Pitman Yor Diffusion Trees for Bayesian hierarchical clustering

David A. Knowles, *Stanford University* and Zoubin Ghahramani, *University of Cambridge*

Abstract—In this paper we introduce the Pitman Yor Diffusion Tree (PYDT), a Bayesian non-parametric prior over tree structures which generalises the Dirichlet Diffusion Tree [Neal, 2001] and removes the restriction to binary branching structure. The generative process is described and shown to result in an exchangeable distribution over data points. We prove some theoretical properties of the model including showing its construction as the continuum limit of a nested Chinese restaurant process model. We then present two alternative MCMC samplers which allows us to model uncertainty over tree structures, and a computationally efficient greedy Bayesian EM search algorithm. Both algorithms use message passing on the tree structure. The utility of the model and algorithms is demonstrated on synthetic and real world data, both continuous and binary.

Index Terms—Machine learning, unsupervised learning, clustering methods, phylogeny, density estimation, robust algorithm

1 Introduction

Tree structures play an important role in machine learning and statistics. Learning a tree structure over data points gives a straightforward picture of how objects of interest are related. Trees are easily interpreted and intuitive to understand. Sometimes we may know that there is a true hierarchy underlying the data: for example species in the tree of life or duplicates of genes in the human genome, known as paralogs. Typical mixture models, such as Dirichlet Process mixture models, have independent parameters for each component. We might expect for example that certain clusters are similar, being sub-groups of some larger group. By learning this hierarchical similarity structure, the model can share statistical strength between components to make better estimates of parameters using less data.

Classical hierarchical clustering algorithms employ a bottom up "agglomerative" approach [Duda et al., 2001] based on distances which hides the statistical assumptions being made. Heller and Ghahramani [2005] use a principled probabilistic model in lieu of a distance metric but simply view the hierarchy as a tree consistent mixture over partitions of the data. If instead a full generative model for both the tree structure and the data is used [Williams, 2000, Neal, 2003b, Teh et al., 2008, Blei et al., 2010] Bayesian inference machinery can be used to compute posterior distributions over the tree structures themselves.

An advantage of generative probabilistic models for trees is that they can be used as a building block for other latent variable models [Rai and Daumé III, 2008, Adams et al., 2010]. We could use this technique to build topic models with hierarchies on the topics, or hidden Markov models where the states are hierarchically related. Greedy agglomerative approaches can only cluster latent variables *after* inference has been done and hence they cannot be used in a principled way to aid inference in the latent variable model.

Both heuristic and generative probabilistic approaches to learning hierarchies have focused on learning binary trees. Although computationally convenient this restriction may be undesirable: where appropriate, arbitrary trees provide a more interpretable, clean summary of the data. For example, a tree equivalent to a flat clustering can be learnt if this is appropriate for the data, which gives a simpler picture of the similarity between objects than any binary tree. In phylogenetics allowing multifurcation is motivated either by situations where the data is not strong enough to determine the order of binary speciation events, or by specific models of the evolutionary process, such as Hedgecock's sweepstakes [Hedgecock, 1994] where a single or small number of individuals (species) give rise to a disproportionate fraction of the next generation. Some recent work has aimed to address this [Blundell et al., 2010, Adams et al., 2010], which we discuss in Section 3.

The Dirichlet Diffusion Tree (DDT) introduced in Neal [2003b], and reviewed in Section 4, is a simple yet powerful generative model which specifies a distribution on binary trees with multivariate Gaussian distributed variables at the leaves. The DDT is a Bayesian nonparametric prior, and is a generalization of Dirichlet Process mixture models [Antoniak, 1974, Rasmussen, 2000]. The DDT can be thought of as providing a very flexible density model, since the hierarchical structure is able to effectively fit non-Gaussian distributions. Indeed, in Adams et al. [2008] the DDT was shown to significantly outperform a Dirichlet Process mixture model in terms of predictive performance, and in fact slightly outperformed the Gaussian Process Density Sampler. The DDT also formed part of the winning strategy in the NIPS 2003 feature extraction challenge [Guyon et al., 2005]. The DDT is thus both a mathematically elegant nonpara-

metric distribution over hierarchies and provides state-of-the-art density estimation performance.

We introduce the Pitman Yor Diffusion Tree (PYDT), a generalization of the DDT to trees with arbitrary branching structure. While allowing atoms in the divergence function of the DDT can in principle be used to obtain multifurcating branch points [Neal, 2003b], our solution is both more flexible and more mathematically and computationally tractable. An interesting property of the PYDT is that the implied distribution over tree structures corresponds to the multifurcating Gibbs fragmentation tree [McCullagh et al., 2008], a very general process generating exchangeable and consistent trees (here consistency can be understood as coherence under marginalization of subtrees).

This paper is organised as follows. Section 2 formalises the various notions of a "tree" used in this paper. Section 3 briefly describes related work and Section 4 gives background material on the DDT. In Section 5 we describe the generative process corresponding to the PYDT. In Section 6 we derive the probability of a tree and show some important properties of the process. Section 7 describes our hierarchical clustering models utilising the PYDT. In Section 8 we present two alternative MCMC samplers and a greedy Bayesian EM algorithm for the PYDT. We present results demonstrating the utility of the PYDT in Section 9. An earlier version of this paper was presented in Knowles and Ghahramani [2011].

2 Hierarchical partitions, phenograms and diffusion trees

A partition of $[N] := \{1, \ldots, N\}$ is a collection of disjoint, non-empty subsets $\{B_k \subseteq [N] : k = 1, .., K\}$, which we will refer to as "blocks", whose union is $[N]$. The canonical distribution over the space of partitions is the Chinese restaurant process [CRP, Aldous, 1983]. We give the two parameter CRP corresponding to the Pitman Yor process here: the one parameter CRP corresponding to the Dirichlet process is recovered by setting $\alpha = 0$. The CRP is constructed iteratively for $n = 1, 2, ...$ Data point n joins an existing block k with probability

$$\frac{|B_k| - \alpha}{\theta + n - 1} \tag{1}$$

and forms its own new block with probability

$$\frac{\theta + K\alpha}{\theta + n - 1}, \tag{2}$$

where $|B_k|$ is the cardinality of B_k, (θ, α) are the concentration and discount parameter respectively. The canonical parameter range is $\{0 \leq \alpha \leq 1, \theta > -\alpha\}$ but other valid ranges exist.

We can take two closely related views of "tree structures": as hierarchical partitions of $[N]$ or as tree graphs with labelled leaves $[N]$ and a special root node. A hierarchical partition is defined recursively: a hierarchical partition $\mathcal{T}_B$ of a finite non-empty set B is a collection of non-empty subsets of B that a) contains B b) if $|B| \geq 2$ is a union of $\{B\}$ and k "child" hierarchical partitions $\mathcal{T}_{B_i}$ where $\{B_1, \ldots, B_k\}$ is a partition of B. We have $B_i \in \mathcal{T}_B$ for all $i \in [k]$ by this construction. To construct the corresponding graph (V, E), let the set of vertices (nodes) V be the elements of the hierarchical partition, i.e. $V = \mathcal{T}_{[N]}$, and include an edge between node u and v if v is a child of u in the hierarchy, i.e. $E = \{\{u, v\} : u, v \in V; v \subset u; \nexists w \in V \text{ s.t. } v \subset w \subset u\}$. We specify node $[N]$ as the root and the singletons $\{i\}$ for all $i \in [N]$ as "leaves". We refer to both the hierarchical partition and the corresponding graph as $\mathcal{T}_N$, and the space of such objects as $\mathbb{T}_N$. In phylogenetics such objects are referred to as "cladograms". Our construction of hierarchical partitions precludes nodes with a single child (since a set cannot contain duplicate elements), but extending the graph representation to allow such nodes is straightforward. The space of such tree graphs is $\mathbb{T}'_N$.

We can endow each edge $e \in E$ of a tree graph with a weight $w_e \in \mathbb{R}$ known as a branch length. Such trees are refered to as "phenograms" in phylogenetics. We denote the space of phenograms on N leaves by $\mathbb{P}_N$. Finally, we will also be interested in phenograms where each node v is associated with some value $x_v \in \mathcal{X}$ where $\mathcal{X}$ will always be $\mathbb{R}^D$ for some D in this paper. We refer to such objects as diffusion trees and the space of diffusion trees as $\mathbb{F}_N$.

3 Related work

Most hierarchical clustering methods, both distance based [Duda et al., 2001] and probabilistic [Teh et al., 2008, Heller and Ghahramani, 2005], have focused on the case of binary branching structure. In Bayesian hierarchical clustering [Heller and Ghahramani, 2005] the traditional bottom-up agglomerative approach is kept but a principled probabilistic model is used to find subtrees of the hierarchy $\mathcal{T}_N \in \mathbb{T}_N$. Bayesian evidence is then used as the metric to decide which node to incorporate in the tree. An extension where the restriction to binary trees is removed is proposed in Blundell et al. [2010]. They use a greedy agglomerative search algorithm based on various possible ways of merging subtrees. As for Heller and Ghahramani [2005] the lack of a generative process prohibits modelling uncertainty over the space of tree structures $\mathbb{T}_N$.

Non-binary trees are possible in the model proposed in Williams [2000] since each node independently picks a parent in the layer above, but it is necessary to prespecify the number of layers and number of nodes in each layer. Their attempts to learn the number of nodes/layers were in fact detrimental to empirical performance. Unlike the DDT or PYDT, the model in Williams [2000] is parametric in nature, so its complexity cannot automatically adapt to the data.

The nested Chinese restaurant process has been used to define probability distributions over tree structures in $\mathbb{T}'_N$. In Blei et al. [2010] each data point is drawn from

a mixture over the parameters on the path from the root to the data point, which is appropriate for mixed membership models but not standard clustering. It is possible to use the nested CRP for hierarchical clustering, but either a finite number of levels must be pre-specified, some other approach of deciding when to stop fragmenting must be used, or chains of infinite length must be integrated over [Steinhardt and Ghahramani, 2012]. We will show in Section 6.8 that the DDT and PYDT priors on $\mathbb{P}$ can be reconstructed as the continuum limits of particular nested CRP models.

An alternative prior to the PYDT over $\mathbb{T}'_N$ which also allows trees of unbounded depth and width is given by Adams et al. [2010], which is closely related to the nested CRP. They use a nested stick-breaking representation to construct the tree, which is then endowed with a diffusion process. At each node there is a latent probability of the current data point stopping, and so data live at internal nodes of the tree, rather than at leaves as in the PYDT. Despite being computationally appealing, this construction severely limits how much the depth of the tree can adapt to data [Steinhardt and Ghahramani, 2012].

Kingman's coalescent [Kingman, 1982, Teh et al., 2008] is similar to the Dirichlet Diffusion Tree in spirit. Both can be considered as priors on $\mathbb{P}_N$. For Kingman's coalescent (KC) the generative process is defined going backwards in time as datapoints coalesce together, rather than forward in time as for the DDT. KC is the dual process to the Dirichlet diffusion tree, in the following sense. Imagine we sample a partition of $[n]$ from the Chinese restaurant process with concentration parameter θ, coalesce this partition for a small time dt, and then "fragment" the resulting partition according to the DDT with constant rate function for time dt. The final partition will be CRP distributed with concentration parameter θ, showing that the DDT fragmentation has "undone" the effect of the coalescent process. This duality is used in Teh et al. [2011] to define a partition valued stochastic process through time. A prior on $\mathbb{T}_N$ can be derived from KC by marginalising over the possible orderings of the coalescent events [Boyles and Welling, 2012]. The generalisation of KC to arbitrary branching structures has been studied in the probability literature under the name Λ-coalescent [Pitman, 1999, Sagitov, 1999]. While Steinrücken et al. [2012], Eldon and Wakeley [2006] used summary statistics to fit parameters of specific Λ-coalescent models, we are unaware of attempts to directly infer the phylogeny itself under this prior.

4 The Dirichlet Diffusion Tree

The Dirichlet Diffusion Tree was introduced in Neal [2003b] as a top-down generative model for trees in $\mathbb{F}_N$ over N datapoints $x_1, x_2, \cdots, x_N \in \mathbb{R}^D$. We will describe the generative process for the data in terms of a diffusion process in fictitious "time" on the unit interval. The observed data points (or latent variables) correspond to the locations of the diffusion process at time $t = 1$. The first datapoint starts at time 0 at the origin in a D-dimensional Euclidean space and follows a Brownian motion with variance σ^2 until time 1. If datapoint 1 is at position $x_1(t)$ at time t, the point will reach position $x_1(t+dt) \sim \mathrm{N}(x_1(t), \sigma^2 I dt)$ at time $t+dt$. It can easily be shown that $x_1(t) \sim \mathrm{Normal}(0, \sigma^2 I t)$. The second point x_2 in the dataset also starts at the origin and initially follows the path of x_1. The path of x_2 will diverge from that of x_1 at some time T_d after which x_2 follows a Brownian motion independent of $x_1(t)$ until $t = 1$. In other words, the infinitesimal increments for the second path are equal to the infinitesimal increments for the first path for all $t < T_d$. After T_d, the increments for the second path $\mathrm{N}(0, \sigma^2 I dt)$ are independent. The probability of diverging in an interval $[t, t+dt]$ is determined by a "divergence function" $a(t)$ (see Equation 10 below) which is analogous to the hazard function in survival analysis.

The generative process for datapoint i is as follows. Initially $x_i(t)$ follows the path of the previous datapoints. If at time t the path of $x_i(t)$ has not diverged, it will diverge in the next infinitesimal time interval $[t, t+dt]$ with probability

$$\frac{a(t)dt}{m} \tag{3}$$

where m is the number of datapoints that have previously followed the current path. The division by m is a reinforcing aspect of the DDT: the more datapoints follow a particular branch, the more likely subsequent datapoints will not diverge off this branch (this division is also required to ensure exchangeability). If x_i does not diverge before reaching a previous branching point, the previous branches are chosen with probability proportional to how many times each branch has been followed before. This reinforcement scheme is similar to the Chinese restaurant process. For the single data point $x_i(t)$ this process is iterated down the tree until divergence, after which $x_i(t)$ performs independent Brownian motion until time $t = 1$. The i-th observed data point is given by the location of this Brownian motion at $t = 1$, i.e. $x_i(1)$.

For the purpose of this paper we use the divergence function $a(t) = \frac{c}{1-t}$, with "smoothness" parameter $c > 0$. Larger values of c give smoother densities because divergences typically occur earlier, resulting in less dependence between the datapoints. Smaller values of c give rougher more "clumpy" densities with more local structure since divergence typically occurs later, closer to $t = 1$. We refer to Neal [2001] for further discussion of the properties of this and other divergence functions. Figure 1 illustrates the Dirichlet diffusion tree process for a dataset with $N = 4$ datapoints.

The probability of generating the tree, latent variables and observed data under the DDT can be decomposed into two components. The first component specifies the distribution over the tree structure and the divergence

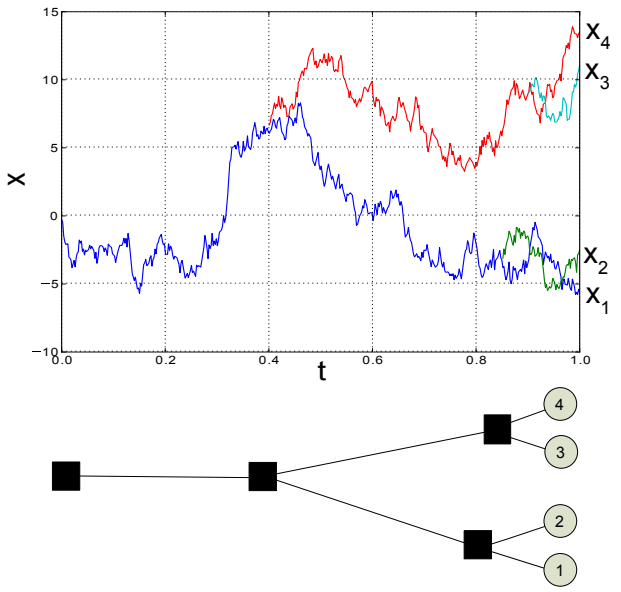


Fig. 1: A sample from the Dirichlet Diffusion Tree with $N = 4$ datapoints. Top: the location of the Brownian motion for each of the four paths. Bottom: the corresponding tree structure. Each branch point corresponds to an internal tree node.

times, $\mathcal{P}_N \in \mathbb{P}_N$. The second component specifies the distribution over the specific locations of the Brownian motion when the tree structure and divergence times are given.

Before we describe the functional form of the DDT prior we will need two results. First, the probability that a new path does not diverge between times $s < t$ on a segment that has been followed m times by previous data-points can be written as

$$P(\text{not diverging}) = \exp\left[(A(s) - A(t))/m\right], \tag{4}$$

where $A(t) = \int_0^t a(u)du$ is the cumulative rate function. For our divergence function $A(t) = -c\log(1-t)$. Second, the DDT prior defines an exchangeable distribution: the order in which the datapoints were generated does not change the joint density. See Neal [2003b] for a proof.

We now consider the tree as a set of segments $\mathcal{S}(\mathcal{T})$ each contributing to the joint probability density. The tree structure $\mathcal{T}_n \in \mathbb{T}_N$ encodes the counts of how many datapoints traversed each segment. Consider an arbitrary segment $[uv] \in \mathcal{S}(\mathcal{T})$ from node u to node v with corresponding locations x_u and x_v and divergence times t_u and t_v, where $t_u < t_v$. Let $m(v)$ be the number of leaves under node v, i.e. the number of datapoints which traversed segment $[uv]$. Let $l(v)$ and $r(v)$ be the number of leaves under the left and right child of node v respectively, so that $l(v) + r(v) = m(v)$.

By exchangeability we can assume that it was the second path which diverged at v. None of the subsequent paths that passed through u diverged before time t_v (otherwise $[uv]$ would not be a contiguous segment). The probability, $P(t_v|[uv], t_u)$ of this happening is

$$\overbrace{\frac{a(t_v)}{1}}^{\text{2nd branch diverges}} \prod_{i=1}^{m(v)-1} \overbrace{\exp[(A(t_u) - A(t_v))/i]}^{\text{(i+1)th branch does not diverge before v}}$$
$$= a(t_v)\exp\left[(A(t_u) - A(t_v))H_{m(v)-1}\right], \tag{5}$$

where $H_n = \sum_{i=1}^{n} 1/i$ is the nth harmonic number. This expression factorizes into a term for t_u and t_v. Collecting such terms from the branches attached to an internal node i the factor for t_i for the divergence function $a(t) = c/(1-t)$ is

$$a(t_i)e^{[A(t_i)(H_{l(i)-1} + H_{r(i)-1} - H_{m(i)-1})]}$$
$$= c(1-t_i)^{cJ_{l(i),r(i)}-1}, \tag{6}$$

where $J_{l,r} = H_{r+l-1} - H_{l-1} - H_{r-1}$.

Each path that went through x_v, except the first and second, had to choose to follow the left or right branch. Again, by exchangeability, we can assume that all $l(v)-1$ paths took the left branch first, then all $r(v) - 1$ paths chose the right branch. The probability of this happening is

$$P([uv]) = \frac{(l(v)-1)!(r(v)-1)!}{(m(v)-1)!}. \tag{7}$$

Finally, we include a term for the diffusion locations:

$$P(x_v|x_u, t_u, t_v) = \mathrm{N}(x_v; x_u, \sigma^2(t_v - t_u)). \tag{8}$$

The full joint probability for the DDT is now a product of terms for each segment

$$P(x, t, \mathcal{T}) = \prod_{[uv]\in\mathcal{S}(\mathcal{T})} P(x_v|x_u, t_u, t_v)P(t_v|[uv], t_u)P([uv]). \tag{9}$$

5 Generative process for the PYDT

The PYDT generative process is analogous to that for the DDT, but altered to allow arbitrary branching structures. Firstly, the probability of diverging from a branch having previously been traversed by m data points in interval $[t, t+dt]$ is given by

$$\frac{a(t)\Gamma(m-\alpha)dt}{\Gamma(m+1+\theta)} \tag{10}$$

where $\Gamma(.)$ is the standard Gamma function, θ is the concentration parameter and α is the discount parameter by analogy to the Pitman Yor process (see Section 6.2 for discussion of allowable parameter ranges). When $\theta = \alpha = 0$ we recover binary branching and the DDT expression in Equation 3. Secondly, if x_i does not diverge before reaching a previous branching point, it may either follow one of the previous branches, or diverge at the branch point (adding one to the degree of this node in the tree). The probability of following one of the existing branches k is

$$\frac{n_k - \alpha}{m + \theta} \tag{11}$$

where n_k is the number of samples which previously took branch k and m is the total number of samples through this branch point so far. The probability of diverging at the branch point and creating a new branch is

$$\frac{\theta + \alpha K}{m + \theta} \tag{12}$$

where K is the current number of branches from this branch point. By summing Equation 11 over $k = \{1, \ldots, K\}$ with Equation 12 we get 1, since $\sum_k n_k = m$, as required. This reinforcement scheme is analogous to the Pitman Yor process [Teh, 2006, Pitman and Yor, 1997] version of the Chinese restaurant process [Aldous, 1983].

5.1 Sampling the PYDT in practice

It is straightforward to sample from the PYDT prior. This is most easily done by sampling the tree structure and divergence times first, followed by the divergence locations. We will need the inverse cumulative divergence function, e.g. $A^{-1}(y) = 1.0 - \exp(-y/c)$ for the divergence function $a(t) = c/(1-t)$.

Each point starts at the root of the tree. The cumulative distribution function for the divergence time of the i-th sample is

$$C(t) = 1 - \exp\left\{-A(t)\frac{\Gamma(i-1-\alpha)}{\Gamma(i+\theta)}\right\} \tag{13}$$

We can sample from this distribution by drawing $U \sim \text{Uniform}[0,1]$ and setting

$$t_d = C^{-1}(U) := A^{-1}\left(-\frac{\Gamma(i+\theta)}{\Gamma(i-1-\alpha)}\log(1-U)\right) \tag{14}$$

If t_d is actually past the next branch point, we diverge at this branch point or choose one of the previous paths with the probabilities defined in Equations 12 and 11 respectively. If we choose one of the existing branches then we must again sample a divergence time. On an edge from node u to v previously traversed by $m(v)$ data points, the cumulative distribution function for a new divergence time is

$$C(t) = 1 - \exp\left\{-[A(t) - A(t_u)]\frac{\Gamma(m(v)-\alpha)}{\Gamma(m(v)+1+\theta)}\right\} \tag{15}$$

which we can sample as follows

$$t_d := A^{-1}\left(A(t_u) - \frac{\Gamma(m(v)+1+\theta)}{\Gamma(m(v)-\alpha)}\log(1-U)\right) \tag{16}$$

We do not actually need to be able to evaluate $A(t_u)$ since this will necessarily have been calculated when sampling t_u. If $t_d > t_v$ we again choose whether to follow an existing branch or diverge according to Equations 12 and 11.

Given the tree structure and divergence times sampling the locations simply involves a sweep down the tree sampling $x_v \sim N(x_u, \sigma^2(t_v - t_u)I)$ for each branch $[uv]$.

6 THEORY

Now we present some important properties of the PYDT generative process.

6.1 Probability of a tree

The probability of generating a specific tree structure with associated divergence times and locations at each node can be written analytically since the specific diffusion path taken between nodes can be ignored. We will need the probability that a new data point does not diverge between times $s < t$ on a branch that has been followed m times by previous data-points. This can straightforwardly be derived from Equation 10:

$$P\left(\begin{array}{c}\text{not diverging}\\ \text{in } [s,t]\end{array}\right) = \exp\left[(A(s) - A(t))\frac{\Gamma(m-\alpha)}{\Gamma(m+1+\theta)}\right], \tag{17}$$

where $A(t) = \int_0^t a(u)du$ is the cumulative rate function.

Consider the tree of $N = 4$ data points in Figure 2. The probability of obtaining this tree structure and associated divergence times is:

$$\begin{aligned}
&e^{-A(t_u)\frac{\Gamma(1-\alpha)}{\Gamma(2+\theta)}}\frac{a(t_u)\Gamma(1-\alpha)}{\Gamma(2+\theta)}\\
&\times e^{-A(t_u)\frac{\Gamma(2-\alpha)}{\Gamma(3+\theta)}}\frac{1-\alpha}{2+\theta}e^{[A(t_u)-A(t_v)]\frac{\Gamma(1-\alpha)}{\Gamma(2+\theta)}}\frac{a(t_v)\Gamma(1-\alpha)}{\Gamma(2+\theta)}\\
&\times e^{-A(t_u)\frac{\Gamma(3-\alpha)}{\Gamma(4+\theta)}}\frac{\theta+2\alpha}{3+\theta}.
\end{aligned} \tag{18}$$

The first data point does not contribute to the expression. The second point contributes the first line: the first term results from not diverging between $t = 0$ and t_u, the second from diverging at t_u. The third point contributes the second line: the first term comes from not diverging before time t_u, the second from choosing the branch leading towards the first point, the third term comes from not diverging between times t_u and t_v, and the final term from diverging at time t_v. The fourth and final data point contributes the final line: the first term for not diverging before time t_u and the second term for diverging at branch point u.

Although not immediately obvious, we will see in Section 6.3, the tree probability in Equation 18 is invariant to reordering of the data points.

The component of the joint probability distribution resulting from the branching point and data locations for the tree in Figure 2 is

$$\begin{aligned}
&N(x_u; 0, \sigma^2 t_u)N(x_v; x_u, \sigma^2(t_v - t_u))\\
&\times N(x_1; x_v, \sigma^2(1-t_v))N(x_2; x_u, \sigma^2(1-t_u))\\
&\times N(x_3; x_v, \sigma^2(1-t_v))N(x_4; x_u, \sigma^2(1-t_u))
\end{aligned} \tag{19}$$

where we see there is a Gaussian term associated with each branch in the tree.

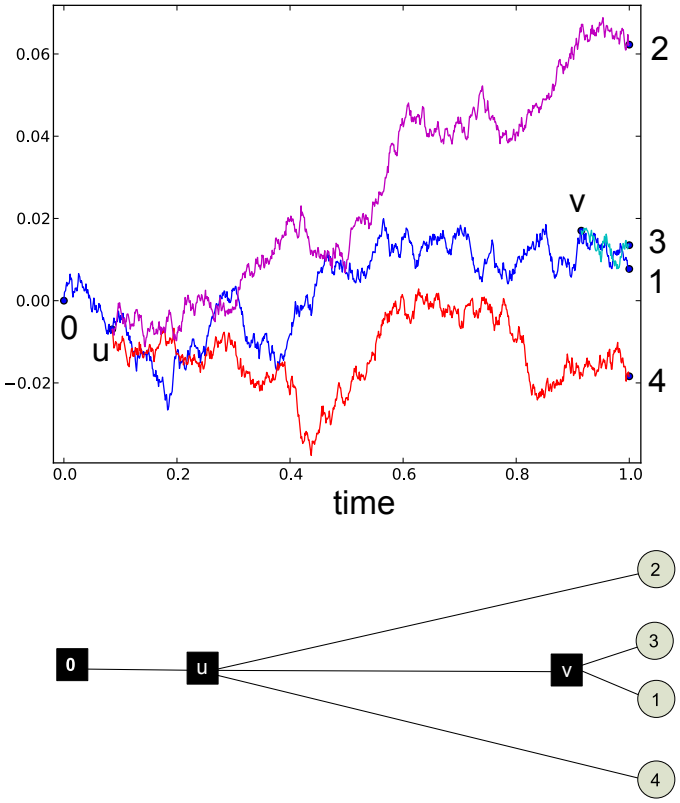


Fig. 2: A sample from the Pitman-Yor Diffusion Tree with $N = 4$ datapoints and $a(t) = 1/(1-t), \theta = 1, \alpha = 0$. Top: the location of the Brownian motion for each of the four paths. Bottom: the corresponding tree structure. Each branch point corresponds to an internal tree node.

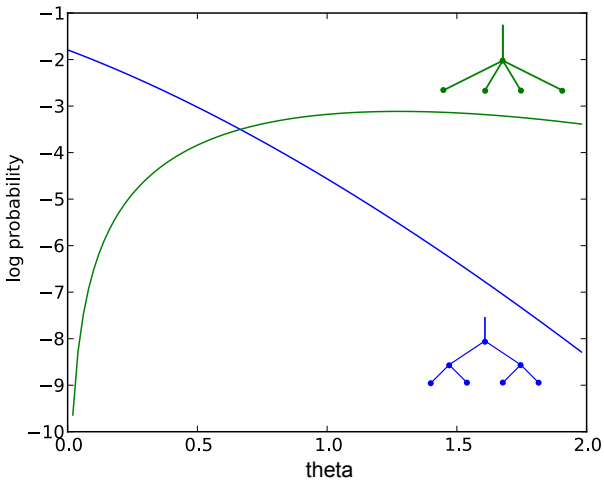


Fig. 3: The effect of varying θ on the log probability of two tree structures (i.e. the product of the terms in Equation 22 over the segments in the tree), indicating the types of tree preferred. Small $\theta < 1$ favours binary trees while larger values of θ favors higher order branching points.

6.2 Parameter ranges and branching degree

McCullagh et al. [2008] calculated the valid parameter ranges for multifurcating Gibbs fragmentation trees on $\mathbb{T}_N$, which correspond to the PYDT after marginalising over the divergence times (see Section 6.6). Following their result, there are several valid ranges of the parameters (θ, α) :

- $0 \leq \alpha < 1$ and $\theta > -2\alpha$. This is the general multifurcating case with arbitrary branching degree which we will be most interested in (although in fact we will often restrict further to $\theta > 0$). $\alpha < 1$ ensures the probability of going down an existing branch is non-negative in Equation 11. $\theta > -2\alpha$ and $\alpha \geq 0$ together ensure that the probability of forming a new branch is non-negative for any K in Equation 12.
- $\alpha < 0$ and $\theta = -\kappa\alpha$ where $\kappa \in \mathbb{Z}$ and $\kappa \geq 3$. Here κ is the maximum number of children a node can have since the probability of forming a new branch at a node with $K = \kappa$ existing branches given by Equation 12 will be zero. We require $\alpha < 0$ to ensure the probability of following an existing branch is always positive.
- $\alpha < 1$ and $\theta = -2\alpha$. This gives binary branching, and specifically the DDT for $\alpha = \theta = 0$. Interestingly however we see that this gives a parameterised family of priors over binary trees, which was in fact proposed by MacKay and Broderick [2007].

There are two other degenerate cases which are of little interest for statistical modeling. The first is $\alpha = 1$ and the second is $\alpha = -\infty$ and $\theta \in \{2, 3, \dots\}$. In both cases we have instantaneous divergence at time $t = 0$ (since the numerator in Equation 10 contains the term $\Gamma(m - \alpha)$) so every data point is independent. The first case, $\alpha = 1$ corresponds to a deterministic split into singleton blocks. The second case, $\alpha = -\infty$ and $\theta \in \{2, 3, \dots\}$, actually gives a non-degenerate distribution over cladograms corresponding to a recursive "coupon collector problem" conditioned such that at least two coupons are collected at each split.

Consider the parameter range $0 \leq \alpha < 1$ and $\theta > -2\alpha$. By varying θ we can move between flat (large θ) and "bushy" clusterings (small θ), as shown in Figure 3 (here we have fixed $\alpha = 0$).

6.3 Exchangeability

Exchangeability is both a key modelling assumption and a property that greatly simplifies inference. We show that analogously to the DDT, the PYDT defines an infinitely exchangeable distribution over the data points. We first need the following lemma.

Lemma 1: The probability of generating a specific tree structure, divergence times, divergence locations and corresponding data set is invariant to the ordering of data points.

Proof: The probability of a draw from the PYDT can be decomposed into three components: the probability

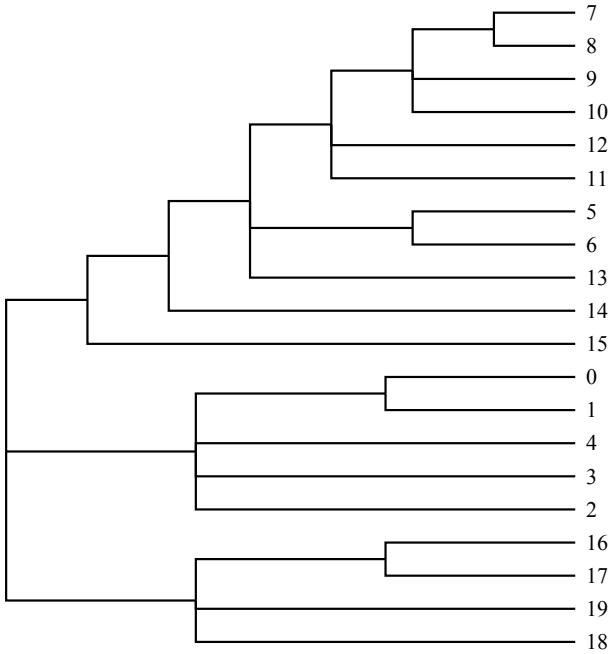
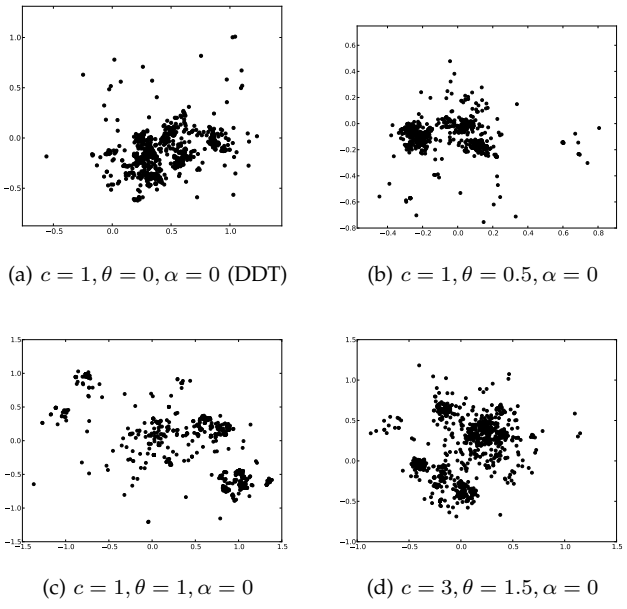


Fig. 4: A sample from the Pitman-Yor Diffusion Tree with $N = 20$ datapoints and $a(t) = 1/(1-t), \theta = 1, \alpha = 0$ showing the branching structure including non-binary branch points.



(a) $c = 1, \theta = 0, \alpha = 0$ (DDT) (b) $c = 1, \theta = 0.5, \alpha = 0$

(c) $c = 1, \theta = 1, \alpha = 0$ (d) $c = 3, \theta = 1.5, \alpha = 0$

Fig. 5: Samples from the Pitman-Yor Diffusion Tree with $N = 1000$ datapoints in $D = 2$ dimensions and $a(t) = c/(1-t)$. As θ increases more obvious clusters appear.

of the underlying tree structure, the probability of the divergence times given the tree structure, and the probability of the divergence locations given the divergence times. We will show that none of these components depend on the ordering of the data. Consider the tree, $\mathcal{T}$ as a set of edges, $\mathcal{S}(\mathcal{T})$ each of which we will see contributes to the joint probability density. The tree structure $\mathcal{T}$ contains the counts of how many datapoints traversed each edge. We denote an edge by $[uv] \in \mathcal{S}(\mathcal{T})$, which goes from node u to node v with corresponding locations x_u and x_v and divergence times t_u and t_v. Let the final number of branches from v be K_v, and the number of samples which followed each branch be $\{n_k^v : k \in [1 \dots K_v]\}$. The total number of datapoints which traversed edge $[uv]$ is $m(v) = \sum_{j=1}^{K_v} n_k^v$. Denote by $\mathcal{S}'(\mathcal{T}) = \{[uv] \in \mathcal{S}(\mathcal{T}) : m(v) \geq 2\}$ the set of all edges traversed by $m \geq 2$ samples (for divergence functions which ensure divergence before time 1 this is the set of all edges not connecting to leaf nodes).

Probability of the tree structure. For segment $[uv]$, let i be the index of the sample which diverged to create the branch point at v. The first $i-1$ samples did not diverge at v so only contribute terms for not diverging (see Equation 23 below). From Equation 10, the probability of the i-th sample having diverged at time t_v to form the branch point (conditional on not diverging before t_v) is

$$\frac{a(t_v)\Gamma(i-1-\alpha)}{\Gamma(i+\theta)}. \tag{20}$$

We now wish to calculate the probability of final configuration of the branch point. Following the divergence of sample i there are $K_v - 2$ samples that form new branches from the same point, which from Equation 12 we see contribute $\theta + (k-1)\alpha$ to the numerator for $k \in \{3, \dots, K_v\}$. Let c_l be the number of samples having previously followed path l, so that c_l ranges from 1 to $n_l^v - 1$, which by Equation 11 contributes a term $\prod_{c_l=1}^{n_l^v-1}(c_l - \alpha)$ to the numerator for $l = 2, ..., K_v$. c_1 only ranges from $i-1$ to $n_1^v - 1$, thereby contributing a term $\prod_{c_1=i-1}^{n_1^v-1}(c_1 - \alpha)$. The j-th sample contributes a factor $j - 1 + \theta$ to the denominator, regardless of whether it followed an existing branch or created a new one, since the denominator in Equations 12 and 11 are equal. The factor associated with this branch point is then:

$$\begin{aligned}
&\frac{\prod_{k=3}^{K_v}[\theta + (k-1)\alpha] \prod_{c_1=i-1}^{n_1^v-1}(c_1 - \alpha) \prod_{l=2}^{K_v} \prod_{c_l=1}^{n_l^v-1}(c_l - \alpha)}{\prod_{j=i+1}^{m(v)}(j - 1 + \theta)} \\
&= \frac{\prod_{k=3}^{K_v}[\theta + (k-1)\alpha] \prod_{l=1}^{K_v} \prod_{c_l=1}^{n_l^v-1}(c_l - \alpha)}{\prod_{j=i+1}^{m(v)}(j - 1 + \theta) \prod_{c_1=1}^{i-2}(c_1 - \alpha)} \\
&= \frac{\prod_{k=3}^{K_v}[\theta + (k-1)\alpha]\Gamma(i+\theta) \prod_{l=1}^{K_v} \Gamma(n_l^v - \alpha)}{\Gamma(m(v)+\theta)\Gamma(i-1-\alpha)\Gamma(1-\alpha)^{K_v-1}}.
\end{aligned} \tag{21}$$

Multiplying by the contribution from data point i in Equation 20 we have

$$\frac{a(t_v)\prod_{k=3}^{K_v}[\theta+(k-1)\alpha]\prod_{l=1}^{K_v}\Gamma(n_l^v-\alpha)}{\Gamma(m(v)+\theta)\Gamma(1-\alpha)^{K_v-1}}. \quad (22)$$

Each segment $[uv] \in \mathcal{S}'(\mathcal{T})$ contributes such a term. Since this expression does not depend on the ordering of the branching events (that is, on the index i), the overall factor does not either. Since $a(t_v)$ is a multiplicative factor we can think of this as part of the probability factor for the divergence times.

Probability of divergence times. The $m(v)-1$ points that followed the first point along this path did not diverge before time t_v (otherwise $[uv]$ would not be an edge), which from Equation 17 we see contributes a factor

$$\prod_{i=1}^{m(v)-1}\exp\left[(A(t_u)-A(t_v))\frac{\Gamma(i-\alpha)}{\Gamma(i+1+\theta)}\right] = \exp\left[(A(t_u)-A(t_v))H^{\theta,\alpha}_{m(v)-1}\right], \quad (23)$$

where we define $H^{\theta,\alpha}_n = \sum_{i=1}^{n}\frac{\Gamma(i-\alpha)}{\Gamma(i+1+\theta)}$. All edges $[uv] \in \mathcal{S}'(\mathcal{T})$ contribute the expression in Equation 23, resulting in a total contribution

$$\prod_{[uv]\in\mathcal{S}'(\mathcal{T})}\exp\left[(A(t_u)-A(t_v))H^{\theta,\alpha}_{m(v)-1}\right]. \quad (24)$$

This expression does not depend on the ordering of the datapoints.

Probability of node locations. Generalizing Equation 19 it is clear that each edge contributes a Gaussian factor, resulting an overall factor:

$$\prod_{[uv]\in\mathcal{S}(\mathcal{T})}\mathrm{N}(x_v;x_u,\sigma^2(t_v-t_u)I). \quad (25)$$

The overall probability of a specific tree, divergence times and node locations is given by the product of Equations 22, 24 and 25, none of which depend on the ordering of the data. □

The term $\prod_{k=3}^{K_v}[\theta+(k-1)\alpha]$ in Equation 22 can be calculated efficiently depending on the value of α. For $\alpha = 0$ we have $\prod_{k=3}^{K_v}\theta = \theta^{K_v-2}$. For $\alpha \neq 0$ we have

$$\prod_{k=3}^{K_v}[\theta+(k-1)\alpha] = \alpha^{K_v-2}\prod_{k=3}^{K_v}[\theta/\alpha+(k-1)] = \frac{\alpha^{K_v-2}\Gamma(\theta/\alpha+K_v)}{\Gamma(\theta/\alpha+2)}. \quad (26)$$

The factor for the divergence times in Equation 24 itself factorizes into a term for t_u and t_v. Collecting such terms from the branches attached to an internal node v the factor for t_v for the divergence function $a(t) = c/(1-t)$ is

$$P(t_v|\mathcal{T}) = a(t_v)\exp\left[A(t_v)\left(\sum_{k=1}^{K_v}H^{\theta,\alpha}_{n_k^v-1}-H^{\theta,\alpha}_{m(v)-1}\right)\right] = c(1-t_v)^{cJ^{\theta,\alpha}_{\mathbf{n}^v}-1} \quad (27)$$

where $J^{\theta,\alpha}_{n^v} = H^{\theta,\alpha}_{\sum_{k=1}^{K}n_k^v-1} - \sum_{k=1}^{K}H^{\theta,\alpha}_{n_k^v-1}$ with $\mathbf{n} \in \mathbb{N}^K$ being the number of datapoints having gone down each branch. Equation 27 is the generalisation of Equation 6 for the DDT to the PYDT. A priori the divergence times are independent apart from the constraint that branch lengths must be non-negative.

Theorem 1: The Pitman-Yor Diffusion Tree defines an infinitely exchangeable distribution over data points.

Proof: Summing over all possible tree structures, and integrating over all branch point times and locations, by Lemma 1 we have exchangeability for any finite number of datapoints, N. As a virtue of its sequential generative process, the PYDT is clearly projective (i.e. the model for $N-1$ datapoints is given by marginalising out the N-th datapoint from the model with N datapoints). Being exchangeable and projective, the PYDT is infinitely exchangeable. □

Corollary 1: There exists a prior ν on probability measures on $\mathbb{R}^D$ such that the samples $x_1, x_2, \ldots$ generated by a PYDT are conditionally independent and identically distributed (iid) according to $\mathcal{F} \sim \nu$, that is, we can represent the PYDT as

$$PYDT(x_1, x_2, \ldots) = \int\left(\prod_i \mathcal{F}(x_i)\right)d\nu(\mathcal{F}).$$

Proof: Since the PYDT defines an infinitely exchangeable process on data points, the result follows directly by de Finetti's Theorem [Hewitt and Savage, 1955]. □

Another way of expressing Corollary 1 is that data points $x_1, \ldots, x_N$ sampled from the PYDT could equivalently have been sampled by first sampling a probability measure $\mathcal{F} \sim \nu$, then sampling $x_i \sim \mathcal{F}$ iid for all i in $\{1, \ldots, N\}$. For divergence functions such that $A(1)$ is infinite, divergence will necessarily occur before time $t = 1$, so that there is zero probability of two data points having the same location, i.e. the probability measure $\mathcal{F}$ is continuous almost surely. Characterising when $\mathcal{F}$ is absolutely continuous (the condition required for a density to exist) remains an open question.

6.4 Relationship to the DDT

The PYDT is a generalisation of the Dirichlet diffusion tree:

Lemma 2: The PYDT reduces to the Dirichlet diffusion tree [Neal, 2001] in the case $\theta = \alpha = 0$.

Proof: This is clear from the generative process: for $\theta = \alpha = 0$ there is zero probability of branching at a previous branch point (assuming continuous cumulative divergence function $A(t)$). The probability of diverging in the time interval $[t, t+dt]$ from a branch previously traversed by m datapoints becomes:

$$\frac{a(t)\Gamma(m-0)dt}{\Gamma(m+1+0)} = \frac{a(t)(m-1)!dt}{m!} = \frac{a(t)dt}{m}, \quad (28)$$

as for the DDT. □

It is straightforward to confirm that the DDT probability factors are recovered when $\theta = \alpha = 0$. In this

case $K_v = 2$ since non-binary branch points have zero probability, so Equation 22 reduces as follows:

$$\frac{a(t_v)\prod_{l=1}^{K_v=2}\Gamma(n_l^v-0)}{\Gamma(m(v)+0)} = \frac{a(t_v)(n_1^b-1)!(n_2^b-1)!}{(m(v)-1)!}, \quad (29)$$

as for the DDT. Equation 24 also reduces to the DDT expression since

$$H_n^{0,0} = \sum_{i=1}^{n}\frac{\Gamma(i-0)}{\Gamma(i+1+0)} = \sum_{i=1}^{n}\frac{(i-1)!}{i!} = \sum_{i=1}^{n}\frac{1}{i} = H_n, \quad (30)$$

where H_n is the n-th Harmonic number.

6.5 Prior over $\mathbb{T}$ is invariant to $a(t)$

The following lemma shows that the prior over tree structures $\mathbb{T}_N$ resulting from sampling a PYDT and discarding the divergence times is invariant to the choice of divergence function $a(t)$, assuming that $a(t)$ is non-atomic.

Lemma 3: Let $a : [0,1) \to \mathbb{R}^+$ be a finite non-atomic divergence function such that $A(1) = \infty$ where $A(t) := \int_0^t a(u)du$. Thus $A^{-1} : \mathbb{R}^+ \to [0,1)$ exists. The PYDT P_1 in $\mathbb{P}_N$ with divergence function $a(t)$, and parameters (θ, α) is equal in distribution to the PYDT P_2 with constant divergence function $a'(t) = 1$ on $\mathbb{R}^+$ and parameters (θ, α), after remapping the divergence times according to A^{-1}.

Intuitively Lemma 3 says that one way to sample a PYDT with divergence function $a(t)$ is to first sample the tree structure and divergence times from a PYDT with constant $a'(t) = 1$, and then remap each the divergence time t_v to $A^{-1}(t_v)$. Finally the Brownian diffusion process can be run on the resulting tree.

Proof: We will show the generative process for a single data point is equal for both processes, so that the result follows by induction. Equality for the first data point requires only that $A^{-1}(0) = 0$ and $A^{-1}(\infty) = 1$. Consider the generative process for data point i starting at the root in either tree, where we assume the trees generated by the two processes up to data point $i-1$ are equal. The choice of which branch to follow (or whether to form a new branch) at an existing branch point is the same for both processes since this does not depend on $a(t)$. The time until divergence on a segment $[uv]$ can be viewed as the waiting time until the first atom of a Poisson process on $[uv]$ with intensity function $a(t)\frac{\Gamma(m-\alpha)}{\Gamma(m+1+\theta)}$ where m is the number of previous data points having traversed $[uv]$. If the Poisson process has no atoms on $[uv]$ then i does not diverge and continues to the next branch point. The Poisson process on $[uv]$ in P_2 has constant intensity function $\frac{\Gamma(m-\alpha)}{\Gamma(m+1+\theta)}$ and therefore rate measure $A_2([a,b]) = (b-a)\frac{\Gamma(m-\alpha)}{\Gamma(m+1+\theta))}$ for $t'_u \le a \le b \le t'_v$. The rate measure A_1 on $[uv]$ in P_1 as a result of the mapping A^{-1} can be calculated using the Poisson process mapping theorem (see for example Kingman [1993]) as

$$\begin{aligned} A_1([a,b]) &= A_2(A([a,b])) = A_2([A(a),A(b)]) \\ &= (A(b)-A(a))\frac{\Gamma(m-\alpha)}{\Gamma(m+1+\theta))} \end{aligned} \quad (31)$$

This is equal to the rate measure when the divergence function is $a(t)$, so the probability of diverging on $[uv]$ is equal in both processes, and if divergence occurs the distribution over when is also equal. □

Lemma 3 makes it possible to derive the distribution over the tree structure $\mathcal{T}_N \in \mathbb{T}_N$ by marginalising over the divergence times. Under the PYDT with constant divergence function $a(t) = 1$ on $\mathbb{R}^+$ the probability factor for the divergence times in Equation 23 simplifies to

$$\exp\left[-(t_v - t_u)H^{\theta,\alpha}_{m(v)-1}\right], \quad (32)$$

We apply a change of variables to use the branch lengths $b_{[uv]} := t_v - t_u$. This reparameterisation has Jacobian 1 and the $b_{[uv]}$ are conditionally independent given the counts $m(v)$. Since Equation 32 is of exponential distribution form integrating over $b_{[uv]}$ we see that each segment contributes a term

$$1/H^{\theta,\alpha}_{m(v)-1}. \quad (33)$$

6.6 Relationship to Gibbs fragmentation trees

Various models studied in the probability literature relate to the PYDT and DDT. We discuss some of the most closely related here. Gibbs fragmentation trees [McCullagh et al., 2008] define a Markovian, consistent probability distribution over the space of cladograms, $\mathbb{T}_N$. A random tree is consistent if its subtrees are distributed like the whole tree, and Markovian if disjoint subtrees are distributed independently of each other and their ancestors. Markovian random trees have a distribution defined by a splitting rule, p, which gives the probability of a specific tree, $T \in \mathbb{T}_N$ through

$$\mathbb{P}(T) = \prod_{v\in\mathcal{S}'(T)} p(n_1^v, ..., n_{K_v}^v) \quad (34)$$

where $\mathcal{S}'(T)$ is the set of internal nodes of T, n_k^v is the number of leaves below the k-th of v. Gibbs fragmentation trees have a splitting rule p of the form

$$p(n_1, ..., n_K) = \frac{g(K)}{Z(m)}\prod_{i=1}^{K} w(n_i) \quad (35)$$

where $m = \sum_i n_i$. McCullagh et al. [2008] (Theorem 8) show that for such a splitting rule to be consistent, we must have

$$w(n) = \frac{\Gamma(n-\alpha)}{\Gamma(1-\alpha)}, \quad g(K) = \alpha^{K-2}\frac{\Gamma(K+\theta/\alpha)}{\Gamma(2+\theta/\alpha)} \quad (36)$$

Comparing $g(K)$ with Equation 26 and the product over $w(n_i)$ with Equation 22, we see that the dependence on the n_k's and K at each node is the same for the PYDT and the Gibbs fragmentation tree is the

same, so the distribution over tree structures $\mathbb{T}_N$ is the same marginalising over divergence times in the PYDT (the terms resulting from this marginalisation, shown in Equation 33, contribute to the normalisation $Z(m)$ in Equation 35). In the binary case McCullagh et al. [2008] discuss an embedding into continuous time which is analogous to that for the PYDT, and which is extended to the multifurcating case in Haas et al. [2008], Proposition 3. These references confirm the uniqueness, up to changes in $a(t)$, of our embedding into continuous time.

6.7 Relationship to Aldous' beta-splitting model and tree balance

Gibbs fragmentation trees generalise the earlier beta splitting model of Aldous [1996], which correspond to the binary branching ($\theta = -2\alpha$) PYDT. The valid parameter range for the binary PYDT is $\alpha < 1$. As mentioned in MacKay and Broderick [2007] the parameter α controls the balance of the tree. As noted by Aldous [1996], for $\alpha < 0$ the reinforcing scheme here can be considered as the result of marginalising out a latent variable, p_v at every internal node, v with prior, $p_v \sim \text{Beta}(-\alpha, -\alpha)$. For $\alpha = -1$ this is a uniform distribution. For α close to 0 the distribution will concentrate towards point masses at 0 and 1, i.e. towards $(\delta(0) + \delta(1))/2$, so that one branch will be greatly preferred over the other, making the tree more unbalanced. As $\alpha \to -\infty$ the mass of the beta distribution concentrates towards a point mass at 0.5 encouraging the tree to be more balanced. For $0 \leq \alpha < 1$, p_v no longer has a valid prior density but we see the reinforcing is still valid. A simple measure of the imbalance of tree is given by Colless's I_n [Colless, 1982], given by

$$I_n = \sum_{v \in \mathcal{T}} |l(v) - r(v)| \tag{37}$$

where n is the number of leaves, v ranges over all internal nodes in the tree, and $l(v)$ and $r(v)$ are the number of data points that followed the left and right branches respectively. The maximum of I_n is $I_n^{\max} = (n-1)(n-2)/2$ so we can define the normalised version $\bar{I}_n = I_n / I_n^{\max} \in [0, 1]$. An alternative is the number of unbalanced nodes in a tree, J_n [Rogers, 1996], i.e.

$$J_n = \sum_{v \in \mathcal{T}} (1 - \mathbb{I}[l(v) = r(v)]) \tag{38}$$

where $\mathbb{I}$ is the indicator function. The maximum for J_n is $J_n^{\max} = n - 2$ so again we can define $\bar{J}_n := J_n / J_n^{\max} \in [0, 1]$. While we are unaware of formulae for $\mathbb{E}I_n$ or $\mathbb{E}J_n$ for general α, using the connection to Aldous' beta-splitting trees these expectations are easily calculated for any n by the recursion

$$\mathbb{E}L_n = \sum_{i=1}^{n-1} p(n-i, i) \left[\mathbb{E}L_i + \mathbb{E}L_{n-i} + g(n-i, i) \right], \tag{39}$$

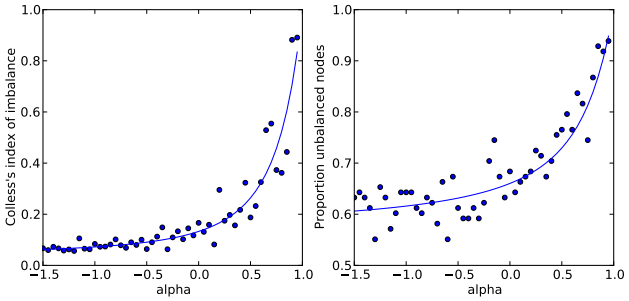


Fig. 6: Two measures of tree imbalance for samples from the binary Pitman-Yor Diffusion Tree with $\theta = -2\alpha$ for varying α and $N = 100$. *Solid lines*: expected values calculated using recursion formulae. *Points:* empirical indices calculated using generated trees. **Left**: Colless's index of balance, see Equation 37. **Right**: Proportion of unbalanced nodes, see Equation 38.

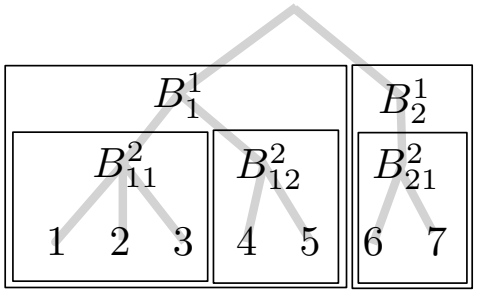


Fig. 7: A hierarchical partitioning of the integers $\{1, ..., 7\}$ showing the underlying tree structure.

where $p(.,.)$ is defined as in Equation 35, $L \in \{I, J\}$ and $g(l, r) := |l - r|$ for I_n or $g(l, r) := 1 - \mathbb{I}[l = r]$ for J_n. Both measures of tree imbalance increase with α, as shown in Figure 6, with the biggest effects occurring in the interval $\alpha \in [0, 1]$. Both expected values calculated using Equation 39 and empirical values calculated by explicitly sampling trees from the binary PYDT with varying α are shown.

6.8 The continuum limit of a nested CRP

The PYDT can be derived as the limiting case of a specific nested Chinese Restaurant Process [Blei et al., 2004] model (nCRP). We will first show how to construct the Dirichlet Diffusion Tree as the limit of a simple nCRP model. We then modify this model so that the limiting process is instead the PYDT.

The nested CRP gives a distribution over hierarchical partitions (see Section 2). Denote the K blocks in the first level as $\{B_k^1 : k = 1, ..., K\}$. We can now imagine partitioning the elements in each first level block, B_k^1, according to independent CRPs. Denote the blocks in the second level partitioning of B_k^1 as $\{B_{kl}^2 : l = 1, ..., K_k\}$. We can recurse this construction for as many iterations S as we please, forming a S deep hierarchy of blocks v. Each element belongs to just a single block at each level, and the partitioning forms a tree structure: consider the

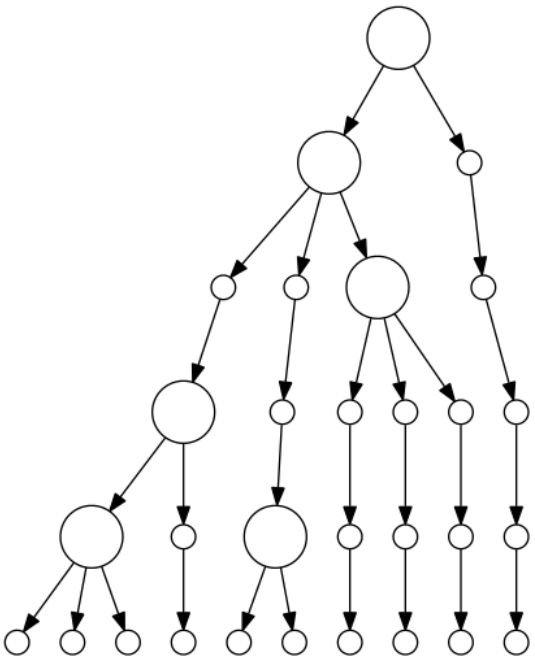


Fig. 8: A draw from a $S = 10$-level nested Chinese restaurant process.

unpartitioned set $[n]$ as the root, with children B_k^1. Each B_k^1 then has children B_{kl}^2, and so on down the tree, see Figure 7. Nodes with only a single child are allowed under this construction, resulting in a prior over $\mathbb{T}'_N$ (see Section 2). An example draw from an $S = 10$ level nested CRP is shown in Figure 8. It is certainly possible to work with this model directly (see Blei et al. [2004, 2010] and more recently Steinhardt and Ghahramani [2012]), but there are disadvantages, such as having to choose the depth S, or avoid this in some more convoluted way: Adams et al. [2010] use a stick breaking representation of the nCRP with unbounded depth S augmented with a probability of stopping at each internal node, and Steinhardt and Ghahramani [2012] allow $S \to \infty$ but integrate over countably infinite chains of nodes with only one child. While the latter is appealing for discrete data where any bounded diffusion process on the infinitely deep hierarchy concentrates towards a point mass, it is not clear how to adapt this approach to modelling continuous data.

Theorem 2: Associate each level s in an S-level nCRP on $\mathbb{T}'_N$ with "time" $t_s = \frac{s-1}{S} \in [0, 1)$, and let the concentration parameter at level s be $a(t_s)/S$, where $a : [0, 1] \mapsto \mathbb{R}^+$ is Riemann integrable. Taking the limit $S \to \infty$ recovers the Dirichlet Diffusion Tree [Neal, 2003b] on $\mathbb{P}_N$ with divergence function $a(t)$.

Intuitively, any connected chains of nodes with only one child in the nCRP will become branches in the DDT: the length of the chain in $\mathbb{T}'_N$ becomes an increasingly accurate approximation to the branch length in $\mathbb{P}_N$. Nodes in the nCRP which do have multiple children become branch points in the DDT, but we find that these will always be binary splits.

Proof: From Equation 2 the probability of forming a new branch (block) at a node on a chain of nodes with only single children (a single block) at level u is (from the definition of the CRP)

$$\frac{a(t_u)/S}{m + a(t_u)/S}, \tag{40}$$

where m is the number of previous data points that went down this chain. This behaves as $a(t_u)/(Sm)$ as S becomes large. Informally associating the time interval $1/S$ with the infinitesimal time interval dt directly yields the DDT divergence probability $a(t)dt/m$. More formally, we aim to show that the distribution over divergence times is given by the DDT in the limit $S \to \infty$. The number of nodes k in a chain starting at level u until divergence at level $v = u + k$ is distributed as

$$\underbrace{\frac{a(t_{u+k})/S}{m + a(t_{u+k})/S}}_{\text{prob new block at level u+k}} \prod_{i=1}^{k-1} \underbrace{\left(1 - \frac{a(t_{u+i})/S}{m + a(t_{u+i})/S}\right)}_{\text{prob not forming new block at level u+i}}, \tag{41}$$

where $t_u = \frac{u}{S+1}$ is the "time" of the branch point at the top of the chain. For constant $a(.)$ Equation 41 is a geometric distribution in k. We now take the limit $S \to \infty$, holding $\frac{k-1}{S} = t - t_u$ and $\frac{u-1}{S} = t_u$ fixed so that we also have $k \to \infty$. We analyse how the product in Equation 41 behaves:

$$\begin{aligned}
&\lim_{S\to\infty} \prod_{i=1}^{k-1} \left(1 - \frac{a(t_{u+i})/S}{m + a(t_{u+i})/S}\right) \\
&= \lim_{S\to\infty} \prod_{i=1}^{k-1} \left(1 - \frac{a(t_{u+i})}{Sm}\right) \\
&= \exp\left\{\lim_{k\to\infty} \sum_{i=1}^{k-1} \log\left(1 - \frac{a(t_u + i\frac{t-t_u}{k-1}))}{m} \frac{t-t_u}{k-1}\right)\right\}
\end{aligned} \tag{42}$$

where we have used that $t_{u+i} = t_u + \frac{i-1}{S}$ and $\frac{1}{S} = \frac{t-t_u}{k-1}$. We are able to exchange the order of the exp and lim operations because of the continuity of exp. Now we use that $\log(1 - x) = -x - O(x^2)$ to give

$$\begin{aligned}
&\log\left(1 - \frac{a(t_u + i(t-t_u)/(k-1))}{m} \frac{t-t_u}{k-1}\right) \\
&\qquad = -\frac{a(t_u + i(t-t_u)/(k-1))}{m} \frac{t-t_u}{k-1} - O(k^{-2})
\end{aligned}$$

which allows us to see that the limiting value of the exponent in Equation 42 is simply a Riemann integral

$$\begin{aligned}
&\lim_{k\to\infty} \sum_{i=1}^{k-1} \left[\frac{a(t_u + i(t-t_u)/(k-1)))}{m} \frac{t-t_u}{k-1} - O(k^{-2})\right] \\
&\qquad = \frac{1}{m} \int_{t_u}^{t} a(\tau) d\tau
\end{aligned} \tag{43}$$

Thus taking the limit $S \to \infty$ of Equation 41 we find the divergence time, $t_v = t_u + \frac{k-1}{S}$ is distributed

$$\frac{a(t_v)}{m} \exp\left\{-\frac{\int_{t_u}^{t_v} a(\tau) d\tau}{m}\right\} \tag{44}$$

as for the Dirichlet Diffusion Tree, the waiting time in a inhomogeneous Poisson process with rate function $a(.)$. In the simple case of constant $a(.) = a$ the geometric distribution becomes an exponential waiting time with parameter a/m.

At existing branch points the probability of going down an existing branch k is $|B_k|/(m+a(t_s)/S)$ which is simply $|B_k|/m$ in the limit $S \to \infty$, recovering the DDT. The probability of a third cluster forming at an existing branch point is given by Equation 40 which clearly tends to 0 in the limit, resulting in the binary nature of the DDT. □

An alternative construction would use a homogeneous (constant) rate $a(.) = 1$ and then use the Poisson process mapping theorem [Kingman, 1993] to transform this process into a DDT with arbitrary non-atomic divergence function $a(.)$, following Lemma 3. This emphasises that the rate function, $a(.)$ can in reality be any measurable function but assuming it is Riemann integrable simplifies the proof.

It was essential in this construction that we drove the concentration parameter to zero as the depth of the tree increases. This avoids complete instantaneous fragmentation of the tree. For any time $\epsilon > 0$ there will be infinitely many levels in the nCRP before time ϵ when we take $S \to \infty$. If the CRPs in these levels have strictly positive concentration parameters, the tree will have completely fragmented to individual samples before ϵ almost surely. This is clearly undesirable from a modelling perspective since the samples are then independent.

It is interesting that despite the finite level nCRP allowing multifurcating "branch points" the continuum limit taken in Theorem 2 results in binary branch points almost surely. We will show how to rectify this limitation in Theorem 3 where we present the analogous construction for the Pitman-Yor Diffusion Tree. First we mention the possibility of using the two parameter Chinese restaurant process (the urn representation of the Pitman-Yor process [Pitman and Yor, 1997]) in the construction of the DDT in Theorem 2. This in principle does not introduce any additional difficulty. One can imagine a nested two parameter CRP, using an analogous rate function $c(t)$ to give the discount parameter for each level. The problem is that it would still be necessary to avoid instantaneous fragmentation by driving the discount parameters to zero as $S \to \infty$, e.g. by setting the discount parameter at time t to $c(t)/S$. It is straightforward to see that this will again recover the DDT, although with rate function $a(t) + c(t)$: the probability of divergence will be $(a(t) + c(t))/(Sm)$ when there is one block, i.e. on a chain, so the logic of Theorem 2 follows; the probability of forming a third cluster at any branch point is $(a(t)+2c(t))/(Sm)$ which tends to zero as $S \to \infty$; and finally the probability of following a branch k at a branch point is $\frac{n_k - c(t_s)/S}{m+a(t_s)/S}$ which again recovers the DDT factor n_k/m in the limit.

Thus the construction of the DDT in Theorem 2 destroys both the arbitrary branching structure of the underlying finite level nCRP and does not allow the extra flexibility provided by the two parameter CRP. This has ramifications beyond the construction itself: it implies that attempting to use a simple nCRP model in a very deep hierarchy has strong limitations. Either only the first few levels will be used, or the probability of higher order branching events must be made exponentially small. This is not necessarily a problem for discrete data [Steinhardt and Ghahramani, 2012]. Additionally, the two parameter generalisation cannot be used to any advantage.

To obtain the multifurcating PYDT rather than the binary DDT we will modify the construction above.

Associate level s of an S-level nested partitioning model with time

$$t_s = (s-1)/S.$$

For a node at level s with only $K = 1$ cluster, let the probability of forming a new cluster be $\frac{a'(m,s)/S}{m+a'(m,s)/S}$ where

$$a'(m,s) = ma(t_s)\frac{\Gamma(m-\alpha)}{\Gamma(m+1+\theta)}, \tag{45}$$

where $0 \leq \alpha < 1, \theta > -2\alpha$ are hyperparameters. At an existing branch point (i.e. if the number of existing clusters is $K \geq 2$) then let the probabilities be given by the two parameter CRP, i.e. the probability of joining an existing cluster k is

$$\frac{n_k - \alpha}{m+\theta}, \tag{46}$$

where n_k is the number of samples in cluster k and m is the total number of samples through this branch point so far. The probability of diverging at the branch point and creating a new branch is

$$\frac{\theta + \alpha K}{m+\theta}, \tag{47}$$

where K is the current number of clusters from this branch point.

Theorem 3: In the limit $S \to \infty$ the construction above becomes equivalent to the PYDT with rate function $a(t)$, concentration parameter θ and discount parameter α.

Proof: Showing the correct distribution for the divergence times is analogous to the proof for Theorem 2. The probability of divergence from a chain at any level s behaves as $\frac{a'(m,s)}{Sm}$ as $S \to \infty$. The number of nodes k in a chain starting at level v until divergence is distributed:

$$\frac{a'(m,b+k)}{Sm}\prod_{i=1}^{k-1}\left(1 - \frac{a'(m,b+i)}{Sm}\right)$$
$$= \frac{a(t_{b+k})\Gamma(m-\alpha)}{S\Gamma(m+1+\theta)}\prod_{i=1}^{k-1}\left(1 - \frac{a(t_{b+i})\Gamma(m-\alpha)}{S\Gamma(m+1+\theta)}\right). \tag{48}$$

Following the proof of Theorem 2 in the limit $S \to \infty$ this becomes

$$\frac{\Gamma(m-\alpha)}{S\Gamma(m+1+\theta)}a(t)\exp\left\{-\frac{\Gamma(m-\alpha)}{\Gamma(m+1+\theta)}\int_{t_v}^{t}a(\tau)d\tau\right\}.$$

Since Equations 12 and 46, and Equations 11 and 47 are the same, it is straightforward to see that the probabilities for higher order branching events are exactly as for the PYDT, i.e. given by Equation 22. □

The finite level model of Theorem 3 is not exchangeable until we take the limit $S \to \infty$. Every node at level s with only $K = 1$ cluster contributes a factor

$$\prod_{i=1}^{m-1}\left(1-\frac{a'(i,s)/S}{j+a'(i,s)/S}\right), \tag{49}$$

where $a'(.)$ is defined in Equation 45 and m is the total number of samples having passed through this node. This factor does not depend on the order of the data points. Now consider a node with $K \geq 2$ clusters at level s. Assume the i-th sample diverged to create this branch point initially. The first $i-1$ samples did not diverge, the first contributing no factor, and the subsequent $i-2$ contributing a total factor

$$\prod_{j=2}^{i-1}\left(1-\frac{a'(j,s)/S}{m+a'(j,s)/S}\right). \tag{50}$$

Although this factor tends to 1 as $S \to \infty$, for finite S it depends on i. The probability of the i-th sample diverging to form the branch point is

$$\frac{a'(i,s)/S}{m+a'(i,s)/S} = \frac{a(t_s)}{S+a'(i,s)/i}\frac{\Gamma(i-\alpha)}{\Gamma(i+1+\theta)}. \tag{51}$$

The probability contributed by the samples after i is exactly the same as Equation 21 in Lemma 1, given by

$$\frac{\prod_{k=3}^{K_b}[\theta+(k-1)\alpha]\Gamma(i+\theta)\prod_{l=1}^{K_b}\Gamma(n_l^b-\alpha)}{\Gamma(m(v)+\theta)\Gamma(i-1+\alpha)}. \tag{52}$$

Multiplying this by Equation 51 we obtain

$$\frac{a(t_s)}{S+a'(i,s)/i}\frac{\prod_{k=3}^{K_b}[\theta+(k-1)\alpha]\prod_{l=1}^{K_b}\Gamma(n_l^b-\alpha)}{\Gamma(m(v)+\theta)}. \tag{53}$$

It is easy enough to see that we will recover the correct expression for the PYDT in the limit $S \to \infty$, using $1/S \to dt$. However, for finite S this factor, and the factor in Equation 50, depend on i, so we do not have exchangeability.

While other, exchangeable, finite S models might exist that give the PYDT in the continuum limit we are unaware of such a construction.

7 Hierarchical clustering model

To use the PYDT as a hierarchical clustering model we must specify a likelihood function for the data given the leaf locations of the PYDT, and priors on the hyperparameters. We use a Gaussian observation model for multivariate continuous data and a probit model for binary vectors. We use the divergence function $a(t) = c/(1-t)$ and specify the following priors on the hyperparameters:

$$\theta \sim \mathrm{G}(a_\theta, b_\theta), \qquad \alpha \sim \mathrm{Beta}(a_\alpha, b_\alpha), \tag{54}$$

$$c \sim \mathrm{G}(a_c, b_c), \qquad 1/\sigma^2 \sim \mathrm{G}(a_{\sigma^2}, b_{\sigma^2}), \tag{55}$$

where $G(a,b)$ is a Gamma distribution with shape, u and rate, v. In all experiments we used $a_\theta = 2, b_\theta = .5, a_\alpha = 1, b_\alpha = 1, a_c = 1, b_c = 1, a_{\sigma^2} = 1, b_{\sigma^2} = 1$.

8 Inference

We propose three inference algorithms: two MCMC samplers and a more computationally efficient greedy EM algorithm. All three algorithms marginalise out the locations of internal nodes using belief propagation, and are capable of learning the hyperparameters $\Theta := \{c, \sigma^2, \theta, \alpha\}$ if desired. Let $\mathcal{P} \in \mathbb{P}_N$ be a tree structure including branch lengths and $\mathcal{D}$ be data observed at the leaves.

8.1 MCMC sampler

We demonstrate two alternative but related MCMC sampling methods to explore the posterior over the tree structure and divergence times, i.e. over the space $\mathbb{P}_N$ of phenograms. Both sample the structure and divergence times using moves that detach and reattach subtrees. For both samplers, subtrees are detached using the function

- $(\mathcal{S}, \mathcal{R}, x_0) := \textsc{RandomDetach}(\mathcal{P})$, which chooses a node uniformly at random from $\mathcal{P}$ and detaches the subtree $\mathcal{S}$ rooted at that node. The detached subtree $\mathcal{S}$, the remaining tree $\mathcal{R}$ and the detachment position $x_0 \in \mathcal{R}$ are returned.

The subtree may be a single leaf node. Both samplers make use of the unnormalised posterior $f : \mathbb{P}_N \to \mathbb{R}^+$, i.e. $f(\mathcal{P}) = P(\mathcal{D}|\mathcal{P}, \sigma^2)P(\mathcal{P}|c, \theta, \alpha) \propto P(\mathcal{P}|\mathcal{D}, \Theta)$, where $P(\mathcal{D}|\mathcal{P}, \sigma^2)$ is the marginal likelihood of the tree structure $\mathcal{P}$ calculated using belief propagation. Let root($\mathcal{S}$) be the root node of subtree $\mathcal{S}$ and $t(u)$ be the time of node u.

We confirmed the correctness of both samplers using joint distribution tests [Geweke, 2004], using test functions such as the time to the first divergence (the root node).

8.1.1 MH sampler

The simplest sampler is based on Metropolis Hastings, for which pseudocode is given in Algorithm 1. To propose a new position in the tree $\mathcal{P} \in \mathbb{P}_N$ for the detached subtree we use the functions

- $x' := \textsc{Prior}(\mathcal{R})$ which follows the procedure for generating a new sample on the remaining tree $\mathcal{R}$ and returns the divergence position $x' \in \mathcal{R}$ and,
- $\mathcal{P}' := \textsc{Attach}(\mathcal{S}, \mathcal{R}, x')$ which attaches the subtree $\mathcal{S}$ at $x' \in \mathcal{R}$, which may be on a segment, in which case a new parent node is created, or at an existing internal node, in which case the subtree becomes a child of that node. The resulting tree $\mathcal{P}'$ is returned.

If divergence occurred at a time later than the divergence time of the root of the subtree, $t(\text{root}(\mathcal{S}))$, we must repeat the procedure until this is not the case. The acceptance ratio a is then calculated using the marginal likelihood f of the new proposed tree $\mathcal{P}'$ marginalizing over the internal node locations, the marginal likelihood for the original tree, and the proposal and reverse proposal probabilities, which are simply given by the probability of divergence at the reattachment and detachment positions respectively (since which subtree to detach is chosen uniformly at random). Denote by $q_{\mathcal{R}}(x)$ the probability under the prior of a new data point diverging off the remaining subtree $\mathcal{R}$ at x.

Algorithm 1 MH sampler

Require: Initial tree $\mathcal{P}^0 \in \mathbb{P}_N$, unnormalised posterior $f(.)$, number of samples S
  for $i = 1 \to S$ **do**
    $(\mathcal{S}, \mathcal{R}, x_0) := \text{RANDOMDETACH}(\mathcal{P}^{i-1})$
    $x' := \text{PRIOR}(\mathcal{R})$
    while $t(x') > t(\text{root}(\mathcal{S}))$ **do**
      $x' := \text{PRIOR}(\mathcal{R})$
    end while
    $\mathcal{P}' := \text{ATTACH}(\mathcal{S}, \mathcal{R}, x')$
    $a := \frac{f(\mathcal{P}')q_{\mathcal{R}}(x_0)}{f(\mathcal{P}^{i-1})q_{\mathcal{R}}(x')}$
    Sample $u \sim U[0,1]$
    if $u < a$ **then**
      $\mathcal{P}^i := \mathcal{P}'$
    else
      $\mathcal{P}^i := \mathcal{P}^{i-1}$
    end if
    Sample hyperparameters
  end for

8.1.2 Slice sampler

We propose a novel sampler based on slice sampling. Our slice sampling scheme is distinct from that proposed in Neal [2003b] in that a detached subtree may be reattached anywhere in the remaining tree, whereas Neal's scheme only allows reattachment along the path from the root to the sibling of the detached node. While slice sampling is most commonly used for univariate distributions, it is straightforward to extend to a tree. We first consider the binary setting, and then discuss the extension to the general multifurcating case. The steps of the slice sampler are shown in the cartoon in Figure 9 and described in pseudocode in Algorithm 2. For notational convenience we define the unnormalised posterior probability of reattachment at any point on the remaining tree $\mathcal{R}$ as $F : \mathcal{R} \to \mathbb{R}^+$. i.e. $F(x) = f(\text{ATTACH}(\mathcal{S}, \mathcal{R}, x))$, where we have suppressed the dependence on $\mathcal{S}$ and $\mathcal{R}$. Slice sampling involves introducing the auxiliary real variable y, and defining the joint distribution which is uniform over the region $U = \{(x, y) : x \in \mathcal{R}, 0 < y < F(x)\}$. Sampling from this joint distribution and disregarding y gives samples from the normalised $F(x)$ as required.

The sampler proceeds as follows. As for the MH sampler, a subtree $\mathcal{S}$ is picked uniformly at random from the initial tree $\mathcal{P}^0$ using the function RANDOMDETACH (see Figure 9a) where we again denote the original attachment position as $x_0 \in \mathcal{R}$. We then sample $y \sim U[0, F(x_0)]$, implicitly defining a slice $S = \{x \in \mathcal{R} : y < F(x)\}$. In Figure 9b the value of $F(.)$ on S is shown by the width of the grey region perpendicular to the edge. We must now define a set $\mathcal{I}_1 \subseteq \mathcal{R}$ containing x_0 and most of S. Since $\mathcal{R}$ is bounded we simply set $\mathcal{I}_1 := \{x \in \mathcal{R} : t(x) < t(root(\mathcal{S}))\}$ where we have excluded reattachment positions that would result in a negative branch length. An analogous method to the standard stepping out approach for univariate slice sampling might be useful, but maintaining detailed balance in this setting is more challenging. We sample a potential reattachment position x_1 from the uniform distribution on $\mathcal{I}_1$, an operation we denote by $x_1 \sim \text{UNIFORM}[\mathcal{I}_1]$. If $F(x_1) < y$ then x_1 is rejected we shrink $\mathcal{I}$ using the function

- $\mathcal{I}_{j+1} := \text{SHRINK}(\mathcal{I}_j, x_0, x_j)$, which calculates $\mathcal{I}_{j+1}$ by removing from $\mathcal{I}_j$ all parts of the tree that could only be reached from x_0 by passing through x_j.

This is shown in Figure 9c where the extent of the grey regions specifies the shrunk $\mathcal{I}'$. We repeat this procedure: sampling $x_j \sim \text{UNIFORM}[\mathcal{I}_j]$, shrinking $\mathcal{I}$ and incrementing j until x_j is accepted since $F(x_j) > y$ and $\mathcal{R}$ is reattached at x_j (see Figure 9d and e). Eventually x_j will be accepted since $\mathcal{I}_j$ will concentrate around x_0.

The method maintains detailed balance by the same argument as for the univariate shrinkage procedure [Neal, 2003a]: the probability of transitioning from x_0 to x_j (where x_j is accepted) is equal to that for x_j to x_0 using the same intermediate rejected x's. The advantage of the slice sampling procedure over the simple MH sampler is that the slice sampler uses rejected reattachment positions to rapidly narrow down where in the tree might be a sensible reattachment position, where MH continues to blindly propose reattachment positions from the prior.

It is not immediately obvious how to use the slice sampling approach in the multifurcating setting because of the atoms at the nodes in the unnormalised posterior F. Our solution is to represent each atom, i, with mass g_i, by a rectangle of width $\delta = 0.1$ and height g_i/δ, inserted in $\mathcal{R}$ at the nodes of the tree: these are the rectangles shown at the nodes in Figure 9. The approach described above can then be straightforwardly applied.

8.1.3 Smoothness hyperparameter, c.

From Equation 27 the conditional posterior for c is

$$\text{G}\left(a_c + |\mathcal{I}|, b_c + \sum_{i \in \mathcal{I}} J_{n^i}^{\theta,\alpha} \log(1 - t_i)\right), \qquad (56)$$

where $\mathcal{I}$ is the set of internal nodes of the tree.

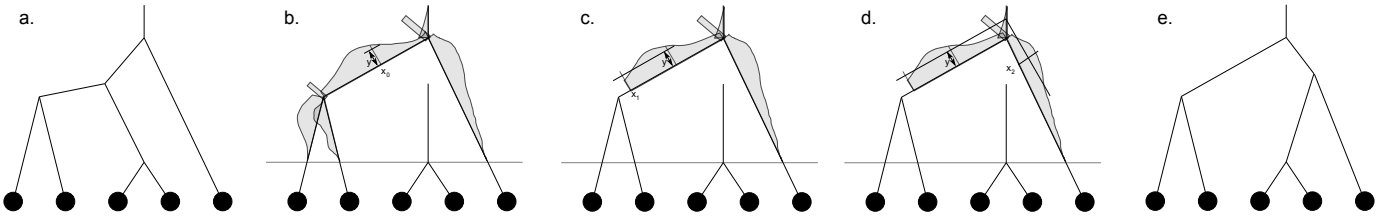


Fig. 9: Steps involved in the slice sampling procedure. Width of the grey region perpendicular to each edge shows the unnormalised posterior $F(x)$ on the remaining tree $\mathcal{R}$, and the extent of this region shows $\mathcal{I}$. Atoms in $F(x)$ at the nodes are represented both schematically and mathematically as rectangles. a) Initial tree. b) Randomly chosen subtree is detached at x_0 and we sample $y \sim U[0, F(x_0)]$. c) $x_1 \sim$ Uniform$[\mathcal{I}_1]$ is rejected because $F(x_1) < y$ and $\mathcal{I}_1$ is shrunk to give $\mathcal{I}_2$. d) $x_2 \sim$ Uniform$[\mathcal{I}_2]$ is accepted because $F(x_2) > y$. e) Subtree is reattached at x_2.

Algorithm 2 Slice sampler

Require: Initial tree $\mathcal{P}^0 \in \mathbb{P}_N$, unnormalised posterior $f(.)$, number of samples S
  for $i = 1 \rightarrow S$ **do**
    $(\mathcal{S}, \mathcal{R}, x_0) :=$ RANDOMDETACH$(\mathcal{P}^{i-1})$
    Sample $y \sim U[0, f(\mathcal{P}^{i-1})]$
    $\mathcal{I}_1 := \{x \in \mathcal{R} : t(x) < t(root(\mathcal{S}))\}$
    $j := 1$
    Sample $x_1 \sim$ UNIFORM$(\mathcal{I}_1)$
    $\mathcal{P}' :=$ ATTACH$(\mathcal{S}, \mathcal{R}, x_1)$
    while $f(\mathcal{P}') < y$ **do**
      $\mathcal{I}_{j+1} :=$ SHRINK$(\mathcal{I}_j, x_0, x_j)$
      Sample $x_{j+1} \sim$ UNIFORM$(\mathcal{I}_{j+1})$
      $\mathcal{P}' :=$ ATTACH$(\mathcal{S}, \mathcal{R}, x_{j+1})$
      $j := j + 1$
    end while
    $\mathcal{P}^i := \mathcal{P}'$
    Sample hyperparameters
  end for

8.1.4 Data variance, σ^2.

It is straightforward to sample $1/\sigma^2$ given divergence locations. Having performed belief propagation it is easy to jointly sample the divergence locations using a pass of backwards sampling. From Equation 25 the Gibbs conditional for the precision $1/\sigma^2$ is then

$$\mathrm{G}(a_{\sigma^2}, b_{\sigma^2}) \prod_{[uv] \in \mathcal{S}(\mathcal{T})} \mathrm{G}\left(D/2 + 1, \frac{||x_u - x_v||^2}{2(t_v - t_u)}\right), \quad (57)$$

where $||\cdot||$ denotes Euclidean distance.

8.1.5 Pitman-Yor hyperparameters, θ and α.

We use slice sampling to sample θ and α. We reparameterise in terms of the logarithm of θ and the logit of α to extend the domain to the whole real line. The terms required to calculate the conditional probability are those in Equations 22 and 24.

8.2 Greedy Bayesian EM algorithm

As an alternative to MCMC here we use a Bayesian EM algorithm to approximate the marginal likelihood for a given tree structure, which is then used to drive a greedy search over tree structures, following our work in Knowles et al. [2011].

8.2.1 EM algorithm.

In the E-step, we use message passing to integrate over the locations and hyperparameters. In the M-step we maximize the lower bound on the marginal likelihood with respect to the divergence times. For each node i with divergence time t_i we have the constraints $t_p < t_i < \min(t_l, t_r)$ where t_l, t_r, t_p are the divergence times of the left child, right child and parent of i respectively.

We jointly optimise the divergence times using LBFGS [Liu and Nocedal, 1989]. Since the divergence times must lie within $[0, 1]$ we use the reparameterisation $s_i = \log[t_i/(1 - t_i)]$ to extend the domain to the real line, which we find improves empirical performance. From Equations 25 and 6 the lower bound on the log evidence is a sum over all branches $[pi]$ of expressions of the form:

$$(\langle c \rangle J^{\alpha,\beta}_{\mathbf{n}^i} - 1)\log(1 - t_i) - \frac{D}{2}\log(t_i - t_p) - \langle \frac{1}{\sigma^2} \rangle \frac{b_{[pi]}}{t_i - t_p} \quad (58)$$

where $b_{[pi]} = \frac{1}{2}\sum_{d=1}^{D} \mathbb{E}[(x_{di} - x_{dp})^2]$, x_{di} is the location of node i in dimension d, and p is the parent of node i. The full lower bound is the sum of such terms over all nodes. The expectation required for $b_{[pi]}$ is readily calculated from the marginals of the locations after message passing. Differentiating to obtain the gradient with respect to t_i is straightforward so we omit the details. Although this is a constrained optimization problem (branch lengths cannot be negative) it is not necessary to use the log barrier method because the $1/(t_i - t_p)$ terms in the objective implicitly enforce the constraints.

8.2.2 Hyperparameters.

We use variational inference to learn Gamma posteriors on the inverse data variance $1/\sigma^2$ and smoothness c. The variational updates for c and $1/\sigma^2$ are the same as the conditional Gibbs distributions in Equations 56 and 57 respectively. We optimize the Pitman-Yor parameters, θ and α by coordinate descent using golden section search on the terms in Equations 22 and 24.

8.2.3 Search over tree structures

The EM algorithm approximates the marginal likelihood for a fixed tree structure $\mathcal{P}$. We maintain a list of K-best trees (typically $K = 10$) which we find gives good empirical performance. Similarly to the sampler, we search the space of tree structures by detaching and reattaching subtrees. We choose which subtree to detach at random. We can significantly improve on reattaching at random by calculating the local contribution to the evidence that would be made by attaching the root of the subtree to the midpoint of each possible branch and at each possible branch point. We then run EM on just the three best resulting trees. We found construction of the initial tree by sequential attachment of the data points using this method to give very good initializations.

8.3 Likelihood models

Connecting our PYDT module to different likelihood models is straightforward: we use a Gaussian observation model and a probit model for binary vectors. The MCMC algorithm slice samples auxiliary variables and the EM algorithm uses EP [Minka, 2001] on the probit factor, implemented using the runtime component of the Infer.NET framework [Minka et al., 2010].

9 Results

We present results on synthetic and real world data, both continuous and binary.

9.1 Synthetic data

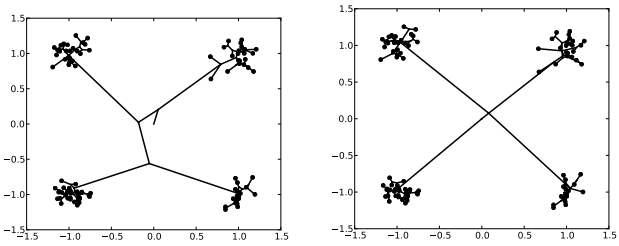
We first compare the PYDT to the DDT on a simple synthetic dataset with $D = 2, N = 100$, sampled from the density

$$f(x, y) = \frac{1}{4} \sum_{\bar{x} \in \{-1,1\}} \sum_{\bar{y} \in \{-1,1\}} N(x; \bar{x}, 1/8) N(y; \bar{y}, 1/8)$$

The optimal trees learnt by 100 iterations of the greedy EM algorithm are shown in Figure 10. While the DDT is forced to arbitrarily choose a binary branching structure over the four equi-distant clusters, the PYDT is able to represent the more parsimonious solution that the four clusters are equally dependent. Both models find the fine detail of the individual cluster samples which may be undesirable; investigating whether learning a noise model for the observations alleviates this problem is a subject of future work.

9.2 Density modeling

In Adams et al. [2008] the DDT was shown to be an excellent density model on a $D = 10, N = 228$ dataset of macaque skull measurements, outperforming a kernel density and Dirichlet process mixture of Gaussians, and sometimes the Gaussian process density sampler proposed in that paper. We compare the PYDT to the DDT on the same dataset, using the same data preprocessing



(a) DDT (b) PYDT

Fig. 10: Optimal trees learnt by the greedy EM algorithm for the DDT and PYDT on a synethic dataset with $D = 2, N = 100$.

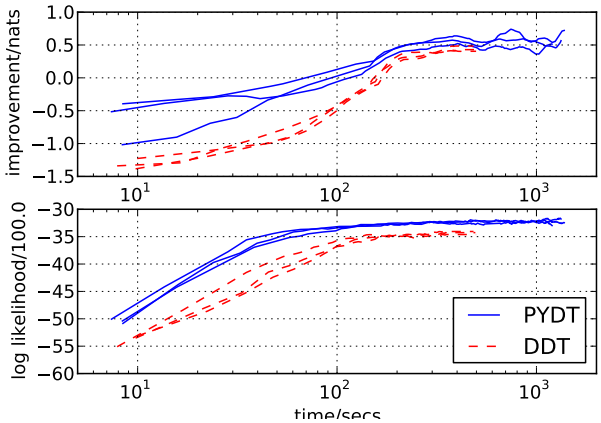


Fig. 11: Density modelling of the $D = 10, N = 200$ macaque skull measurement dataset of Adams et al. [2008]. *Top*: Improvement in test predictive likelihood compared to a kernel density estimate. *Bottom*: Marginal likelihood of current tree. The shared x-axis is computation time in seconds.

and same three train test splits ($N_{\text{train}} = 200, N_{\text{test}} = 28$) as Adams et al. [2008]. The performance using the MH sampler is shown in Figure 11. The PYDT finds trees with higher marginal likelihood than the DDT, which corresponds to a moderate improvement in predictive performance. The posterior hyperparameters were reasonably consistent across the three train/test splits, with $\theta = 2.3 \pm 0.4$ and $\alpha = 0.23 + 0.08$ averaged over the last 100 samples for the first training split for example. Inference in the PYDT is actually slightly more efficient computationally than in the DDT because on average the smaller number of internal nodes reduces the cost of belief propagation over the divergence locations, which is the bottleneck of the algorithm (being a subroutine of the tree search procedure).

9.3 Binary example

To demonstrate the use of an alternative observation model we use a probit observation model in each dimension to model 102-dimensional binary feature vectors

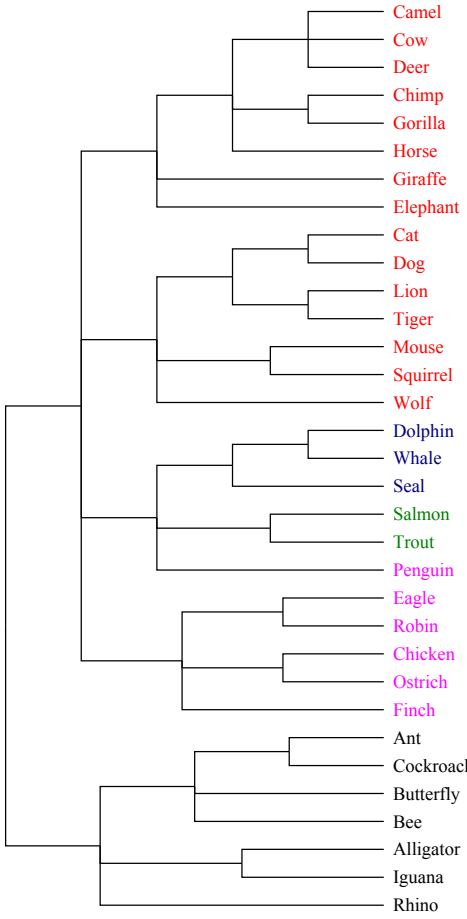


Fig. 12: PYDT structure learnt for the animals dataset of Tenenbaum and Kemp [2008], learnt using the EM algorithm.

relating to attributes (e.g. being warm-blooded, having two legs) of 33 animal species from Tenenbaum and Kemp [2008]. The MAP tree structure learnt using EM, as shown in Figure 12, is intuitive, with subtrees corresponding to land mammals, aquatic mammals, reptiles, birds, and insects (shown by colour coding). Penguins cluster with aquatic species rather than birds, which is not surprising since the data includes attributes such as "swims", "flies" and "lives in water".

9.4 Cancer cell line encyclopedia

The Cancer cell line encyclopedia (CCLE) consists of measurements of the sensitivity of 504 cancer derived cell lines to 24 drugs [Barretina et al., 2012]. Such data has the potential to help us understand the relationship between different cancer types in different tissues and the drugs' various mechanisms of action, and to aid in clinical practice. We use the PYDT slice sampling algorithm to hierarchically cluster the drugs according their sensitivity patterns across the cell lines, see Figure 13. Here we also show the known molecular inhibition targets of the drugs. We see that drugs close in the tree often have shared inhibition targets, for example PD_0325901 and AZD6244 are siblings in the tree and are both MAPK/ERK kinase (MEK) inhibitors. However, it is interesting that there are both drugs with shared inhibition targets that are distant in the tree, such as Nutlin_3 and 17_AAG, suggesting there maybe be biologically significant differences in their mechanism of action, and drugs with no known shared target such as Lapatinib and Erlotinib which are very close in the tree, suggesting their targets may be part of the same biological pathway. Such a hierarchical clustering could be used clinically, for example if using multiple drugs it could be beneficial to use distant drugs in the tree to maximise the diversity of the treatment and therefore the chance of one of the drugs being effective for the particular patient. Qualitatively, for clustering the drugs we find that the trees found using the PYDT are more consistent than with the DDT since where the ordering of divergence events is poorly determined by the data, the PYDT can simply use a higher order branch point. To quantitatively assess how well the PYDT models this data compared to the DDT we performed an imputation experiment where 10% of the cell lines were held out as test data. Repeating on 10 such random training/test splits the average predictive log likelihood was 0.08 ± 0.34 for the PYDT vs. -3.76 ± 0.84 for the DDT (in both cases we used slice sampling with $10,000$ iterations, discarding the first 5000 iterations as burnin).

We also use the CCLE data to assess the convergence properties of our two sampling methods: MH and slice sampling. We first consider hierarchically clustering the 504 cell lines. The marginal likelihood of the inferred trees under the PYDT during a run of MH and slice sampling, starting from a random initial tree is shown in Figure 14. We see that initially MH performs better, but then slice sampling overtakes it. Initially, MH performs well because the random tree structure means that sampling reattachment positions from the prior is not that unreasonable: indeed the initial 1000 proposals obtain an impressive acceptance rate of just over 0.3. However, as the tree structure improves, the probability of the MH proposal being reasonable relative to the current position becomes much smaller, and the acceptance ratio shrinks to around 0.02, so that significant computation is wasted. Slice sampling by contrast maintains roughly the same efficiency as the tree structure improves because its shrinkage procedure is able to rapidly rule out large portions of the tree as unpromising. Throughout the slice sampling run it takes only on average 6 rejections before a rettachment position is accepted. It is unclear exactly why the slice sampling method is less effective than MH early on, but we suspect it is because MH samples from the prior, so that proposing reattachment higher (closer to the root) is more likely than in the slice sampler, where all branches are treated equally, giving undue weight to the large number of the branches lower down in the tree. This could be remedied in the slice sampler, but we leave this to future work.

As with models such as Dirichlet process mixture (DPM) models, it is beneficial to know whether our

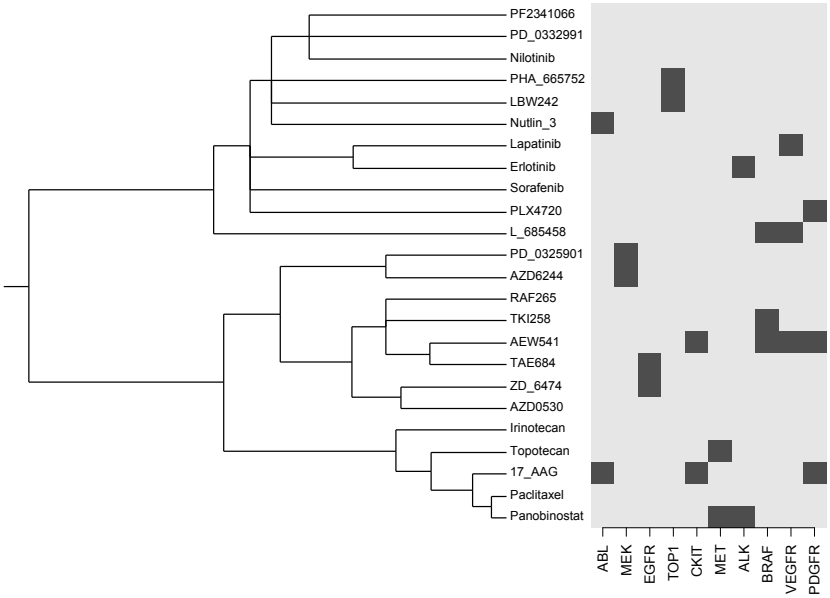


Fig. 13: Highest probability hierarchical clustering found using the slice sampling algorithm for the drugs in CCLE using the PYDT, along with known inhibition targets.

samplers are able to truly explore the space of possible configurations available to the model. One approach for DPMs is to run two MCMC chains: one from a configuration where every data point is in its own mixture component, and one where all datapoints are in a single cluster. One can then monitor how many iterations are required until both chains have a similar number of clusters. We perform an analogous experiment for the PYDT: one chain is initialised with a binary, random tree (sampled from the DDT), and the other chain with a flat clustering. We do this for both clustering the 24 drugs and the 504 cell lines, as shown in Figure 15. Encouragingly we see that even in the more challenging setting of hierarchically clustering the cell lines, the slice sampler appears to have burnt in after around 7 minutes, corresponding to around 2000 iterations. This means for a binary tree each subtree would only have been attached and reattached by the slice sampler only an average of twice (since a binary tree over 504 leaves has $504 + 503 = 1007$ potential subtrees). Typically at convergence the learnt tree has around 290 internal nodes (out of a possible maximum of 503 for a fully binary tree), with an average branching factor of around 2.5 (although the maximum branching factor varies in the range $8 - 12$), corresponding to a moderate but significant level of multifurcation.

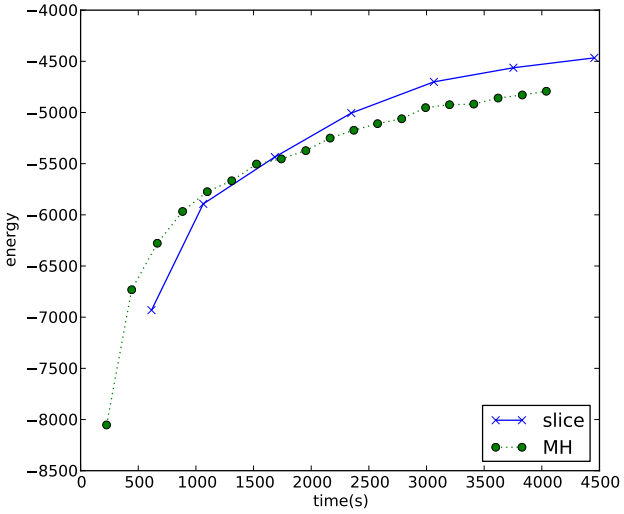
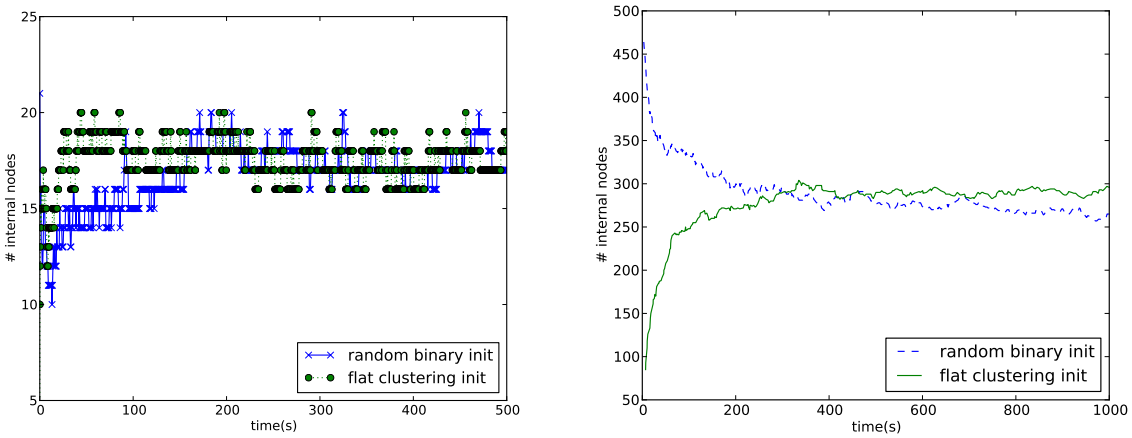


Fig. 14: Marginal likelihood of trees for hierarchically clustering the cell lines in CCLE under the PYDT, using MH and slice sampling.

10 Conclusion

We have introduced the Pitman-Yor Diffusion Tree, a Bayesian nonparametric prior over tree structures with arbitrary branching structure at each branch point. We have shown the PYDT defines an infinitely exchangeable distribution over data points. We demonstrated two alternative MCMC samplers and Bayesian EM with greedy search, all of which using message passing on the tree structure. In ongoing work we are investigating more advanced MCMC methods based on the uniformisation approach introduced by Rao and Teh [2011]. We are also interested in better understanding the underlying process conditional on which individual *paths* through the PYDT are iid. Informally such an object will be a continuously branching tree with branch weights. Relevant prior work includes Haas et al. [2008] who formally studied the convergence of discrete fragmentation trees to continuum trees [Aldous, 1991]. Quantitatively we have shown a modest improvement relative to the DDT



(a) Drugs, MH sampler

(b) Cell lines, slice sampler

Fig. 15: Running the samplers from two different initial states: a binary random clustering, or a flat clustering.

on a density estimation task. However, we see improved interpretability as the key benefit of removing the restriction to binary trees, especially since hierarchical clustering is typically used as a data exploration tool. Qualitatively, we have shown the PYDT can find simpler and more consistent representations of data than the DDT.

11 Acknowledgements

DAK would like to thank Wolfson College, Cambridge and Microsoft Research Cambridge for funding through the Roger Needham Scholarship. ZG would like to acknowledge EPSRC grants EP/I036575/1 and EP/H019472/1 and support from Google and Micrsoft. We thank Peter Orbanz and anonymous reviewers for help improving the manuscript.

References

R P Adams, I Murray, and D J C MacKay. The Gaussian process density sampler. In *Advances in Neural Information Processing Systems*, 2008.

R P Adams, Z Ghahramani, and M I Jordan. Tree-structured stick breaking for hierarchical data. In *Advances in Neural Information Processing Systems (NIPS) 23*, 2010.

D J Aldous. Exchangeability and related topics. In *Ecole d'Ete de Probabilities de Saint-Flour*, volume XIII, pages 1–198. Springer, 1983.

D J Aldous. The continuum random tree I. *The Annals of Probability*, pages 1–28, 1991.

D J Aldous. Probability distributions on cladograms. *Random discrete structures*, 76:1–18, 1996.

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

J Barretina, G Caponigro, N Stransky, K Venkatesan, A A Margolin, S Kim, C J Wilson, J Lehár, G V Kryukov, D Sonkin, et al. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–307, 2012.

D M Blei, T L Griffiths, M I Jordan, and J B Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. *Advances in Neural Information Processing Systems (NIPS)*, 2004.

D M Blei, T L Griffiths, and M I Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57 (2):7:1—-7:30, 2010.

C Blundell, Y W Teh, and K A Heller. Bayesian rose trees. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

L Boyles and M Welling. The time-marginalized coalescent prior for hierarchical clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

D Colless. Phylogenetics: the theory and practice of phylogenetic systematics. *Systematic Zoology*, 31(1): 100–104, 1982.

R O Duda, P E Hart, and D G Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2001.

B Eldon and J Wakeley. Coalescent processes when the distribution of offspring number among individuals is highly skewed. *Genetics*, 172(4), 2006.

J. Geweke. Getting it right. *Journal of the American Statistical Association*, 99(467):799–804, 2004.

I Guyon, S Gunn, A Ben-Hur, and G Dror. Result analysis of the NIPS 2003 feature selection challenge. *Advances in Neural Information Processing Systems (NIPS)*, 2005.

B Haas, G Miermont, J Pitman, and M Winkel. Continuum tree asymptotics of discrete fragmentations and applications to phylogenetic models. *The Annals of Probability*, 36(5):1790–1837, 2008.

D Hedgecock. Does variance in reproductive success limit effective population sizes of marine organisms? *Genetics and evolution of aquatic organisms*, 1994.

K A Heller and Z Ghahramani. Bayesian hierarchical clustering. In *22nd International Conference on Machine Learning (ICML)*, page 304, 2005.

E Hewitt and L J Savage. Symmetric measures on Cartesian products. *Transactions of the American Mathematical Society*, 80:470–501, 1955.

J F C Kingman. The coalescent. *Stochastic processes and their applications*, 13(3):235–248, 1982.

J F C Kingman. *Poisson processes*. Oxford University Press, USA, 1993.

D A Knowles and Z Ghahramani. Pitman-Yor diffusion trees. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011.

D A Knowles, J Van Gael, and Z Ghahramani. Message passing algorithms for Dirichlet diffusion trees. In *International Conference on Machine Learning (ICML)*, 2011.

D C Liu and J Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(3):503–528, 1989. ISSN 0025-5610.

D J C MacKay and T Broderick. Probabilities over trees: generalizations of the Dirichlet diffusion tree and Kingman's coalescent. Website, 2007. URL http://www.inference.phy.cam.ac.uk/mackay/trees/.

P McCullagh, J Pitman, and M Winkel. Gibbs fragmentation trees. *Bernoulli*, 14(4):988–1002, 2008.

T P Minka. Expectation propagation for approximate Bayesian inference. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001.

T P Minka, J M Winn, J P Guiver, and D A Knowles. Infer.NET 2.4, 2010.

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

R M Neal. Slice sampling. *The Annals of Statistics*, 31(3): 705–741, 2003a.

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

J Pitman. Coalescents with multiple collisions. *Annals of Probability*, pages 1870–1902, 1999.

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

P Rai and H Daumé III. The infinite hierarchical factor regression model. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

V A Rao and Y W Teh. Fast MCMC inference for Markov jump processes and continuous time Bayesian networks. In *Uncertainty in Artificial Intelligence (UAI)*, 2011.

C E Rasmussen. The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, 2000.

J S Rogers. Central moments and probability distributions of three measures of phylogenetic tree imbalance. *Systematic biology*, 45(1):99–110, 1996.

S Sagitov. The general coalescent with asynchronous mergers of ancestral lines. *Journal of Applied Probability*, 36(4):1116–1125, 1999.

J Steinhardt and Z Ghahramani. Flexible martingale priors for deep hierarchies. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

M Steinrücken, M Birkner, and J Blath. Analysis of DNA sequence variation within marine species using beta-coalescents. *Theoretical population biology*, 2012.

Y W Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics*, page 992. Association for Computational Linguistics, 2006.

Y W Teh, H Daumé III, and D M Roy. Bayesian agglomerative clustering with coalescents. *Advances in Neural Information Processing Systems*, 2008.

Y W Teh, C Blundell, and L T Elliott. Modelling genetic variations with fragmentation-coagulation processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

J B Tenenbaum and C Kemp. The discovery of structural form. In *Proceedings of the National Academy of Sciences*, volume 105, 2008.

C Williams. A MCMC approach to hierarchical mixture modelling. *Advances in Neural Information Processing Systems*, 2000.

David A. Knowles David A. Knowles is a postdoctoral researcher with Daphne Koller in the Computer Science Department at Stanford University. He did his PhD with Zoubin Ghahramani in the Machine Learning group of the Cambridge University Engineering Department, during which he worked part-time at Microsoft Research Cambridge developing Infer.NET. Prior to his PhD he obtained a masters in Bioinformatics and Systems Biology from Imperial College London. His undergraduate degree at the University of Cambridge comprised two years of Physics before switching to Engineering to complete an MEng with Professor Ghahramani. His research involves both the development of novel machine learning methods and their application to data analysis problems in biology.

Zoubin Ghahramani Zoubin Ghahramani is a Professor of Information Engineering at the University of Cambridge. He studied computer science and cognitive science at the University of Pennsylvania, obtained his PhD from MIT in 1995, and was a postdoctoral fellow at the University of Toronto. His academic career includes concurrent appointments at the Gatsby Computational Neuroscience Unit in London, and as a faculty member of CMU's Machine Learning Department for over 10 years. His current research focuses on nonparametric Bayesian modelling and statistical machine learning. He has over 200 publications in computer science, statistics, engineering, and neuroscience. He has served on the editorial boards of several leading journals in the field, including JMLR, JAIR, Annals of Statistics, Machine Learning, Bayesian Analysis, and was Associate Editor in Chief of IEEE Transactions on Pattern Analysis and Machine Intelligence. More information can be found at http://learning.eng.cam.ac.uk/zoubin/.