BAYESIAN STATISTICS 7, pp. 619–629
J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid,
D. Heckerman, A. F. M. Smith and M. West (Eds.)

Density Modeling and Clustering Using Dirichlet Diffusion Trees

RADFORD M. NEAL
University of Toronto, Canada
radford@stat.utoronto.ca

SUMMARY

I introduce a family of prior distributions over multivariate distributions, based on the use of a "Dirichlet diffusion tree" to generate exchangeable data sets. These priors can be viewed as generalizations of Dirichlet processes and of Dirichlet process mixtures, but unlike simple mixtures, they can capture the hierarchical structure present in many distributions, by means of the latent diffusion tree underlying the data. This latent tree also provides a hierarchical clustering of the data, which, unlike *ad hoc* clustering methods, comes with probabilistic indications of uncertainty. The relevance of each variable to the clustering can also be determined. Although Dirichlet diffusion trees are defined in terms of a continuous-time process, posterior inference involves only finite-dimensional quantities, allowing computation to be performed by reasonably efficient Markov chain Monte Carlo methods. The methods are demonstrated on problems of modeling a two-dimensional density and of clustering gene expression data.

Keywords: DENSITY MODELING; HIERARCHICAL CLUSTERING; PRIOR DISTRIBUTIONS.

1. INTRODUCTION

Unknown distributions are encountered when estimating the density of observed data and when modeling the distribution of random effects or other latent variables. Exploratory data analysis can also be viewed in terms of finding features of the data, such as clusters, that are useful in modeling its distribution. A Bayesian model involving an unknown distribution requires a prior distribution over distributions. For such a model to be useful in practice, the prior must be an adequate approximation to our actual prior beliefs about the unknown distribution, and it must be possible to compute the predictive distribution for new data with reasonable efficiency.

The Dirichlet process (Ferguson 1973) is a simple and computationally tractable prior for an unknown distribution. However, it produces distributions that are discrete with probability one, making it unsuitable for density modeling. This can be avoided by convolving the distribution with some continuous kernel, or more generally, by using a Dirichlet process to define a mixture distribution with infinitely many components, of some simple parametric form (Antoniak 1973; Ferguson 1983). Such Dirichlet process mixture models are not always ideal, however, because they use a prior distribution in which the parameters of one mixture component are independent of the parameters of other components. For many problems, we would expect instead that the components will be hierarchically organized, in ways analogous to the hierarchical grouping of organisms belonging to various species. Even if no obvious hierarchy is present, modeling a complex distribution by a mixture of simple distributions will require that each mode of the distribution be modeled using many of these simple mixture components, which will have similar parameters, and therefore form clusters themselves. Since Dirichlet process mixture models don't capture this hierarchical structure, inference using these models will be inefficient

— it will take more data than it should to force the model to create appropriate components, since the model does not "know" that these components will likely resemble other components.

The prior distribution over distributions I discuss here can be seen as a hierarchical generalization of Dirichlet process mixture models. The latent structure underlying the data for this model is what I call a "Dirichlet diffusion tree", whose terminal nodes (leaves) are the data points, and whose non-terminal nodes represent groupings of data points in a hierarchy. This latent tree structure generalizes the one-level grouping of data points into clusters that underlies a Dirichlet process mixture model. For some problems, this structure may be merely a device for obtaining a more suitable prior distribution over distributions. For exploratory applications, however, the latent diffusion tree structure may be of intrinsic interest, as it provides a hierarchical clustering of the data.

Most commonly-used hierarchical clustering methods are not based on any probabilistic model of the data. Williams (2000) discusses hierarchical Bayesian mixture models that are similar in objective to the models described here, but which have a finite (though unknown) number of components. Polya trees are another generalization of the Dirichlet process that produces distributions that have hierarchical structure, and which can be continuous. However, Polya tree priors have an unfortunate dependence on an arbitrary set of division points, at which discontinuities in the density functions occur. Walker, *et al*. (1999) review these and other related priors over distributions, including the "reinforced random walks" of Coppersmith and Diaconis, which resemble the diffusion trees discussed here, but differ in crucial respects.

Inference involving Dirichlet diffusion trees will not be as computationally easy as inference for simple Dirichlet process or Polya tree models, or as simple hierarchical clustering. Markov chain methods are needed to sample from the posterior distribution over trees. I describe here some techniques for constructing suitable Markov chain samplers, and demonstrate that they work well in a simple two-dimensional density modeling problem, and on a more difficult problem of clustering tumor cells based on gene expression data.

2. THE DIRICHLET DIFFUSION TREE PRIOR

I will define the Dirichlet diffusion tree prior over distributions by giving a procedure for randomly generating a data set of n points, each a vector of p real numbers, in which the data points are drawn independently from a common distribution drawn from the prior. The procedure generates these random data sets one point at time, with each point being drawn from its conditional distribution given the previously generated points, in a manner analogous to the "Polya urn" procedure for defining a Dirichlet process prior (Blackwell and MacQueen 1973). Such a procedure will be consistent with the data points being independently drawn from some unknown distribution as long as the distribution over data sets produced by the procedure is exchangeable, since such an exchangeable prior on data sets is equivalent to a prior over distributions by de Finetti's Representation Theorem (Bernardo and Smith 1994, Section 4.3).

2.1. *Generation of data points using a diffusion tree*

Each point in a data set drawn from the Dirichlet diffusion tree prior is generated by following a path of a diffusion process. Paths to different data points are linked in a tree structure, according to when each path diverges from previous paths. Generation of these paths will be described below by a sequential procedure, in which paths are generated for each data point in turn, but we will see in Section 2.3 that the ordering of the data points is in fact immaterial.

The first data point is generated by a Gaussian diffusion process (ie, Brownian motion), beginning at some origin, which I will here fix at zero. From this origin, the diffusion process operates for a predetermined length of time, which without loss of generality can be fixed at

one. If at time t, the process has reached the point $X_1(t)$, the point reached an infinitesimal time, dt, later will be $X_1(t+dt) = X_1(t) + N_1(t)$, where $N_1(t)$ is a Gaussian random variable with mean zero and covariance $\sigma^2 I dt$, were σ^2 is a parameter of the diffusion process. The $N_1(t)$ at different times are independent. The end point of the path, $X_1(1)$, is the sum of the infinitesimal increments $N_1(t)$, and is easily seen to have a Gaussian distribution with mean zero and covariance $\sigma^2 I$. This end point is the first point in the data set.

The second point in the data set is also generated by following a path from the origin. This path, $X_2(t)$, initially follows the path leading to the first point. However, the two paths will diverge at some time, T_d, after which the path to the second point is independent of the remainder of the first path. In other words, the infinitesimal increments for the second path, $N_2(t)$, are equal to the corresponding increments for the first path, $N_1(t)$, for $t < T_d$, but thereafter, $N_2(t)$ and $N_1(t)$ are independent. The distribution of the divergence time, T_d, can be expressed in terms of a "divergence function", $a(t)$. At each time t before divergence occurs, the probability that the path to the second data point will diverge from the path to the first data point within the next infinitesimal time period of duration dt is given by $a(t)dt$.

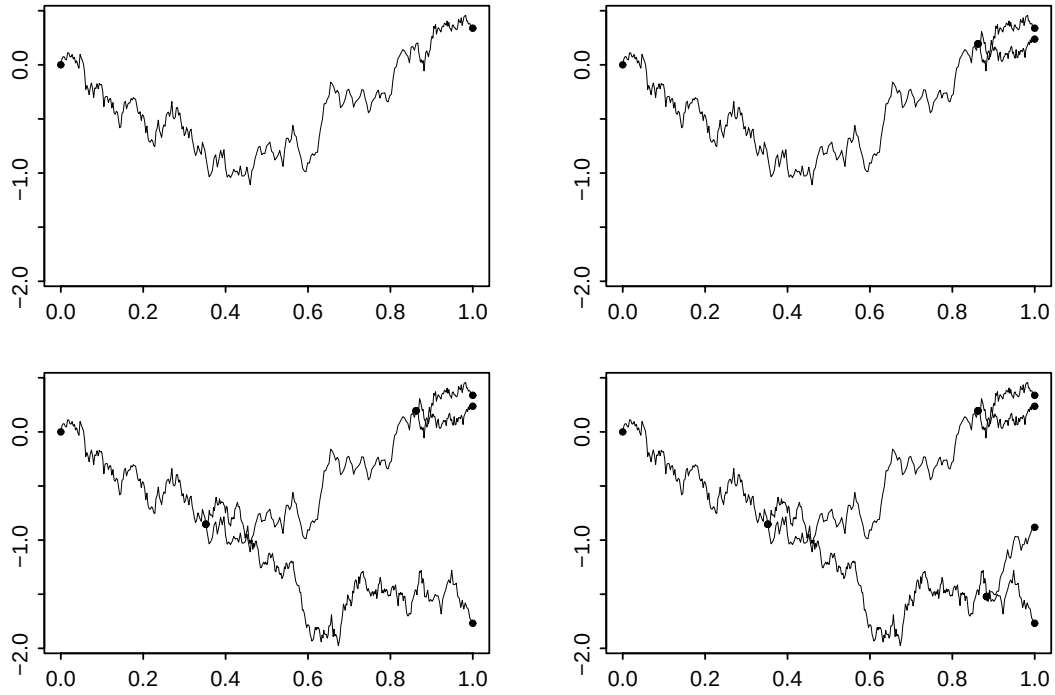

Figure 1. *Generation of a data set of four real numbers from the Dirichlet diffusion tree prior with* $\sigma = 1$ *and* $a(t) = 1/(1-t)$*. Diffusion time is on the horizontal axis, data values on the vertical axis. The upper-left plot shows the path to the first data point (0.34). The upper-right plot shows the path to the second data point (0.24) diverging from the first path at* $t = 0.86$*. In the lower-left plot, the path to the third data point (*-1.77*) diverges from the first two paths at* $t = 0.35$*. In the lower-right plot, the path to the fourth data point (*-0.87*) follows the path to the third data point until* $t = 0.88$*.*

In general, the ith point in the data set is obtained by following a path from the origin that initially coincides with the path to the previous $i-1$ data points. If the new path has not diverged at a time when paths to past data points diverged, the new path chooses between these past paths with probabilities proportional to the numbers of past paths that went each way — this reinforcement of previous events is characteristic of "Polya urn" schemes. If at time t, the new path is following a path traversed by m previous paths, the probability that it will diverge from this path within an infinitesimal interval of duration dt is $a(t)dt/m$ — note the division by m,

which is another aspect of reinforcement of past events. Once divergence occurs, the new path moves independently of previous paths.

Figure 1 illustrates the diffusion tree process for a data set of $n = 4$ data points, each a single real number (ie, $p = 1$).

2.2. *Probability of generating a given tree of data points*

The probability of obtaining a given data set along with its underlying tree can be expressed more easily if the details of the continuous paths taken are suppressed, leaving only the structure of the tree (ie, how data points are hierarchically grouped), the times at which paths diverged, and the locations of these divergence points and of the final data points. Figure 2 shows this less-detailed representation of the example shown in Figure 1.

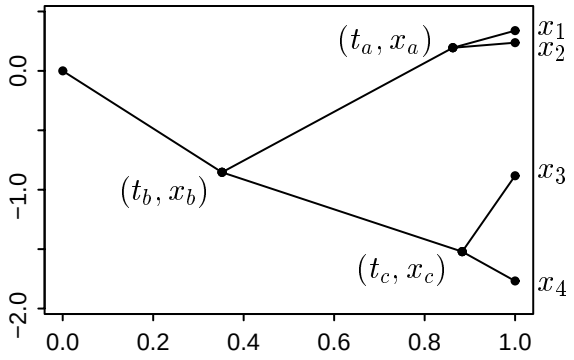

Figure 2. *The data x_1, x_2, x_3, and x_4 generated above and its underlying tree, with details of the paths suppressed except at the divergence points a, b, and c.*

The probability of obtaining a given tree and data set can be written as a product of two factors. The tree factor is the probability of obtaining the given tree structure and divergence times. The data factor is the probability of obtaining the given locations for divergence points and final data points, when the tree structure and divergence times are as given.

The tree factor can be found without reference to the locations of the points, since the divergence function depends only on t. The probability that a new path following a path previously traversed m times will not diverge between time s and time t can be found by dividing the time from s to t into k intervals of duration $(t-s)/k$, and letting k go to infinity:

$$P(\text{no divergence}) \;=\; \lim_{k\to\infty} \prod_{i=0}^{k-1} \Big(1 - a(s + i(t-s)/k)[(t-s)/k]/m\Big) \;=\; e^{(A(s)-A(t))/m}$$

Here, $A(t) = \int_0^t a(u)\,du$. Using this formula, the probability density for obtaining the tree structure and divergence times for the example in Figure 2 is found to be as follows:

$$e^{-A(t_a)}\; a(t_a) \;\times\; e^{-A(t_b)/2}\; (a(t_b)/2) \;\times\; e^{-A(t_b)/3}\; (1/3)\; e^{A(t_b)-A(t_c)}\; a(t_c)$$

Given the tree structure and divergence times, the data factor is just a product of Gaussian densities, since the distribution for the location of a point that has diffused for time d starting from location x is Gaussian with mean x and covariance $\sigma^2 Id$. Letting $\mathrm{N}(x \,|\, \mu, \sigma^2)$ be the Gaussian probability density function, we can write the data factor for the example in Figure 2 as follows:

$$\begin{aligned}\mathrm{N}(x_b \,|\, 0,\, \sigma^2 t_b) \times \mathrm{N}(x_a \,|\, x_b,\, \sigma^2(t_a - t_b)) \times \mathrm{N}(x_1 \,|\, x_a,\, \sigma^2(1-t_a)) \times \mathrm{N}(x_2 \,|\, x_a,\, \sigma^2(1-t_a)) \\ \times\, \mathrm{N}(x_c \,|\, x_b,\, \sigma^2(t_c - t_b)) \times \mathrm{N}(x_3 \,|\, x_c,\, \sigma^2(1-t_c)) \times \mathrm{N}(x_4 \,|\, x_c,\, \sigma^2(1-t_c))\end{aligned}$$

2.3. *Proof of exchangeability*

To show that the procedure for generating a data set using a Dirichlet diffusion tree defines a valid prior over distributions, we must show that the probability density for a data set does not change when the order of the data points is permuted — ie, that the prior is "exchangeable". I will show this by proving the stronger property that the probability density for producing a data set along with its underlying tree structure and the times and locations of the divergence points is the same for any ordering of data points. Exchangeability follows from this by summing over all possible tree structures and integrating over times and locations of divergences.

The probability density for a data set along with an underlying tree can be written as a product of factors, each pertaining to one segment of the tree. (For example, the tree in Figure 2 has seven segments.) For each segment, $(t_u, x_u) - (t_v, x_v)$, there is a factor in the probability density that corresponds to the density for the diffusion process starting at x_u to move to x_v in time $t_v - t_u$, which is $\mathrm{N}(x_v \mid x_u,\, \sigma^2 I(t_v - t_u))$. The product of these factors is the overall data factor in the density. Since the set of segments making up the tree does not depend on the order of the data points, neither will this data factor.

A segment of the tree traversed by more than one path will also be associated with a factor in the overall probability density pertaining to the lack of divergence of these paths before the end of the segment. If such a segment ends before $t = 1$, there will be another a factor for the probability density for one path to diverge at the end of this segment, along with factors for the probabilities of later paths taking the branches they did. The product of such factors for all segments is the overall factor relating to the tree structure. We need to show that this factor does not actually depend on the ordering of the data points, even though it is expressed in an order-dependent way above.

Consider a segment, $(t_u, x_u) - (t_v, x_v)$, that was traversed by $m > 1$ paths. The probability that the $m-1$ paths after the first do not diverge before t_v is $\prod_{i=1}^{m-1} e^{(A(t_u) - A(t_v))/i}$, which does not depend on the order of the data points. If $t_v = 1$, this is the whole factor for this segment. Otherwise, suppose that the first $i-1$ paths do not diverge at t_v, but that path i does diverge at t_v. (Note that i will be at least two, and that some path must diverge at t_v, as otherwise the segment would not end at that time.) The probability density for this divergence is $a(t_v)/(i-1)$. Subsequent paths take one or the other branch at this divergence point, with probabilities proportional to the number that have gone each way previously. Suppose that n_1 paths in total go the way of the first path, and n_2 go the way of path i. (Note that $n_1 \geq i-1$, and $n_1 + n_2 = m$.) The probability that path j (with $j > i$) goes the way of the first path will be $c_1/(j-1)$, where c_1 is the number of paths that went that way before, which will vary from $i-1$ to $n_1 - 1$. The probability that path j goes the other way will be $c_2/(j-1)$, where c_2 is the number of paths that went the other way before, which will vary from 1 to $n_2 - 1$. The product of these branching probabilities for all $j > i$ will be

$$\prod_{c_1=i-1}^{n_1-1} c_1 \cdot \prod_{c_2=1}^{n_2-1} c_2 \Big/ \prod_{j=i+1}^{m} (j-1) \;=\; \frac{(n_1-1)!}{(i-2)!} \cdot (n_2-1)! \cdot \frac{(i-1)!}{(m-1)!}$$

$$= \; (i-1) \cdot \frac{(n_1-1)!\,(n_2-1)!}{(m-1)!}$$

When this is multiplied by the probability density of $a(t_v)/(i-1)$ for path i diverging at time t_v, the two factors of $i-1$ cancel, leaving a result that does not depend on i, and hence is independent of the order of the data points. This argument can be modified to handle the case where $a(t)$ has an infinite peak, allowing more than one path to diverge at the same time.

3. PROPERTIES OF DIRICHLET DIFFUSION TREE PRIORS

The properties of a Dirichlet diffusion tree prior vary with the choice of divergence function, $a(t)$. I will investigate these properties here by looking at data sets generated from these priors. I also investigate when the distributions produced are continuous and absolutely continuous. More details on the properties of Dirichlet diffusion tree priors are found in (Neal 2001).

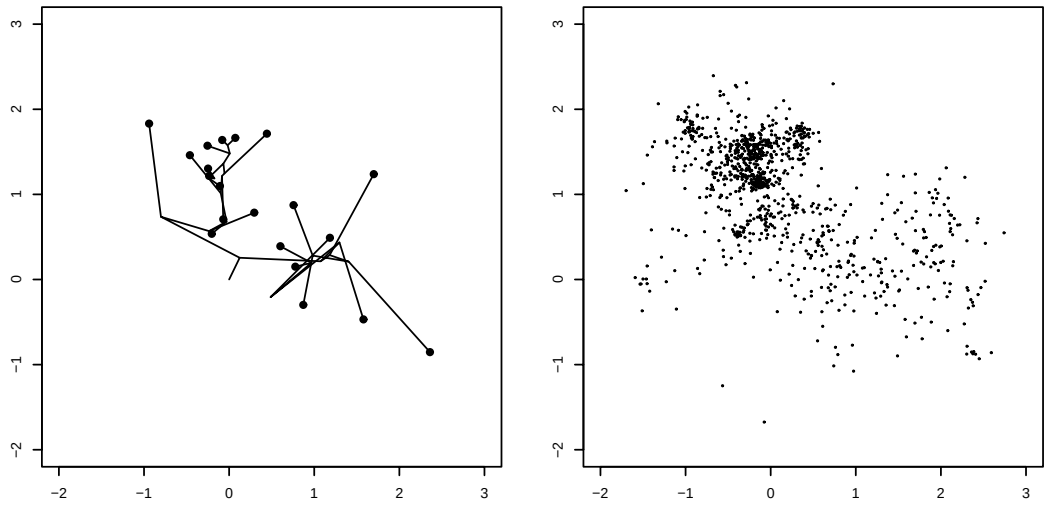

Figure 3. *Generation of a two-dimensional data set from the Dirichlet diffusion tree prior with* $\sigma = 1$ *and* $a(t) = 1/(1-t)$*. The plot on the left shows the first twenty data points generated, along with the underlying tree structure. The right plot shows 1000 data points obtained by continuing the procedure beyond these twenty points.*

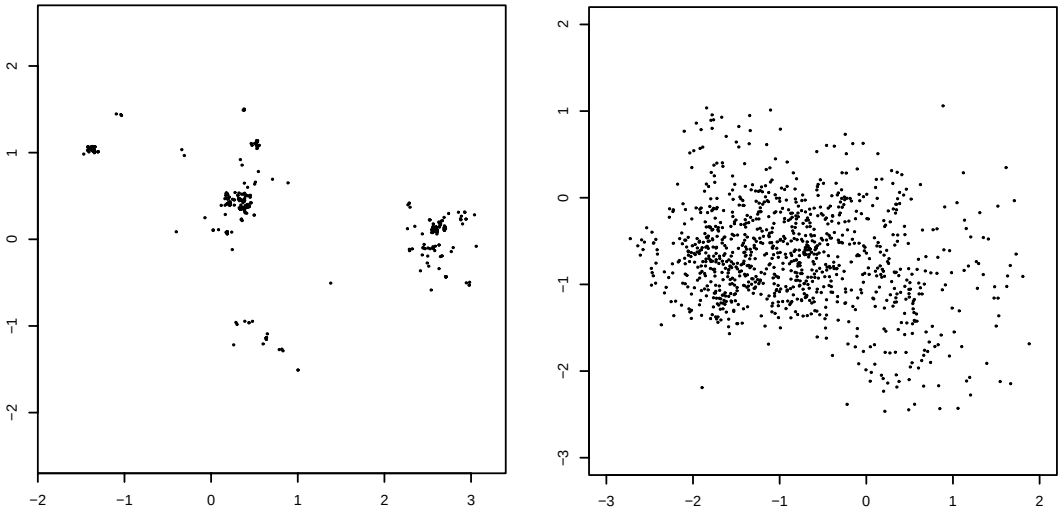

Figure 4. *Two data sets of 1000 points drawn from Dirichlet diffusion tree priors with* $\sigma = 1$*. For the data set on the left, the divergence function used was* $a(t) = (1/4)/(1-t)$*. For the data set on the right,* $a(t) = (3/2)/(1-t)$*.*

Divergence functions of the form $a(t) = c/(1-t)$ have integrals that diverge only logarithmically as $t \to 1$: $A(t) = \int_0^t a(u)\,du = -c\log(1-t)$. Distributions drawn from such a prior will be continuous — ie, the probability that two data points will be identical is zero.

These distributions will, however, show some degree of tight clustering, varying with c, due to the possibility that divergence will not occur until quite close to $t = 1$. Figure 3 shows the generation of a two-dimensional data set of 1000 points, drawn from this prior with $c = 1$. We see here the hierarchical structure that the Dirichlet diffusion tree prior can produce — there are not just multiple modes in this data, but further structure within each mode. This hierarchy is more obvious in the data set shown on the left of Figure 4, produced from the prior with $c = 1/4$, which has tighter clusters. As seen in the right of Figure 4, when $c = 3/2$, the distributions generated are smoother, and do not exhibit any obvious hierarchical structure.

Priors with different characteristics can be obtained using a divergence function of the form $a(t) = b + d/(1-t)^2$. This divergence function can produce well-separated clusters that exhibit a clear hierarchical structure, but with the points within each cluster being smoothly distributed, due to the rapid increase in $a(t)$ as t approaches one.

The distributions produced by a Dirichlet diffusion tree prior will be continuous (with probability one) when the divergence function, $a(t)$, is such that $A(t) = \int_0^1 a(t)\,dt$ is infinite. However, it does not follow that distributions drawn from such a prior will be absolutely continuous, which is what is required for them to have density functions. I have investigated empirically when Dirichlet diffusion tree priors produce absolutely continuous distributions by looking at distances to nearest neighbors in a large sample from the distribution. For a distribution with a continuous density function, the density in the region near where the two nearest neighbors of a data point are located will be approximately constant. From this one can derive that if r is the ratio of the distance to the nearest other data vector divided by the distance to the second-nearest other data vector, the distribution of r^p will be uniform over $(0, 1)$. The empirical distribution of r^p for a large sample therefore provides evidence of whether or not the distribution from which the sample came is absolutely continuous.

The empirical results I have obtained (Neal 2001) lead me to conjecture that Dirichlet diffusion tree priors for p-dimensional distributions with $a(t) = c/(1-t)$ produce distributions that are not absolutely continuous when $c < p/2$, but which are absolutely continuous when $c > p/2$. I conjecture that, in contrast, all Dirichlet diffusion tree priors with divergence functions of the form $a(t) = b + d/(1-t)^2$ with $d > 0$ produce absolutely continuous distributions.

4. MODELS BASED ON DIRICHLET DIFFUSION TREES

In practice, the smoothness and other characteristics of an unknown distribution will seldom be known exactly, and it will therefore be appropriate to give higher-level prior distributions to the parameters of the divergence function, allowing these characteristics to be inferred from the data. The variance of the diffusion process for a variable would also usually be a hyperparameter, which can adapt to the scale of that variable; different variables might well be given different variances.

If the data were observed with some amount of noise, or the data values were rounded, it would be appropriate to regard the Dirichlet diffusion tree as defining the prior for the distribution of the unrounded, noise-free values, not for the data actually observed. For some problems, a heavy-tailed noise distribution may be appropriate, to prevent outliers from having an undue effect on the clustering. When the data is categorical, unobserved latent values can be introduced that determine the probabilities of the observed data (eg, via a logistic model). The distribution of these vectors of latent values could then be given a Dirichlet diffusion tree prior, indirectly defining a prior for the joint distribution of the categorical values.

A model in which the amount of noise for each variable is controlled by a hyperparameter may be useful even when there actually is no appreciable noise in the measurements, since with such a model, some or all of the variation of some variables can be explained as "noise" that is

unrelated to variation in other variables. If the posterior distribution for the noise hyperparameter for a variable is concentrated near a large value, the model will have effectively learned that this variable is not relevant to the clustering of data items.

5. MARKOV CHAIN SAMPLING FOR DIRICHLET DIFFUSION TREE MODELS

The finite representation of a Dirichlet diffusion tree shown in Figure 2 is used for Markov chain Monte Carlo computations. The state of the Markov chain used to sample from the posterior distribution will consist of at least the *structure* of the tree (the hierarchical organization of data points) and the *divergence times* for non-terminal nodes. Depending on the model and sampling method used, the Markov chain state may also include the *locations* of non-terminal nodes, *latent vectors* underlying the data (eg, the noise-free values of observations), and *hyperparameters* such as diffusion variances, noise levels, and parameters of the divergence function.

For models of real data without noise, the locations of terminal nodes are fixed at the data points. For models with latent vectors, the terminal nodes are fixed to these latent vectors, if they are represented explicitly. Neither latent vectors nor terminal node locations need be represented explicitly when the data is modeled as having Gaussian noise, since their distributions conditional on the structure of the tree and the divergence times will be Gaussian, and hence easily integrated over. The locations of non-terminal nodes can also be integrated over, since their joint distribution given the data or latent vectors is Gaussian. This can be done in time proportional to the number of data points, by exploiting the tree structure, as is done in a similar context by Williams (2000). If non-terminal node locations are instead kept as part of the Markov chain state, it takes only linear time to draw new values for them from their joint distribution (conditional on the tree structure, hyperparameters, and latent vectors).

It is easy to sample for the diffusion and noise variance hyperparameters when divergence times and locations for all nodes are known. Node locations can be temporarily sampled for this purpose if they are not being retained in the state. Metropolis updates could be done for the parameters of the divergence function, though I instead use slice sampling (Neal, 2003).

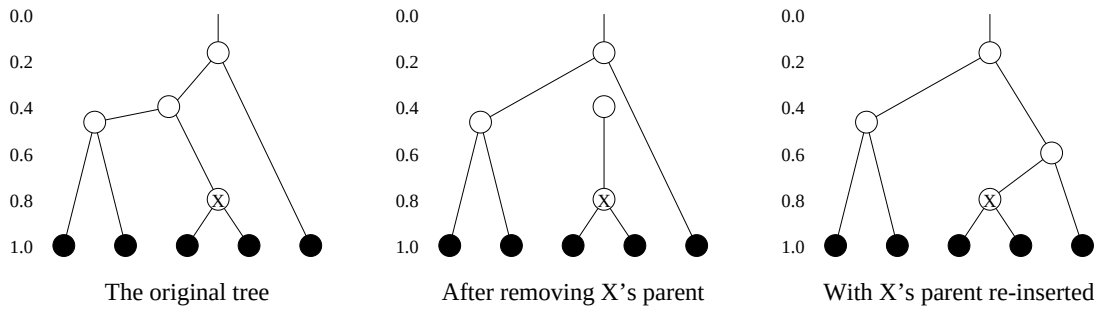

Figure 5. *Modifying a tree with a parent move. Divergence time is shown on the vertical axis, location on the horizontal axis. Black circles are the terminal nodes.*

The most crucial issue is how to sample for different tree structures, a problem that is similar to sampling for phylogenetic trees (eg, see Mau, *et al.* 1999). Figure 5 illustrates one approach that I have used, based on Metropolis-Hastings updates that propose to move the parent of a randomly chosen terminal or non-terminal node to a new location in the tree. After the parent of the chosen node is temporarily removed from the tree, a new position for it in the tree is found by simulating the data generation process described in Section 2.1, up to the time where the new path diverges from previous paths. (If this time is later than the child node's divergence time, the process is repeated.) This proposed position in the tree is then accepted or rejected based on the likelihood, integrating over any missing node locations. It is important that at least the location

of the parent node being moved be integrated over. This is easily done, and if non-terminal node locations are being retained, a new location for the parent can easily be sampled after the update (whether the proposal was accepted or not). If non-terminal node locations are not being retained, the change in likelihood resulting from the proposed move can be computed in time proportional to the depth of the tree, which typically grows logarithmically with the number of data points. I have also tried another approach to modifying the tree structure, in which a non-terminal node's position in the tree is updated by slice sampling. Best results are obtained when both approaches are combined.

The models and Markov chain sampling methods described above are implemented as part of my software for "flexible Bayesian modeling", which is available from my web page, `http://www.cs.utoronto.ca/~radford`. This software was used for the two examples that follow.

6. MODELING A TWO-DIMENSIONAL DENSITY

I tested the ability of Dirichlet diffusion trees to model a bivariate distribution using an artificial data set, generated by a somewhat complex procedure that produces an absolutely continuous distribution that is not of any simple form. (For details of the distribution and of the models and sampling procedures, see the software documentation, in which this is an example.) The data set of 500 points is shown on the left of Figure 6, along with a simple kernel density estimate, produced by the `kde2d` function in R's MASS library, with default parameter settings.

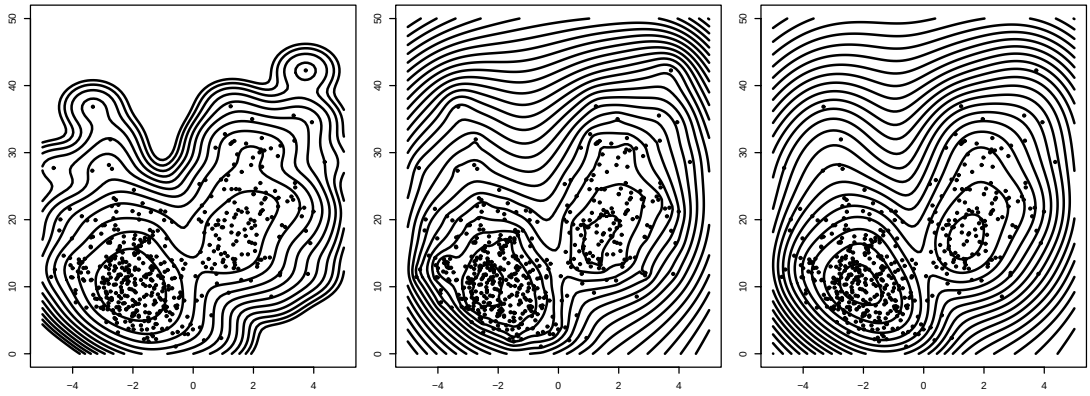

Figure 6. *The two-dimensional density modeling example. The plots show the data points along with contours of the natural log of the estimated density, spaced 0.5 apart. The left plot shows a simple kernel density estimate; the middle and right plots show Dirichlet diffusion tree estimates with different divergence functions.*

This data was modeled directly (with no noise) using two Dirichlet diffusion tree models, with divergence functions of $a(t) = c/(1-t)$ and $a(t) = b + d/(1-t)^2$. For both models, the two variables had separate hyperparameters controlling diffusion variances, which were given fairly vague prior distributions. The parameters of the divergence function (c for the first model, b and d for the second) were also given fairly vague priors.

The middle plot in Figure 6 shows the density estimate found using $a(t) = c/(1-t)$. This estimate was computed by simulating the generation of a 501st data point from 75 trees taken from the posterior distribution. For each of these trees, 500 paths were simulated that would end at each of the 500 data points if no divergence were to occur, with the time and location where divergence did occur defining a Gaussian distribution for the generated point. (This stratified sampling scheme improves the approximation, compared to generating 500 independent paths.) The final density estimate is an equal mixture of these 75×500 Gaussian distributions.

The posterior distribution of c for this model had mean 1.5 and standard deviation 0.2, so according to my conjecture, the distribution produced will be absolutely continuous. The actual distribution from which the data was drawn is indeed absolutely continuous, but with a density that is smoother than the estimate found with this divergence function. Even when $c \approx 1.5$ (as in the right of Figure 4), the prior with $a(t) = c/(1-t)$ favours distributions with some small-scale clustering, producing peaks in the density estimate around each data point.

The right plot in Figure 6 shows a density estimate found in the same way using a Dirchlet diffusion tree with $a(t) = b + d/(1-t)^2$. It displays a smoothness that is more appropriate for this data.

7. CLUSTERING GENE EXPRESSION DATA

I have also applied Dirichlet diffusion tree models to data on gene expression in leukemia cells gathered by Golub, *et al.* (1999). This data contains levels of expression for 3571 genes in the tumors of 72 leukemia patients. (Expression levels for more genes were measured, but were discarded as being unreliable. The remaining data was preprocessed to avoid spurious variation, and scaled so the variance of each variable was one.) Each tumor was classified as one of three types — AML, ALL-B, or ALL-T — based on data other than gene expression. The aim of this test is to see whether by clustering the gene expression data we can rediscover these three known types.

I first tried modeling this this data using a subset of only 200 genes, randomly selected from the total of 3571. The expression levels of these genes were modeled as having Gaussian noise, with the noise-free values generated from a Dirichlet diffusion tree with a divergence function of the form $a(t) = b + d/(1-t)^2$. Separate noise and diffusion variances were given to each gene. All hyperparameters were given fairly vague prior distributions.

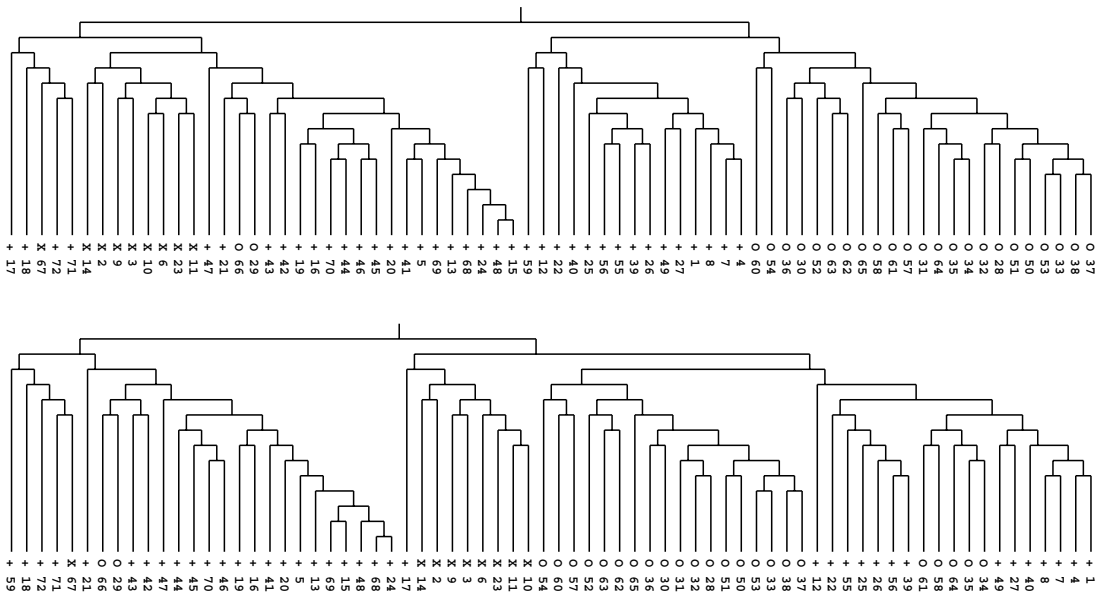

Figure 7. *Two hierarchical clusterings of the 72 tumors, drawn from the posterior distribution. Tumor types are shown by* $\mathrm{o} = \mathrm{AML}$, $+ = \mathrm{ALL\text{-}B}$, *and* $\mathrm{x} = \mathrm{ALL\text{-}T}$.

I ran two replicated Markov chain simulations, each taking approximately 90 minutes on a 1.7 GHz Pentium 4 processor. The state for these Markov chains did not include the locations of non-terminal nodes; these were instead integrated over. The two chains produced consistent results. Both appear to have converged after about one third of the run, judging by trace plots

of hyperparameters. I also ran chains in which non-terminal node locations were explicitly represented. This sped up tree updates by a factor of eight, but reduced their acceptance rate, with the result that efficiency was comparable to when these locations were integrated over.

Figure 7 shows dendrograms derived from two diffusion trees sampled from the posterior distribution. The clustering found is mostly consistent with the pre-existing classification, showing that the Dirichlet diffusion tree method produces useful results on problems of this scale. The posterior means of the diffusion and noise standard deviations for different genes varied by a factor of five, showing that the model has discovered that some genes are more relevant to the clustering than others.

When all 3571 genes are used, each Markov chain update takes much longer (increasing in proportion to the number of genes), and the chain shows some signs of not completely converging. The results produced are reasonable, but correspond to the pre-existing classification less closely than was the case using only 200 genes. I am presently investigating whether annealing methods can improve convergence on this very high dimensional problem.

ACKNOWLEDGEMENTS

I thank Sandrine Dudoit for providing the program for preprocessing the gene expression data. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Statist.* **2**, 1152–1174.

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*, Chichester: Wiley.

Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via Pólya urn schemes. *Ann. Statist.* **1**, 353–355.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *Ann. Statist.* **1**, 209–230.

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. *Recent Advances in Statistics* (H. Rizvi and J. Rustagi, eds.), New York: Academic Press.

Golub, T. R., Slonim, D. K., Tamahyo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. and Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**, 531–537.

Mau, B., Newton, M. A., and Larget, B. (1999). Bayesian phylogenetic inference via Markov chain Monte Carlo methods. *Biometrics* **55**, 1–12.

Neal, R. M. (2001). Defining priors for distributions using Dirichlet diffusion trees. *Tech. Rep.*, University of Toronto, Canada.

Neal, R. M. (2003). Slice sampling. *Ann. Statist.* (to appear, with discussion).

Walker, S. G., Damien, P., Purushottam, W. L., and Smith, A. F. M. (1999). Bayesian nonparametric inference for random distributions and related functions. *J. Roy. Statist. Soc. B* **61**, 485–527, (with discussion).

Williams, C. K. I. (2000). A MCMC approach to hierarchical mixture modelling. *Advances in Neural Information Processing Systems 12* (S. A. Solla, *et al.*, eds.). Cambridge, MA: The MIT Press