subset of WALS and tested on 5% of withheld entries, restoring these with various techniques: Greedy-Rate1; nearest neighbors (use value from nearest observed neighbor); average-linkage (nearest neighbor in the tree); and probabilistic PCA (latent dimensions in $5, 10, 20, 40$, chosen optimistically). We use five subsets of the WALS database of varying size, obtained by sorting both the languages and features of the database according to how many cells are observed. We then use a varying percentage ($10\% - 50\%$) of the densest portion. The results are in Figure 3(b). The performance of PPCA is steady around 76%. The performance of the other algorithms degrades as the sparsity incrases. Our approach performs at least as well as all the other techniques, except at the two extremes.

NIPS We applied Greedy-Rate1 to all NIPS abstracts through NIPS12 (1740, total). The data was preprocessed so that only words occuring in at least 100 abstracts were retained. The word counts were then converted to binary. We performed one iteration of hyperparameter re-estimation. In the supplemental material, we depict the top levels of the coalescent tree. Here, we use use the tree to



(a) Coalescent for a subset of Indo-European languages from WALS.



(b) Data restoration on WALS. Y-axis is accuracy; X-axis is percentage of data set used in experiments. At 10%, there are $N = 215$ languages, $H = 14$ features and $p = 94\%$ observed data; at 20%, $N = 430$, $H = 28$ and $p = 80\%$; at 30%: $N = 645$, $H = 42$ and $p = 66\%$; at 40%: $N = 860$, $H = 56$ and $p = 53\%$; at 50%: $N = 1075$, $H = 70$ and $p = 43\%$. Results are averaged over five folds with a different 5% hidden each time. (We also tried a "mode" prediction, but its performance is in the 60% range in all cases, and is not depicted.)

Figure 3: Results of the phylolinguistics experiments.

LLR *(t)*	**Top Words**	**Top Authors**
32.7 *(-2.71)*	bifurcation attractors hopfield network saddle	Mjolsness (9) Saad (9) Ruppin (8) Coolen (7)
0.106 *(-3.77)*	voltage model cells neurons neuron	Koch (30) Sejnowski (22) Bower (11) Dayan (10)
83.8 *(-2.02)*	chip circuit voltage vlsi transistor	Koch (12) Alspector (6) Lazzaro (6) Murray (6)
140.0 *(-2.43)*	spike ocular cells firing stimulus	Sejnowski (22) Koch (18) Bower (11) Dayan (10)
2.48 *(-3.66)*	data model learning algorithm training	Jordan (17) Hinton (16) Williams (14) Tresp (13)
31.3 *(-2.76)*	infomax image ica images kurtosis	Hinton (12) Sejnowski (10) Amari (7) Zemel (7)
31.6 *(-2.83)*	data training regression learning model	Jordan (16) Tresp (13) Smola (11) Moody (10)
39.5 *(-2.46)*	critic policy reinforcement agent controller	Singh (15) Barto (10) Sutton (8) Sanger (7)
23.0 *(-3.03)*	network training units hidden input	Mozer (14) Lippmann (11) Giles (10) Bengio (9)

Table 2: Nine clusters discovered in NIPS abstracts data.

generate a flat clustering. To do so, we use the log likelihood ratio at each branch in the coalescent to determine if a split should occur. If the log likelihood ratio is greater than zero, we break the branch; otherwise, we recurse down. On the NIPS abstracts, this leads to nine clusters, depicted in Table 2. Note that clusters two and three are quite similar—had we used a slighly higher log likelihood ratio, they would have been merged (the LLR for cluster 2 was only 0.105). Note that the clustering is able to tease apart Bayesian learning (cluster 5) and non-bayesian learning (cluster 7)—both of which have Mike Jordan as their top author!

6 Discussion

We described a new model for Bayesian agglomerative clustering. We used Kingman's coalescent as our prior over trees, and derived efficient and easily implementable greedy and SMC inference algorithms for the model. We showed empirically that our model gives better performance than other agglomerative clustering algorithms, and gives good results on applications to document modeling and phylolinguistics.

Our model is most similar in spirit to the Dirichlet diffusion tree of [2]. Both use infinitely exchangeable priors over trees. While [2] uses a fragmentation process for trees, our prior uses the reverse—a

coalescent process instead. This allows us to develop simpler inference algorithms than those in [2], though it will be interesting to consider the possibility of developing analogous algorithms for [2]. [3] also describes a hierarchical clustering model involving a prior over trees, but his prior is not infinitely exchangeable. [5] uses tree-consistent partitions to model relational data; it would be interesting to apply our approach to their setting. Another related work is the Bayesian hierarchical clustering of [6], which uses an agglomerative procedure returning a tree structured approximate posterior for a Dirichlet process mixture model. As opposed to our work [6] uses a flat mixture model and does not have a notion of distributions over trees.

There are a number of unresolved issues with our work. Firstly, our algorithms take $O(n^3)$ computation time, except for Greedy-Rate1 which takes $O(n^2)$ time. Among the greedy algorithms we see that there are no discernible differences in quality of approximation thus we recommend Greedy-Rate1. It would be interesting to develop SMC algorithms with $O(n^2)$ runtime. Secondly, there are unanswered statistical questions. For example, since our prior is infinitely exchangeable, by de Finetti's theorem there is an underlying random distribution for which our observations are i.i.d. draws. What is this underlying random distribution, and how do samples from this distribution look like? We know the answer for at least a simple case: if the Markov process is a mutation process with mutation rate $\alpha/2$ and new states are drawn i.i.d. from a base distribution H, then the induced distribution is a Dirichlet process $\mathrm{DP}(\alpha, H)$ [8]. Another issue is that of consistency—does the posterior over random distributions converge to the true distribution as the number of observations grows? Finally, it would be interesting to generalize our approach to varying mutation rates, and to non-binary trees by using generalizations to Kingman's coalescent called Λ-coalescents [16].

References

[1] R. O. Duda and P. E. Hart. *Pattern Classification And Scene Analysis*. Wiley and Sons, New York, 1973.

[2] R. M. Neal. Defining priors for distributions using Dirichlet diffusion trees. Technical Report 0104, Department of Statistics, University of Toronto, 2001.

[3] C. K. I. Williams. A MCMC approach to hierarchical mixture modelling. In *Advances in Neural Information Processing Systems*, volume 12, 2000.

[4] C. Kemp, T. L. Griffiths, S. Stromsten, and J. B. Tenenbaum. Semi-supervised learning with trees. In *Advances in Neural Information Processing Systems*, volume 16, 2004.

[5] D. M. Roy, C. Kemp, V. Mansinghka, and J. B. Tenenbaum. Learning annotated hierarchies from relational data. In *Advances in Neural Information Processing Systems*, volume 19, 2007.

[6] K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the International Conference on Machine Learning*, volume 22, 2005.

[7] N. Friedman. Pcluster: Probabilistic agglomerative clustering of gene expression profiles. Technical Report Technical Report 2003-80, Hebrew University, 2003.

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

[9] J. F. C. Kingman. The coalescent. *Stochastic Processes and their Applications*, 13:235–248, 1982.

[10] A. Doucet, N. de Freitas, and N. J. Gordon. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. New York: Springer-Verlag, May 2001.

[11] P. Fearnhead. *Sequential Monte Carlo Method in Filter Theory*. PhD thesis, Merton College, University of Oxford, 1998.

[12] R. M. Neal. Annealed importance sampling. Technical Report 9805, Department of Statistics, University of Toronto, 1998.

[13] A. McMahon and R. McMahon. *Language Classification by Numbers*. Oxford University Press, 2005.

[14] M. Haspelmath, M. Dryer, D. Gil, and B. Comrie, editors. *The World Atlas of Language Structures*. Oxford University Press, 2005.

[15] H. Daumé III and L. Campbell. A Bayesian model for discovering typological implications. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2007.

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

Bayesian Agglomerative Clustering with Coalescents —Supplemental Material—

Hyperparameter Estimation

In the Brownian motion case, the only hyperparameter in this model is the covariance matrix Λ. For simplicity, we consider only the diagonal case: $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_D)$. We place independent Gamma priors on the inverse variances with hyperparameters a and b. In our experiments we set $a = b = 1.1$ so that the prior has mode at 1. Conditioned on a geneology, the posterior distribution of λ_d^{-1} is again Gamma, with hyperparameters $\widehat{a}_d$ and $\widehat{b}_d$ given by:

$$\widehat{a}_d = a + \frac{1}{2}(n-1); \quad \widehat{b}_d^{-1} = b^{-1} + \frac{1}{2}\sum_{i=1}^{n-1} \frac{(\widehat{y}_{\rho_{li},d} - \widehat{y}_{\rho_{ri},d})^2}{v_{\rho_{li}} + v_{\rho_{ri}} + t_{li} + t_{ri} - 2t_i} \tag{1}$$

The MAP $\lambda_d^{-1} = (\widehat{a_d} - 1)/\widehat{b_d}$.

Next consider the binary vector case. The two hyperparameters q_{d1} and λ_d can be optimized separately for each entry d. Unfortunately there is no closed form solution and we used Newton steps, reparametrizing q_{d1} as $q_{d1} = 1/(1 + \exp(-v_d))$ so that the resulting optimization is unconstrained. The cost function to be maximized is:

$$\mathcal{L}_d(v_d, \lambda_d) = \sum_{i=1}^{n-1} \log\left(1 - e^{\lambda_h(2t_i - t_{li} - t_{ri})}\left(1 - (1-q_{d1})M_{\rho_{li}}^{d0}M_{\rho_{ri}}^{d0} - q_{d1}M_{\rho_{li}}^{d1}M_{\rho_{ri}}^{d1}\right)\right) \tag{2}$$

Updates for the multinomial case can be derived analogously.

Predictive Density

Given a tree and a new individual y' we wish to know: (a) where y' might coalescent and (b) what the density is at y'. To answer (a), assume that y' coalesces with the genealogy at time t, where $t_j > t > t_{j+1}$. The prior probability of this coalescesce is:

$$\exp[-\sum_{i=1}^{j}(n-i+1)\delta_i - (n-j)(t_j - t)] \tag{3}$$

At time t, there are $n - j$ individuals that y' could coalesce with. In the Brownian motion case, y' may merge with sibling ρ_s, and the parent of ρ_s is ρ_p. To perform this merge, we need to create a new parent $\rho_{p'}$ between ρ_s and ρ_p to become the parent of y' and ρ_s. The probability of this change is the probability of $\rho_{p'}$ under ρ_p, times the probability of ρ_s and y' under $\rho_{p'}$, divided by the probability of ρ_s under ρ_p. Marginalizing out $\rho_{p'}$, we obtain:

$$\left[(2\pi(v_0 - t))^D \det\Lambda\right]^{-1/2} \exp\left[-\frac{1}{2}\,||y_0 - y'||_{\Lambda(v_0 - t)} - (n-j+1)(t_s - t)\right] \tag{4}$$

$$v_0 = [(v_{\rho_s} + t_s - t)^{-1} + (v_{\rho_p} + t - t_p)^{-1}]^{-1}; \quad y_0 = v_0[\hat{y}_{\rho_s}/(v_{\rho_s} + t_s - t) + \hat{y}_{\rho_p}/(v_{\rho_p} + t_p - t)]$$

Here, v_0 is the posterior variance and y_0 is the posterior mean; $\hat{y}_{\rho_s}$ and v_{ρ_s} are the messages passed *up* through the tree, while $\hat{y}_{\rho_p}$ and v_{ρ_p} are the messages passed *down* through the tree. The full

predictive density is obtained by summing the product of the prior and Eq (4) over all siblings at all time steps; we draw 10 equally spaced samples between t_j and t_{j+1}. Care must be made to correctly handle the root: we draw 10 equally spaced samples beginning at the minimum t and $t - \max_j \delta_j$; moreover, there are no messages coming down from the root, so those terms are excluded from the likelihood in (4).

Marginal Likelihood Estimation

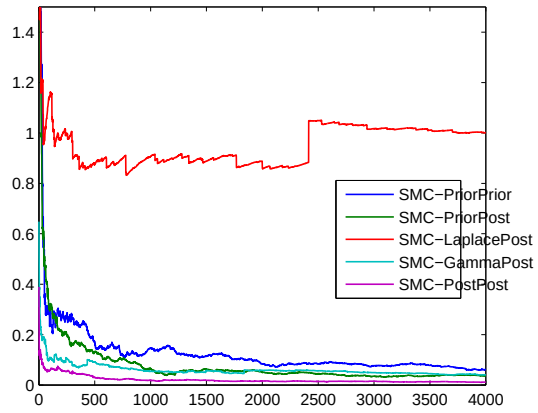


Figure 1: Error in Monte Carlo estimates of the marginal likelihood of a small data set.

In order to evaluate the quality of the proposal distributions, we calculated the exact marginal likelihood under the Brownian diffusion coalescent process on a small tree with data points at $\{-3.1416, 2.1718, 1.618\}$. We then ran the particle filters without resampling to gather 4000 weighted samples, computed the Monte Carlo estimate of the marginal likelihood for $n = 1, \ldots, 4000$, and measured the difference from the true marginal likelihood. Figure 1 shows the results. In summary, as expected, `SMC-PostPost` is the best. Instead of sampling from the coalescent time prior, and as an alternative to sampling from the computational expensive mixture of generalized inverse Gaussian in `SMC-PostPost`, various approximations to the conditional distribution on coalescent times were developed. A gaussian fit failed in this task, suffering from high variance. The gamma fit was superior, but in experience, also suffered from large variance. We believe both of these failed due to tails that are too short (the Gamma assigning too little mass close to zero).

NIPS Coalescent

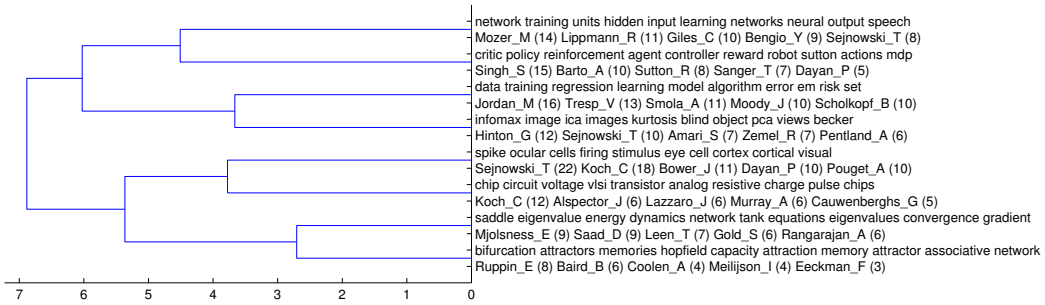


Figure 2: Top of the tree derived from the NIPS abstract data, with most indicative words and most frequent authors for each sub-node.

LLR ***(Time)***	**Top Words and** ***Top Authors***
32.7 *(-2.71)*	bifurcation attractors hopfield network saddle dynamics attractor eigenvalue equilibrium *Mjolsness_E (9) Saad_D (9) Ruppin_E (8) Coolen_A (7) Leen_T (7)*
.106 *(-3.77)*	voltage model cells neurons neuron cell figure spike input time *Koch_C (30) Sejnowski_T (22) Bower_J (11) Dayan_P (10) Pouget_A (10)*
83.8 *(-2.02)*	chip circuit voltage vlsi transistor analog resistive charge pulse chips *Koch_C (12) Alspector_J (6) Lazzaro_J (6) Murray_A (6) Cauwenberghs_G (5)*
140 *(-2.43)*	spike ocular cells firing stimulus eye cell cortex cortical visual *Sejnowski_T (22) Koch_C (18) Bower_J (11) Dayan_P (10) Pouget_A (10)*
2.48 *(-3.66)*	data model learning algorithm training set function latent mixture bayesian *Jordan_M (17) Hinton_G (16) Williams_C (14) Tresp_V (13) Moody_J (12)*
31.3 *(-2.76)*	infomax image ica images kurtosis blind object pca views becker *Hinton_G (12) Sejnowski_T (10) Amari_S (7) Zemel_R (7) Pentland_A (6)*
31.6 *(-2.83)*	data training regression learning model algorithm error em risk set *Jordan_M (16) Tresp_V (13) Smola_A (11) Moody_J (10) Scholkopf_B (10)*
39.5 *(-2.46)*	critic policy reinforcement agent controller reward robot sutton actions mdp *Singh_S (15) Barto_A (10) Sutton_R (8) Sanger_T (7) Dayan_P (5)*
23.0 *(-3.03)*	network training units hidden input learning networks neural output speech *Mozer_M (14) Lippmann_R (11) Giles_C (10) Bengio_Y (9) Sejnowski_T (8)*

Table 1: Nine clusters discovered in NIPS abstracts data.

Phylolinguistics

In the second experiment, we restrict ourselves to languages from the following families: Niger-Congo, Indo-European, Austronesian, Australian, Afro-Asiatic and Sino-Tibetan. We further require that a language have at most 60 of the 139 features unknown—this leaves 64 languages. The coalescent for these languages is shown—together with corresponding language families—in Figure 3. In this figure, we can see that the coalescent is able to identify almost all of Indo-European (with two exceptions: Persian is a bit far away and Hindi/Armenian are also). It does quite well with Austronesian languages, erring only with Paamese. The Australian languages are mixed up a bit with the Sino-Tibetan languages, which can perhaps be accounted for on the basis of areal sharing (i.e., language change due to close proximity).

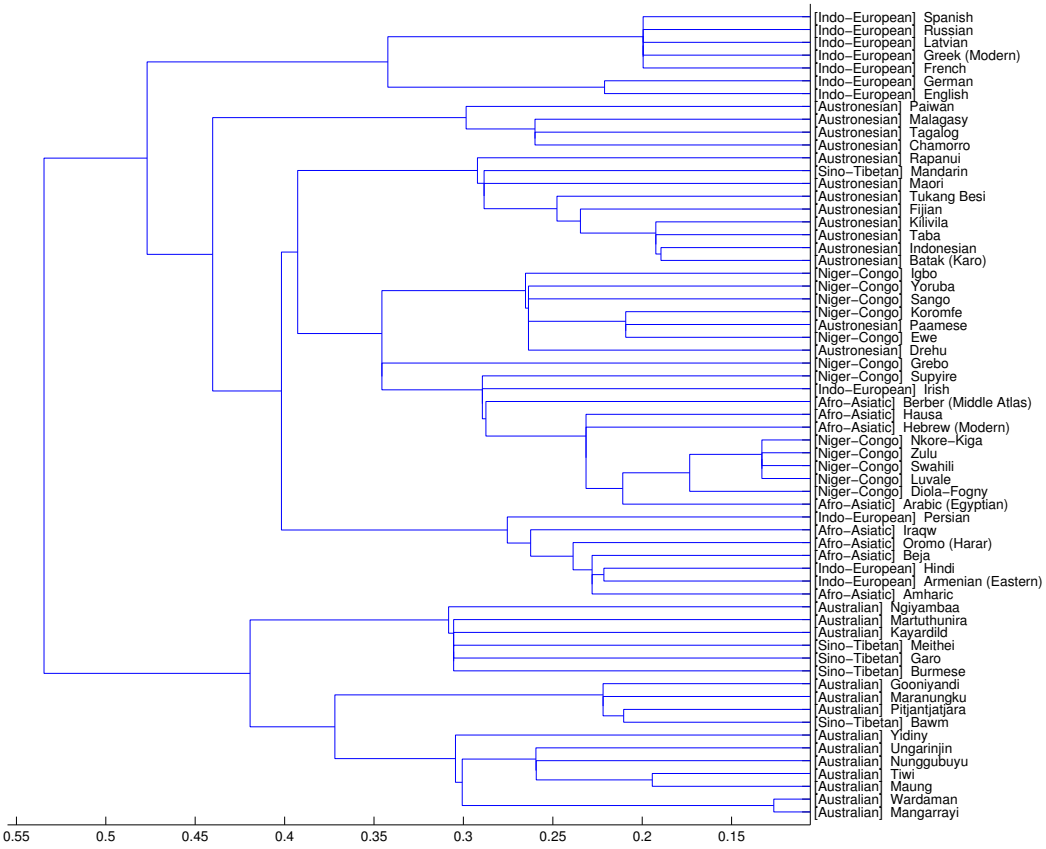


Figure 3: Coalescent for a subset of 64 languages from WALS.