1

Discovering Non-binary Hierarchical Structures with Bayesian Rose Trees

Charles Blundell,[†] Yee Whye Teh,[†] and Katherine A. Heller[§]

[†]*Gatsby Unit, UCL, UK.*
[§]*Department of Engineering, University of Cambridge, UK.*

1.1 Introduction

Rich hierarchical structures are common across many disciplines, making the discovery of hierarchies a fundamental exploratory data analysis and unsupervised learning problem. Applications with natural hierarchical structure include topic hierarchies in text (Blei et al. 2010), phylogenies in evolutionary biology (Felsenstein 2003), hierarchical community structures in social networks (Girvan and Newman 2002), and psychological taxonomies (Rosch et al. 1976).

A large variety of models and algorithms for discovering hierarchical structures have been proposed. These range from the traditional linkage algorithms based on distance metrics between data items (Duda and Hart 1973), to maximum parsimony and maximum likelihood methods in phylogenetics (Felsenstein 2003), to fully Bayesian approaches that compute posterior distributions over hierarchical structures (e.g. Neal 2003). We will review some of these in Section 1.2.

A common feature of many of these methods is that their hypothesis spaces are restricted to binary trees, where each internal node in the hierarchical structure has exactly two children. This restriction is reasonable under certain circumstances, and is a natural output of the popular agglomerative approaches to discovering hierarchies, where each step involves the merger of two clusters of data items into one. However, we believe that there are good reasons why restricting to binary trees is often undesirable. Firstly, we simply do not believe that many hierarchies in real world applications are binary trees. Secondly, limiting the hypothesis space to binary trees often forces spurious structure to be "hallucinated" even if this structure

is not supported by data, making the practitioners task of interpreting the hierarchy more difficult. Finally, this spurious structure is also undesirable from an Occam's razor point of view: the methods are not returning the simplest hierarchical structure supported by the data, because the simpler structures which explain the data can involve non-binary trees and these are excluded from the hypothesis space.

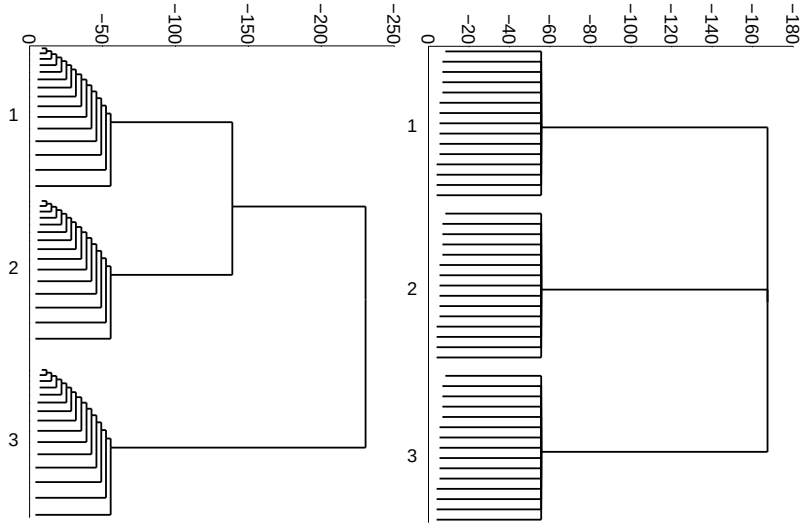


Figure 1.1 Bayesian hierarchical clustering (left) and Bayesian rose trees (right) on the same synthetic data set. The three groups of 15 similar data items all cluster into groups under both models.

Figure 1.1(left) shows an example whereby spurious structure is imposed on data by a model that assumes binary trees. In this case the model is Bayesian hierarchical clustering (BHC) (Heller and Ghahramani 2005), a probabilistic model for binary hierarchical clustering. The data set consists of three clusters of data items, each of which has no further internal substructure. Nevertheless, BHC produced cascades of internal nodes to represent each of the clusters. This is an unnecessarily complex structure for the simple clusters in the data set, and is a telltale sign among probabilistic binary hierarchical clustering algorithms that the tree is not representing large clusters in the data properly. Ideally the tree structure should be simplified by collapsing each cascade into a single node with many children, as in Figure 1.1(right), expressing the lack of substructure among the data items.

In this chapter we describe a probabilistic model for hierarchical structure discovery that operates in a broadened hypothesis space. Here each internal node can have an arbitrary number of children, allowing the model to use simpler, non-binary, trees to describe data if they do so better than binary trees. To help us choose among different trees, we will take a Bayesian model selection approach and choose the tree with highest marginal data likelihood. Part of the contribution of this paper is the design of a likelihood that conforms to our intuitions, such that structures with higher likelihoods also tend to be the ones that are subjectively simpler. We refer to trees in this broadened hypothesis space as *rose trees*, as

they are known in the functional programming literature (Bird 1998; Meertens 1988), and our model as *Bayesian rose tree mixture models*. The non-binary tree given in Figure 1.1(right) is in fact the structure discovered by our model.

We take a Bayesian approach to discovering hierarchy structure. At each step in constructing the tree, we perform a Bayesian hypothesis test for each possible merge. The best merge is then greedily selected. This results in the discovery of a single tree structure, analogous to in Heller and Ghahramani (2005), as opposed to a fully Bayesian approach, where a prior over trees is defined and the posterior approximated using Monte Carlo sampling (e.g. Felsenstein 2003; Neal 2003). A fully Bayesian approach, while in many ways appealing, is computationally very expensive and complex to implement due to the very large (super-exponential) number of trees and the complex Metropolis-Hastings moves that are often necessary for Markov chain Monte Carlo methods to mix properly over the posterior. A deterministic, single solution may also make interpreting the results easier for those modelers who are not very familiar with Bayesian methodology.

Therefore in this chapter we opt for a greedy approach, constructing a tree in an agglomerative bottom-up fashion. This gives an efficient algorithm that we find works well empirically.

The remainder of this chapter is organised as follows: In Section 1.2 we briefly review the existing literature on probabilistic hierarchical structure discovery and place Bayesian rose trees within this context. In Section 1.3 we describe our model in detail. In Section 1.4 we describe our greedy agglomerative construction algorithm. In Section 1.5 we discuss relationships with variants and other plausible extensions to BHC. Finally, in Section 1.6 we report experimental results using Bayesian rose trees, and conclude in Section 1.7.

1.2 Prior work

There is a very diverse range of methods for hierarchical structure discovery, and unfortunately it is not possible to review every contribution here. Most methods for hierarchical structure discovery can be construed as methods for hierarchical clustering, where each subtree corresponds to a cluster of data items. Classic introductions to clustering can be found in Hartigan (1975), McLachlan and Basford (1988) and Kaufman and Rousseeuw (1990), while Jain et al. (1999) is a more recent survey and Murtagh (1983) is a survey of classic methods for hierarchical clustering.

The most popular methods for hierarchical clustering are probably the agglomerative linkage methods (Duda and Hart 1973). These start with each data item in its own cluster and iteratively merge the closest pair of clusters together, as determined by some distance metric, until all data belong to a single cluster. A record is kept of the order of merges and used to form a hierarchy, where data items reside on the leaves, and branch lengths correspond to distances between clusters. Different methods are determined by different distance metrics among data items, and how these distances are combined define distances between clusters. Popular linkage methods include single, complete and average linkage, where the distance between two clusters is defined to be the minimum, maximum and average inter-cluster data item distances respectively. Whilst straightforward and computationally efficient, linkage methods are not model-based, making comparisons between the discovered hierarchies based on different distances difficult due to a lack of an objective criteria.

Another important area for hierarchical structure discovery is phylogenetics, where the

problem is to infer the phylogenetic tree relating multiple species, and where a rich variety of approaches have been explored (Felsenstein 2003). These include non-model-based methods, e.g. linkage algorithms based on distances among species (Fitch and Margoliash 1967; Saitou and Nei 1987; Studier and Keppler 1988) and parsimony-based methods (Camin and Sokal 1965), as well as model-based maximum likelihood methods Felsenstein (1973, 1981). Consistency has been shown of maximum likelihood estimators (Rogers 1997; Yang 1994). Another approach taken in phylogenetics is the Bayesian one, where a prior over phylogenetic trees is defined and the posterior distribution over trees estimated (Huelsenbeck and Ronquist 2001; Yang and Rannala 1997). This ensures that uncertainty in inferred hierarchical structures is accounted for, but is significantly more complex and computationally expensive.

Similar to model-based phylogenetics, recent machine learning approaches to model-based hierarchical structure discovery have a dichotomy between maximum likelihood estimation (Friedman 2003; Heller and Ghahramani 2005; Segal et al. 2002; Vaithyanathan and Dom 2000) and Bayesian posterior inference (Kemp et al. 2004; Neal 2003; Roy et al. 2007; Teh et al. 2008; Williams 2000), reflecting the trade-off between simplicity and computational efficiency on the one hand, and knowledge of structural uncertainty on the other. The approach taken in this paper is a direct extension of those in Friedman (2003) and Heller and Ghahramani (2005); we will discuss these in more detail in Section 1.5.

There are few hierarchical clustering methods that directly produce non-binary hierarchies. Williams (2000) fixes the maximum number of layers and nodes per layer, and defines a prior over trees whereby each node picks a node in the layer above independently. Blei et al. (2010) uses a nested Chinese restaurant process to define a prior over layered trees. Both are Bayesian methods which are quite computationally complex, and use Monte Carlo sampling for inference. This is to be expected since with non-binary hierarchies the number of internal nodes inferred can vary, and methods that cannot account for varying numbers of parameters can easily overfit. In methods that infer branch lengths as well as binary tree structures, non-binary hierarchies can be obtained by visual inspection and by heuristics for collapsing short branches.

1.3 Rose trees, partitions and mixtures

In this section we describe our probabilistic model, as well as the terminology used in the subsequent sections. We shall refer to the hierarchical clustering structure describing a data set as a rose tree, and to the probabilistic model based on a rose tree as a Bayesian rose tree mixture model, or BRT for short. Figure 1.2 gives two examples of rose trees over data items labelled a—e, one with all binary internal nodes, and one with a ternary node. We will use these as running examples throughout this section. Leaves correspond to data items and every node of a tree corresponds to a cluster of its leaves, with nodes higher up the tree corresponding to larger clusters.

Let $\mathcal{D}$ be a set of data items. Our probabilistic model for these data items $\mathcal{D}$ under a rose tree T, $p(\mathcal{D}|T)$, is a mixture model where each mixture component is a partition of the data set, which is in turn a disjoint set of clusters of data items. The role of the rose tree T is as a model index dictating which partitions of the data set are included in the mixture model; in particular, the partitions that are included are those that are "consistent" with the rose tree T. In the following, we will elaborate on the key concepts of clusters, partitions and rose trees, and how each of these are modelled in our probabilistic model.

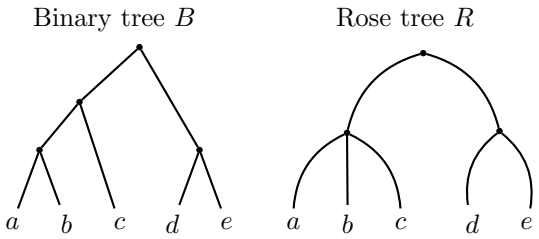


Figure 1.2 Examples of rose trees with five leaves labelled a—e. Left: a rose tree with only binary internal nodes and a cascading subtree (on leaves a, b, and c). Right: a rose tree with a ternary node resulting from collapsing the cascading subtree.

Definition 1.3.1 *A* rose tree T *either consists of a single leaf $x \in \mathcal{D}$, or consists of a root node along with n_T children, say $T_1, \ldots, T_{n_T}$, each of which are rose trees whose leaves are disjoint. We write children(T) for the set of* children *of T, and pa(T_i) for the* parent *of T_i. A* node *or* subtree*, identified by its root, T', of a rose tree T is either T or one of its descendants under the child relation. The* ancestors *of T', ancestor(T'), consists of the nodes of T that has T' as a descendant. The leaves leaves(T') of a node T' is the set of data items that are descendants of T'.*

For example, the tree in Figure 1.2(right) consists of two children, each being itself a tree, one with leaves a, b and c, and one with leaves d and e. Each node of a rose tree can be taken to mean that its leaves form a cluster of data items that are more similar among themselves than to other data items. To make this precise, we will model each cluster of data items using a single shared parameter, with different clusters using different parameters.

Definition 1.3.2 *A* cluster *is a set of data items $D \subseteq \mathcal{D}$. Let θ parameterize a probability distribution for a data item $x \in D$, $f(x|\theta)$, with a corresponding prior $f(\theta|\eta)$ and hyperparameter η. Marginalising out θ, the probability of a cluster of data items D is:*

$$f(D) = \int f(\theta|\eta) \prod_{x \in D} f(x|\theta) d\theta \tag{1.1}$$

In Section 1.3.2 we describe two models for clusters used in our experiments (Section 1.6): a beta-Bernoulli model and a Gaussian process model.

Definition 1.3.3 *A partition ϕ is a disjoint set of clusters whose union is the whole data set $\mathcal{D}$. We shall write partitions using the "$|$" symbol, for example $ab|c$ denotes a partition of the set $\{a, b, c\}$ into clusters $\{a, b\}$ and $\{c\}$. We model the probability of $\mathcal{D}$ under a partition as the product of the probabilities of its constituent clusters,*

$$g(\phi = \{D_1 | \ldots | D_J\}) = \prod_{j=1}^{J} f(D_j) \tag{1.2}$$

where $f(D_j)$ is the probability of the j^{th} cluster of the partition.

Because each cluster in a partition is modelled independently using one parameter, the likelihood of a partition will be high if data items have high intra-cluster similarities and low inter-cluster similarities.

We have shown how partitions are constructed from clusters, now we turn to showing how a Bayesian rose tree is constructed as a mixture over partitions. Each node of the rose tree is meant to represent a group of data items, at leaves, which share some element of similarity. Therefore, it seems reasonable to assume that each cluster of each partition in the mixture corresponds to one subtree of the rose tree. Partitions consisting of such clusters are called tree consistent partitions, and will constitute the components of the mixture model.

Definition 1.3.4 *A partition is consistent with a rose tree T if each cluster in the partition corresponds to the leaves of some subtree in T. Denote the set of all partitions consistent with T by $\mathbf{\Phi}(T)$.*

Our definition of $\mathbf{\Phi}(T)$ is a straightforward generalisation of the definition of tree-consistent partitions found in Heller and Ghahramani (2005) to rose trees. It is easy to see that $\mathbf{\Phi}(T)$ can be constructed explicitly by recursion as follows:

$$\mathbf{\Phi}(T) = \{\text{leaves}(T)\} \cup \{\phi_1 | \ldots | \phi_{n_T} : \phi_i \in \mathbf{\Phi}(T_i), T_i \in \text{children}(T)\} \tag{1.3}$$

where $\text{children}(T)$ are the children of T, n_T are the number of children of T, and $\{\text{leaves}(T)\}$ represents the partition where all data items at the leaves of T are clustered together. Roughly, each partition starts at the root of the tree, and either keeps the leaves in one cluster or partitions the leaves into the subtrees, the process repeating on each subtree. The end result is that each $\phi \in \mathbf{\Phi}(T)$ consists of disjoint clusters, each of which consists of the leaves of some subtree in T. For example, the partitions consistent with the rose trees in Figure 1.2 are given in the middle column of Figure 1.3.

All rose trees include the complete partition $\{\text{leaves}(T)\}$ and (by recursion) the completely discriminating partition where each data item in $\mathcal{D}$ is in its own cluster. Different trees give rise to different sets of partitions between these two extremities. The binary tree with the fewest tree consistent partitions between these extremities is a cascading binary tree, where at each internal node one leaf is separated from the other leaves. On the other hand, the simplest rose tree has just two consistent partitions: the complete partition consisting of all leaves in one cluster, and the completely discriminating partition. In general, a rose tree will have at most the same number of partitions as a binary tree. The rose tree allows us to represent simple hierarchical structure without having to introduce spurious partitions, like in the cascading binary tree.

We are now ready to define our Bayesian rose tree mixture model:

Definition 1.3.5 *Given a rose tree T, A Bayesian rose tree is a mixture over partitions in $\mathbf{\Phi}(T)$ of the cluster of data items at its leaves $\mathcal{D} = \text{leaves}(T)$:*

$$p(\mathcal{D}|T) = \sum_{\phi \in \mathbf{\Phi}(T)} m(\phi|T) g(\phi) \tag{1.4}$$

where $m(\phi|T)$ is the mixing proportion of partition ϕ, and $g(\phi)$ is the data likelihood term for partition ϕ given in (1.2)*.*

In general the number of partitions consistent with T can be exponentially large in the number of leaves, making (1.4) computationally intractable for large data sets. Instead, we propose a specific factorised form for the mixture proportions, m, that allows for (1.4) to be computed efficiently by recursion as well as for the efficient agglomerative tree construction algorithm in Section 1.4. For each subtree T' of T, let $\pi_{T'}$ be a parameter between 0 and 1. We shall discuss the choice of these parameters at length in Section 1.3.1. We will compute (1.4) recursively as follows:

$$p(\text{leaves}(T)|T) = \pi_T f(\mathcal{D}) + (1-\pi_T) \prod_{T_i \in \text{children}(T)} p(\text{leaves}(T_i)|T_i) \tag{1.5}$$

where $f(\mathcal{D})$ is the probability of the cluster $\mathcal{D}$. Equation (1.5) corresponds to the following generative process: Beginning at the root of the tree, with probability π_T, generate θ according to the prior $f(\theta|\eta)$, and all data items according to $f(\cdot|\theta)$. Otherwise, recurse down the tree, with each subtree independently generating the data items at its leaves according to the same process. If a leaf is reached the recursion stops and the data item is generated. Note that it follows from this narrative that we can identify each cluster of the data with a node of the tree.

Comparing (1.5) and (1.4), we find that

$$m(\phi|T) = \prod_{S \in \text{ancestor}_T(\phi)} (1-\pi_S) \prod_{S \in \text{subtree}_T(\phi)} \pi_S \tag{1.6}$$

where $\text{subtree}_T(\phi)$ are the subtrees in T corresponding to the clusters in the partition ϕ, and $\text{ancestor}_T(\phi)$ are the ancestors of subtrees in $\text{subtree}_T(\phi)$. Figure 1.3 gives these mixing proportions for the two example rose trees.

In summary, the marginal probability of a data set $\mathcal{D}$ under a rose tree T is a mixture over the partitions consistent with T, with the probability of $\mathcal{D}$ under a partition $\phi \in \mathbf{\Phi}(T)$ being a product of the probabilities of clusters in ϕ. We call our mixture a *Bayesian rose tree* (BRT) mixture model. In Section 1.3.1 we motivate using a particular choice of the mixing proportions π_T, and in Section 1.5 we contrast our Bayesian rose tree mixture over partitions to a number of related models.

1.3.1 Avoiding needless cascades

In this section we propose a particular choice for the mixing proportions π_T given by:

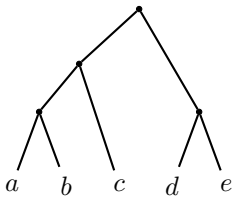
$$\pi_T = 1 - (1-\gamma)^{n_T - 1} \tag{1.7}$$

where $0 \leq \gamma \leq 1$ is a hyperparameter of the model controlling the relative proportion of coarser partitions of the data as opposed to finer ones. When restricted to just binary trees, $\pi_T = \gamma$ and the BRT reduces to one of the models in Heller and Ghahramani (2005). Subtrees T with more children are assigned a larger π_T and so the likelihood of the cluster at the root of the subtree is more highly weighted than smaller clusters further down the subtree.

As we will see, this choice of π_T is intimately related to our maxim that the maximum likelihood tree should be simple if the data are unstructured. We will start by considering the simple situation given by the running examples in Figure 1.3 before the more general case. The two rose trees in Figure 1.3 differ from each other only in that B places the data items

Binary tree B

$m(\phi\|B)$	$\phi \in \mathbf{\Phi}(B)$
γ	$a\ b\ c\ d\ e$
$(1-\gamma)\gamma^2$	$a\ b\ c\|d\ e$
$(1-\gamma)^2\gamma^2$	$a\ b\|c\|d\ e$
$(1-\gamma)^3\gamma$	$a\|b\|c\|d\ e$
$(1-\gamma)^2\gamma$	$a\ b\ c\|d\|e$
$(1-\gamma)^3\gamma$	$a\ b\|c\|d\|e$
$(1-\gamma)^4$	$a\|b\|c\|d\|e$



Rose tree R

$m(\phi\|R)$	$\phi \in \mathbf{\Phi}(R)$
γ	$a\ b\ c\ d\ e$
$(1-(1-\gamma)^2)(1-\gamma)\gamma$	$a\ b\ c\|d\ e$
$(1-(1-\gamma)^2)(1-\gamma)^2$	$a\ b\ c\|d\|e$
$(1-\gamma)^3\gamma$	$a\|b\|c\|d\ e$
$(1-\gamma)^4$	$a\|b\|c\|d\|e$

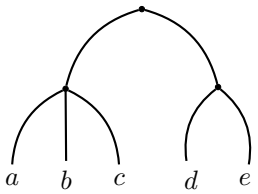


Figure 1.3 Examples of (top) a binary tree B with a cascading subtree (on leaves a, b, and c) and (bottom) a rose tree R with the cascade collapsed into one node. All tree-consistent partitions for each tree, as well as the associated mixture proportions, are listed to the left of the corresponding tree. Note that $m(\{a\,b\,c|d\,e\}|R) = m(\{a\,b\,c|d\,e\}|B) + m(\{a\,b|c|d\,e\}|B)$ and $m(\{a\,b\,c|d|e\}|R) = m(\{a\,b\,c|d|e\}|B) + m(\{a\,b|c|d|e\}|B)$. That is, the mixing proportion assigned to each of the partition in R is the sum of those of partitions in B that are refinements of the partition in R.

a, b, c into a cascading structure with two binary nodes while R uses a single ternary node. The figure also shows the set of partitions and their mixing proportions given by (1.6) and (1.7).

Suppose that a, b and c are similar to each other but are otherwise indistinguishable, yet are distinguishable from d and e. We would like a model that prefers the rose tree R over the binary tree B, since its structure matches the similarity relationships among the data items. To do this, note that because the data items a, b, c belong together in one cluster, we can expect the inequality $f(\{a,b,c\}) > f(\{a,b\})f(\{c\})$ to hold. This implies the following inequalities among the partition likelihoods:

$$\begin{aligned} g(\{a\,b\,c|d\,e\}) &> g(\{a\,b|c|d\,e\}) \\ g(\{a\,b\,c|d|e\}) &> g(\{a\,b|c|d|e\}) \end{aligned} \tag{1.8}$$

Expanding the likelihoods for R and B as a mixture of the likelihoods under each partition, using the inequalities (1.8), and using the equalities in the mixing proportions noted in the caption of Figure 1.3, we find that

$$p(\{a,b,c,d,e\}|R) > p(\{a,b,c,d,e\}|B) \tag{1.9}$$

In other words, the collapsed rose tree R is assigned a higher likelihood than the binary tree B. Therefore if we were to select the rose tree with higher likelihood we would have chosen the one that better describes the data, i.e. R.

In the general case, if we have a cluster of n indistinguishable data items, we can guarantee preferring a rose tree R consisting of a single internal node with n children over a tree B with multiple internal nodes, for example a cascading binary tree, if the mixing proportion of the complete partition in R is the sum over the mixing proportions of all partitions consistent with B except the most discriminating partition. Fortunately, under (1.7) this sum turns out to be the same regardless of the structure of B, and equals,

$$\pi_R = 1 - (1-\gamma)^{n-1} \tag{1.10}$$

Equation (1.10) can be easily proven by induction on the number of internal nodes of B. Using this, we can now show that the collapsed rose tree R will be more likely than a rose tree which introduces spurious structure (e.g. cascades).

Proposition 1.3.6 *Let B be rose tree with $n_B > 1$ children $T_1, \ldots, T_{n_B}$, and T_1 being an internal node. Let R be a rose tree obtained by collapsing the B and T_1 nodes into one, i.e. R has children$(R) =$ children$(T_1) \cup T_2 \cup \cdots T_{n_B}$. Suppose the data items of T are indistinguishable, i.e. the likelihoods of the non-complete partitions are smaller than for the complete partition:*

$$g(\{\textit{leaves}(T_1)|\phi_2|\ldots|\phi_{n_B}\}) < g(\{\textit{leaves}(R)\}) \tag{1.11}$$

for every $\phi_i \in \mathbf{\Phi}(T_i)$, $i = 2, \ldots, n_B$. Then the likelihood of B is lower than for R:

$$p(\textit{leaves}(B)|B) < p(\textit{leaves}(R)|R). \tag{1.12}$$

Proof. By construction, we have $\mathbf{\Phi}(R) \subset \mathbf{\Phi}(B)$. Let $\psi = \{\text{leaves}(R)\}$ be the complete partition and $\delta(B,R) = \mathbf{\Phi}(B) \backslash \mathbf{\Phi}(R)$ be those partitions of B not in R. It is straightforward to see that

$$\delta(B,R) = \{\text{leaves}(T_1)|\phi_2|\ldots|\phi_{n_B} : \phi_i \in \mathbf{\Phi}(T_i)\} \tag{1.13}$$

The mixture likelihood of B can now be decomposed as:

$$p(\text{leaves}(B)|B) = \pi_B g(\psi) + \sum_{\phi \in \delta(B,R)} m(\phi|B) g(\phi) + \sum_{\phi \in \mathbf{\Phi}(R)\backslash\{\psi\}} m(\phi|B) g(\phi) \tag{1.14}$$

From the premise (1.11) we find that:

$$p(\text{leaves}(B)|B) < g(\{\psi\}) \left(\pi_B + \sum_{\phi \in \delta(B,R)} m(\phi|B) \right) + \sum_{\phi \in \mathbf{\Phi}(R)\backslash\{\psi\}} m(\phi|B) g(\phi) \tag{1.15}$$

We now turn to evaluating the summation over $\delta(B,R)$ in (1.15). From (1.13) and (1.5) we see that the mixture proportion assigned to each $\phi \in \delta(B,R)$ is:

$$m(\phi|B) = (1-\pi_B)\pi_{T_1} \prod_{i=2}^{n_B} m(\phi_i|T_i) \tag{1.16}$$

Because (1.16) decomposes into a product over the partition of each subtree T_i, $i = 2, \ldots, n_B$, and $\sum_{\phi_i \in \mathbf{\Phi}(T_i)} m(\phi_i|T_i) = 1$, we see that $\sum_{\phi \in \delta(B,R)} m(\phi|B) = (1-\pi_B)\pi_{T_1}$. Now from (1.7) we see that the term in the parentheses of (1.15) is

$$\pi_B + (1-\pi_B)\pi_{T_1} = \pi_R = m(\psi|R). \tag{1.17}$$

On the other hand $m(\phi|B) = m(\phi|R)$ for each $\phi \in \mathbf{\Phi}(R)\backslash\{\psi\}$. We have now established the right hand side of (1.15) as $p(\text{leaves}(R)|R)$. □

Proposition 1.3.6 applies when the root of the tree along with one of its children are collapsed into one node. This can be trivially generalised to collapsing any subtree.

Corollary 1.3.7 *Let S be a rose tree with a subtree B, and T be constructed by collapsing all internal nodes of B into one node. If the data items under B are indistinguishable, i.e. non-complete partitions in B have lower likelihoods than the complete one, then:*

$$p(\textit{leaves}(S)|S) < p(\textit{leaves}(T)|T). \tag{1.18}$$

1.3.2 Cluster models

Each cluster D of data items has an associated marginal likelihood $f(D)$ defined according to (1.1). In this chapter we consider two families of parameterised cluster models: for d-dimensional binary data, we use a product of beta-Bernoulli distributions, and for curves in $\mathbb{R}^2$, we use Gaussian processes. Other cluster models may be considered, e.g. other exponential families with conjugate priors.

Binary data clusters

For d-dimensional binary data, we model the ith dimension independently using a Bernoulli distribution with parameter θ_i, and use a beta prior for θ_i with hyperparameters (α_i, β_i). Integrating out the parameters, the cluster likelihood $f(D)$ is then the probability mass formed by a product of independent beta-Bernoulli distributions in each dimension:

$$\begin{aligned} f(D) &= \prod_{i=1}^{d} \int f(D_i|\theta_i) f(\theta_i|\alpha_i, \beta_i) d\theta_i \\ &= \prod_{i=1}^{d} \frac{\text{Beta}(\alpha_i + n_i, \beta_i + |D| - n_i)}{\text{Beta}(\alpha_i, \beta_i)} \end{aligned} \tag{1.19}$$

where D_i consists of the ith entry of each data item in D, n_i is the number of ones in D_i, and $|D|$ is the total number of data items in cluster D. The hyperparameters of the entire cluster model are thus $\eta = \{(\alpha_i, \beta_i)\}_{i=1}^{d}$.

Gaussian process expert clusters

Here we consider data items consisting of input-output pairs, $D = \{(x_i, y_i)\}_{i=1}^{n}$, and are interested in modelling the conditional distribution over outputs given inputs. Rasmussen and Ghahramani (2002) proposed a DP mixture of Gaussian process (GP) experts where a data set is partitioned, via the DP mixture, into clusters each of which is modelled by a GP. Such a model can be used for nonparametric density regression, where a full conditional density over an output space is estimated for each value of input. This allows generalisation of GPs allowing for multi-modality and non-stationarity. The original model in Rasmussen and Ghahramani (2002) had mixing proportions which do not depend on input values; this was altered in the paper in an ad hoc manner using radial basis function kernels. Later Meeds and Osindero (2006) extended the model by using a full joint distribution over both inputs and outputs, allowing for properly defined input dependent mixing proportions.

With both approaches MCMC sampling was required for inference, which might be slow in convergence. Here we consider using Bayesian rose trees instead. The joint distribution of each cluster is modelled using a Gaussian over the inputs and a GP over the outputs given the inputs:

$$f(D) = f(\{x_i\})f(\{y_i\}|\{x_i\}) \tag{1.20}$$

where

$$f(\{x_i\}) = \int\int \left[\prod_{i=1}^{n} \mathcal{N}(x_i|\mu, \boldsymbol{R}^{-1})\right] \mathcal{N}(\mu|m, (r\boldsymbol{R})^{-1})\mathcal{IG}(\boldsymbol{R}|\boldsymbol{S}, \nu)\mathrm{d}\mu\mathrm{d}R$$

$$f(\{y_i\}|\{x_i\}) = \mathcal{N}(\{y_i\}|\mathbf{0}, \boldsymbol{K}) \tag{1.21}$$

where $\mathcal{N}(x|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Normal density with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, $\mathcal{IG}(\boldsymbol{R}|\boldsymbol{S}, \nu)$ is a Wishart density with degrees of freedom ν and scale matrix $\boldsymbol{S}$, and the matrix $\boldsymbol{K}$ is a Gram matrix formed by the covariance function of the GP (we used the squared exponential). The normal inverse Wishart prior over the parameters μ and $\boldsymbol{R}$ is conjugate to the normal likelihood, so $f(D)$ can be computed analytically. It follows that for a Gaussian process expert cluster that the hyperparameters η are $(r, \nu, \boldsymbol{S})$ where r is the scaling parameter of the normal inverse Wishart prior.

1.4 Greedy construction of Bayesian rose tree mixtures

We take a model selection approach to finding a rose tree structure given data. Ideally we wish to find a rose tree T^* maximising the marginal probability of the data $\mathcal{D}$:

$$T^* = \operatorname*{argmax}_T p(\mathcal{D}|T) \tag{1.22}$$

This is intractable since there is a super-exponential number of rose trees.

Inspired by the success of other agglomerative clustering algorithms, we instead consider constructing rose trees in a greedy agglomerative fashion as follows. Initially every data item is assigned to its own rose tree: $T_i = \{x_i\}$ for all data items x_i. At each step of our algorithm we pick two rose trees T_i and T_j and merge them into one tree T_m using one of a few merge operations. This procedure repeats until just one tree remains (for n data items this will occur after $n - 1$ merges), and is illustrated in Figure 1.4.

Each step of the algorithm consists of picking a pair of trees as well as a merge operation. We use a maximum likelihood ratio criterion, picking the combination that maximises:

$$L(T_m) = \frac{p(\text{leaves}(T_m)|T_m)}{p(\text{leaves}(T_i)|T_i)p(\text{leaves}(T_j)|T_j)} \tag{1.23}$$

We use the likelihood ratio rather than simply the likelihood $p(\text{leaves}(T_m)|T_m)$ because the denominator makes $L(T_m)$ comparable across different choices of trees T_i and T_j of differing sizes (Friedman 2003; Heller and Ghahramani 2005).

We considered a number of merge operations to allow for nodes with more than two children to be constructed: a *join*, an *absorb*, and a *collapse* operation (see Figure 1.5). In all

input: data $\mathcal{D} = \{x_1 \dots x_n\}$,
cluster model $p(x|\theta)$,
cluster parameter prior $p(\theta|\eta)$,
cluster hyperparameters η
initialise: $T_i = \{x_i\}$ for $i = 1 \dots n$
for $c = n$ to 2 **do**
Find the pair of trees T_i and T_j, and merge operation m with the highest likelihood ratio:

$$L(T_m) = \frac{p(\text{leaves}(T_m)|T_m)}{p(\text{leaves}(T_i)|T_i)p(\text{leaves}(T_j)|T_j)}$$

Merge T_i and T_j into T_m using operation m
$T_{n+c-1} \leftarrow T_m$
Delete T_i and T_j
end for
output: Bayesian rose tree T_{n+1}, a mixture over partitions of $\mathcal{D}$

Figure 1.4 Agglomerative construction algorithm for Bayesian rose trees.

operations the merged rose tree T_m has leaves$(T_m) =$ leaves$(T_i) \cup$ leaves(T_j), the difference being the resulting structure at the root of the merged tree. For a *join*, a new node T_m is created with children T_i and T_j. A join is chosen if the children of T_i and T_j are related, but are sufficiently distinguishable to keep both subtrees separated. For an *absorb* the children of the resulting tree T_m are children$(T_i) \cup \{T_j\}$, that is, tree T_j is absorbed as a child of T_i. This operation is chosen if the children are related, but there exists finger distinguishing structure already captured by T_j. This operation is not symmetric so we also consider the converse, where the children are $\{T_i\} \cup$ children(T_j). Finally, a *collapse* merges the roots of both trees, so that the resulting children of T_m are children$(T_i) \cup$ children(T_j). This is performed when the children of T_i and T_j are indistinguishable so may be combined and treated similarly.

Binary hierarchical clustering algorithms such as Heller and Ghahramani (2005) only need to consider the join operation. To be able to construct every possible rose tree the absorb operation is necessary as well. Intuitively, a join merge makes the tree taller whilst an absorb merge makes the tree wider. The collapse operation is not technically necessary, however we found that including it allowed us to find better rose trees.

In general the computational complexity of the greedy agglomerative clustering algorithm of Section 1.4 is in $\Omega(n^2 \log nL)$ (where L is a contribution to the complexity due to the particular cluster likelihood). Firstly, for every pair of data items we must calculate the likelihood of a merged tree—there are $O(n^2)$ such pairs. Secondly, these pairs must be sorted—requiring $O(n^2 \log n)$ computational complexity. The data structure we use is simply a binary heap.

If the cluster marginal likelihood is a d-dimensional product of beta-Bernoulli distributions (i.e., for d-dimensional binary valued data) then $L = O(d)$. Instead of keeping track of every data items, it is sufficient to keep track of the sufficient statistics (counts of zeros and ones) of each cluster. The same argument applies to any conjugate exponential family cluster model.

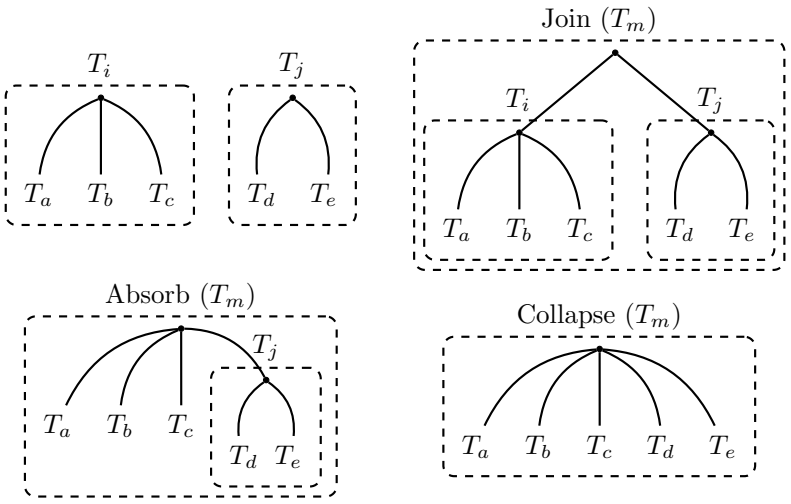


Figure 1.5 Merges considered during greedy search.

For a BRT mixture of Gaussian process experts, $L = O(n^3)$, which comes from performing Cholesky decompositions of Gram matrices.

There are several opportunities for approximations to improved the computational complexity of the greedy agglomerative algorithm. Heller (2008) explored using randomisation to sample random subsets of the data set to make the computational complexity scale more favourably, and Xu et al. (2009) use a flat clustering to constrain one level of the hierarchy, thereby reducing the complexity of discovering the remaining structure. Low-rank approximations (such as Williams and Seeger (2001)) could also be used to reduce the computational complexity of the Gaussian process expert variant.

1.4.1 Prediction

Two kinds of prediction are possible with BRT: predicting partially observed data items, and predicting unobserved data items themselves.

The predictive probability of a Bayesian rose tree for partially observed data is

$$p(\mathcal{D}^m|\mathcal{D}, T) = \frac{p(\mathcal{D}^m, \mathcal{D}|T)}{p(\mathcal{D}|T)} \tag{1.24}$$

where $\mathcal{D}^m$ are the unobserved parts of data items in $\mathcal{D}$. The denominator of (1.24) is the quantity optimised to find T, and calculating it is tractable if marginalising components of the cluster likelihood is tractable.

As the rose tree T only accounts observed data items, predicting unobserved data requires additional assumptions about the location of unobserved data within the rose tree. The assumption we make is the same as in Heller (2008): the probability of an unobserved data item being in a particular cluster is proportional to the number of observed data items in

that cluster and also the number of observed data items in any cluster above it in the tree. Intuitively this assumption means that an unobserved data item is more likely to come from larger cluster than a smaller cluster and it is more likely to come from a cluster higher up the tree than further down the tree. The predictive distribution of an unobserved data item is then a mixture over clusters in T:

$$p(x|\mathcal{D},T) = \sum_{S\in\text{subtree}(T)} w_S f(x|\text{leaves}(S)) \tag{1.25}$$

$$\text{where } w_S = r_S \left[\prod_{A\in\text{ancestor}(S)} (1-r_A)\frac{n_{A\to S}}{|\text{leaves}(S)|}\right]$$

$$r_S = \frac{\pi_S f(\text{leaves}(S))}{p(\mathcal{D}|S)} \tag{1.26}$$

where subtree(T) are the subtrees of T corresponding to each cluster in T, ancestor(S) are the ancestors of the subtree S, and $n_{A\to S}$ is the number of data items in the subtree of A containing all the leaves of S. $f(x|\text{leaves}(S))$ is the predictive cluster distribution of the corresponding cluster model. Since in (1.25) x belongs to every cluster of T with some probability, (1.25) does not describe a Bayesian rose tree mixture model.

1.4.2 Hyperparameter optimisation

We optimise the hyperparameters of the cluster marginal likelihood, η, and the mixture proportion parameter, γ, by gradient ascent on the log marginal likelihood $\log p(\mathcal{D}|T)$. From (1.5), the gradient of the marginal log likelihood $\log p(\mathcal{D}|T)$ with respect to the cluster hyperparameters can be efficiently computed recursively:

$$\frac{\partial \log p(\mathcal{D}|T)}{\partial \eta} = r_T \frac{\partial \log f(\text{leaves}(T))}{\partial \eta} + (1-r_T) \sum_{T_i\in\text{children}(T)} \frac{\partial \log p(\text{leaves}(T_i)|T_i)}{\partial \eta} \tag{1.27}$$

where r_T is given by (1.26).

Similarly, the gradient for the mixture proportion parameter γ:

$$\frac{\partial \log p(\mathcal{D}|T)}{\partial \gamma} = r_T \frac{\partial \log \pi_T}{\partial \gamma} + (1-r_T)\left[\frac{\partial \log(1-\pi_T)}{\partial \gamma} + \sum_{T_i\in\text{children}(T)} \frac{\partial \log p(\text{leaves}(T_i)|T_i)}{\partial \gamma}\right]. \tag{1.28}$$

After optimising the hyperparameters for a particular tree, the marginal log likelihood can be optimised further using these hyperparameters in a coordinate ascent procedure: greedily find a better tree given the current hyperparameters (Figure 1.4), then find the best hyperparameters for that tree (via (1.27) and (1.28)), and repeat until convergence, alternating between optimising the hyperparameters and the tree. This optimisation procedure is not

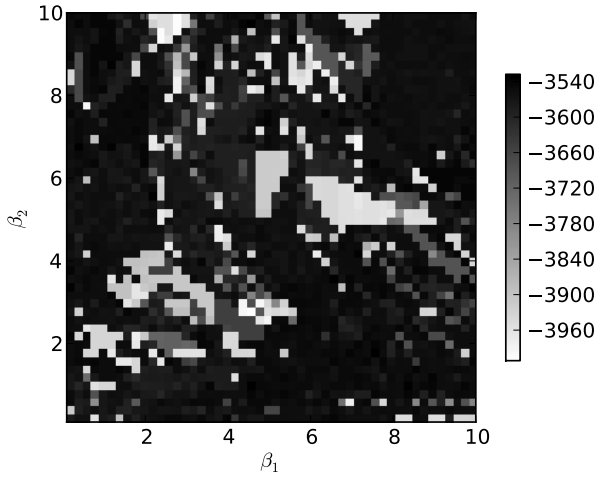


Figure 1.6 Sensitivity of optimising the marginal likelihood to initial conditions. β_1 and β_2 of beta-Bernoulli cluster model initialised as above, all other hyperparameters held fixed. Background colour indicates marginal log likelihood at convergence via scale at left of plot.

guaranteed to find a global optimum of the marginal likelihood, as the marginal likelihood is typically not convex in its cluster hyperparameters. However the optimisation procedure will eventually converge upon a local optimum for the hyperparameters and tree, if the cluster likelihood is bounded, as both steps optimise the same objective function.

We found that, particularly where binary data are missing, optimising the beta-Bernoulli hyperparameters is sensitive to initial conditions: Figure 1.6 shows the effect of changing the value of just two of the hyperparameters of the beta-Bernoulli cluster models on the optimised log marginal likelihood $\log p(\mathcal{D}|T)$ of BRT. All other hyperparameters were held fixed. Consequently, we used 10 restarts at random points around a MAP estimate of the cluster hyperparameters, and for missing data, we averaged the hyperparameters of the beta-Bernoulli model over a small region around the optimum found.

1.5 Bayesian hierarchical clustering, Dirichlet process and product partition models

In this section we describe a number of models related to Bayesian rose trees: finite mixture models, product partition models (Barry and Hartigan 1992), PCluster (Friedman 2003), Bayesian hierarchical clustering (BHC) (Heller and Ghahramani 2005), and Dirichlet process mixture models.

1.5.1 Mixture Models and Product Partition Models

A Bayesian rose tree is a mixture model over partitions of the data. This is an unusual way to model how data items cluster and it may be beneficial to consider how other clustering models relate to mixtures over partitions.

We start by considering a typical Bayesian mixture model consisting of K components. Such a model associates each data item, say x_i, with a latent indicator variable, say $z_i \in \{1, \ldots, K\}$, for the cluster to which it belongs. We can express this as a mixture over partitions as follows:

$$p(\mathcal{D}) = \prod_{i=1}^{n} \sum_{z_i=1}^{K} p(z_i)p(x_i|z_i) = \sum_{\phi} m(\phi) \prod_{D_j \in \phi} f(D_j) = \sum_{\phi} m(\phi)g(\phi) \tag{1.29}$$

where the component parameters have been marginalized out, and ϕ ranges over all possible partitions of the data with up to K clusters. This is a consequence of interchanging the product and summation and using the commonality among the values of the z_i's to form partitions. The likelihood $g(\phi)$ is the probability of the data under the partition ϕ given in (1.2), whilst $m(\phi)$ is the mixing proportion over partitions and is obtained by summing over all assignments $\{z_i\}$ giving rise to partition ϕ.

If the mixing proportion factorises into separate terms for each cluster, say $m(\phi) = \prod_{D \in \phi} m(D)$, the last term of (1.29) is the marginal probability of a product partition model (Barry and Hartigan 1992). An example of a product partition model with an unbounded number of mixture components is the Dirichlet process (DP) mixture model, which has $m(\phi)$ corresponding to the probability of a particular seating arrangement under the Chinese restaurant process:

$$m(\phi) = \frac{\prod_{D \in \phi} \alpha \Gamma(|D|)}{\alpha(\alpha+1)\cdots(\alpha+n-1)} \tag{1.30}$$

In the context of regression, our Bayesian rose tree mixture of Gaussian process experts also bears some resemblance to an extension of product partition models described by Müller and Quintana (In Press). There, the mixing proportions $m(\phi)$ are allowed to depend on covariates in a form functionally similar to one we would obtain if we condition on the inputs in (1.21).

Compared to the finite mixture model and product partition model, the Bayesian rose tree mixture model allows a larger range over the number of components in its partitions, ranging from those with a single cluster to those with as many clusters as there are data items. On the other hand, all the partitions in a Bayesian rose tree have to be consistent with the tree structure. A Bayesian rose tree can be interpreted as follows: the data are partitioned in some unknown way, but all the reasonable ways in which the data could be partitioned are consistent with some rose tree.

1.5.2 PCluster and Bayesian Hierarchical Clustering

Our work on Bayesian rose trees is directly motivated by issues arising from prior works on PCluster and Bayesian hierarchical clustering (BHC). The first model, PCluster (Friedman 2003), is a direct probabilistic analogue of linkage methods where a likelihood ratio similar to (1.23) is used to measure the distance between two clusters. Each iteration of the agglomeration procedure thus produces a partition of the data with a likelihood similar to (1.2). However the resulting tree structure itself does not correspond to a probabilistic model of the data, rather it is simply a trace of the series of partitions discovered by the procedure.

Addressing the lack of a probabilistic model in PCluster, Heller and Ghahramani (2005) proposed a probabilistic model indexed by binary trees called BHC. BHC is a mixture over

partitions consistent with the binary tree, and the BRT approach described in the present chapter is a generalisation and reinterpretation of BHC. There are three distinct differences between BRT and BHC: Firstly, the likelihood and agglomerative construction of BHC only accounts for binary trees. Secondly, Heller and Ghahramani (2005) considered two alternative parameterizations of the mixing proportions π_T. In the first parameterisation of BHC, which we shall call BHC-γ, π_T equals some constant γ, and is in agreement with BRT on binary trees. The second parameterisation of BHC, which we shall call BHC-DP, uses a π_T which leads to $m(\phi)$ being precisely the mixing proportion of ϕ under the DP mixture (1.30). Since the marginal probability of the data under the DP mixture is a sum over all partitions, while that for BHC is only over those consistent with the tree, (1.4) gives a lower bound on the DP mixture marginal probability. Note that a similar setting of π_T in BRT allows it to also produce lower bounds on the DP mixture marginal probability. However, since the set of partitions consistent with a BRT is always a subset of the ones consistent with some binary tree where we replace each non-binary internal node on the rose tree with a cascade of binary nodes, the BRT lower bound will always be no higher than that for the BHC. This argument obviates the use of BRT as approximate inference for DP mixtures. In fact our reason for using rose trees is precisely because the set of partitions is smaller—all else being equal, we should prefer simpler models by Occam's Razor. This view of hierarchical clustering is very different from the one expounded by Heller and Ghahramani (2005), and is the third distinction between BHC and BRT. In the next section we will compare the parameterization of BRTs described in Section 1.3 against BHC as well as other parameterizations of BRT inspired by BHC.

1.6 Results

We compared BRT with several alternate probabilistic hierarchical clustering models: two binary hierarchical clustering algorithms, BHC-γ and BHC-DP, and two other rose tree hierarchical clustering algorithms. We shall call the model where BRT has $\pi_T = \gamma$, BRT-γ. BRT-γ differs from BHC-γ only in the number of possible children. Furthermore we shall call the model where BRT has π_T configured in a similar fashion to BHC-DP, BRT-DP. In this way, all models with prefix "BRT" shall have rose trees and all models with prefix "BHC" shall have binary trees, whilst the suffix denotes how π_T is parameterised. The cluster likelihood models used are described in Section 1.3.2.

For BHC-DP and BRT-DP we report its marginal likelihood $p(\mathcal{D}|T)$, not the lower bound on the DP mixture, which is $p(\mathcal{D}|T)$ multiplied by a factor that is less than one.

1.6.1 Optimality of tree structure

The agglomerative scheme described in Section 1.4 is a greedy algorithm that is not guaranteed to find the optimal tree. Here we compare the trees found by BRT, BHC-γ and BHC-DP against the optimal (maximum likelihood) rose tree T^* found by exhaustive search. We generated data sets of sizes ranging from 2 to 8, each consisting of binary vectors of dimension 64, from a BRT mixture with randomly chosen rose tree structures. On each of the n data sets we compare the performances in terms of the average $\log_2$ probability of the data assigned by the three greedily found trees T relative to the maximum likelihood Bayesian

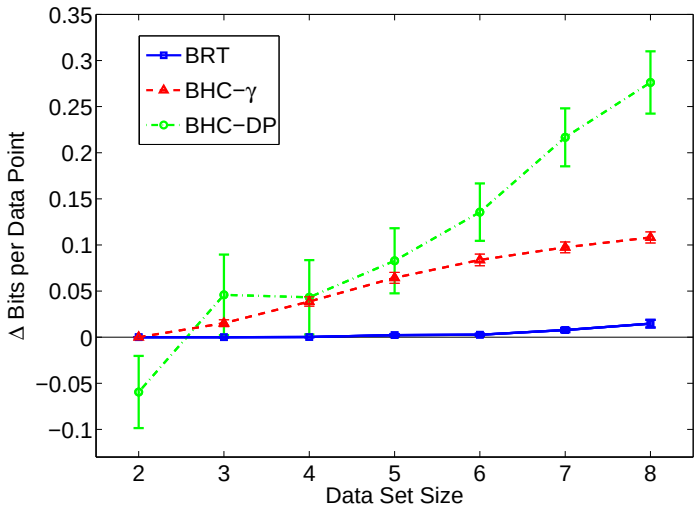


Figure 1.7 Per data item $\log_2$ probability of trees found greedily by BRT, BHC-γ and BHC-DP, relative to the optimal Bayesian rose tree. Error bars are one standard error. Lower in graph is better.

rose tree T^*,

$$\Delta_l = \frac{1}{ln} \sum_{i=1}^{n} \log_2 p(\mathcal{D}_i|T_i^*) - \log_2 p(\mathcal{D}_i|T_i) \tag{1.31}$$

where l is the number of data vectors in the data set. Δ_l measures the average number of bits required to code for a data vector under T, in excess of the same under T^*. The results, averaged over 100 data sets per data set size are shown in Figure 1.7. We see that BRT finds significantly better trees than either BHC algorithms. We also found that BRT frequently finds the optimal tree, e.g. when $l = 8$ BRT found the optimum 70% of the time. Note that when $l = 2$ BHC-DP produced higher log probability than the optimal BRT T^*, although it performed significantly worse than BHC-γ and BRT for larger l. This is because the BHC-DP and BRT models are not nested so BHC-DP need not perform worse than T^*.

1.6.2 Hierarchy likelihoods

We compared the marginal likelihoods of trees found by BHC-γ, BHC-DP, BRT-γ, BRT-DP, and BRT on five binary-valued data sets. These are the same data sets used in Heller and Ghahramani (2005). The characteristics of the data sets are summarised in Table 1.1. `toy` is a synthetic data set constructed where ones only appear in three disjoint parts of the binary vector, with each class having ones in a different part. The hierarchies in Figure 1.1 were found by BHC-γ and BRT on this data set. `spambase` is the UCI repository data set of the same name (Frank and Asuncion 2010). `newsgroups` is the CMU 20newsgroups data set restricted to the news groups `rec.autos`, `rec.sport.baseball`, `rec.sport.hockey`, and `sci.space`, and pre-processed using Rainbow (McCallum 1996). `digits` is a subset of the CEDAR Buffalo

Table 1.1 Characteristics of data sets.

Data set	**Attributes**	**Classes**	**Binarisation**
`toy`	12	3	handcrafted
`spambase`	57	2	zero or non-zero
`newgroups`	485	4	word presence/absence
`digits`	64	10	threshold at 32
`digits024`	64	3	threshold at 32

Table 1.2 Log marginal likelihoods and standard errors

Data set	BHC-DP	BRT-DP	BHC-γ	BRT-γ	BRT
`toy`	-215 ± 0.0	-215 ± 0.0	-168 ± 0.1	-167 ± 0.2	$\mathbf{-166} \pm 0.1$
`spambase`	-2258 ± 7.3	-2259 ± 7.2	-1980 ± 7.0	-2006 ± 8.0	$\mathbf{-1973} \pm 7.6$
`digits024`	-4010 ± 6.8	-4015 ± 6.8	-3711 ± 6.9	-3726 ± 6.9	$\mathbf{-3702} \pm 7.0$
`digits`	-4223 ± 6.9	-4216 ± 6.9	-3891 ± 6.7	-3916 ± 6.6	$\mathbf{-3888} \pm 6.8$
`newsgroups`	-10912 ± 61	-10937 ± 55	$\mathbf{-10606} \pm 63$	-10807 ± 59	-10645 ± 60

Table 1.3 Log_{10} of the number of partitions used by the maximum likelihood tree, with standard errors.

Data set	BHC-DP	BRT-DP	BHC-γ	BRT-γ	BRT
`toy`	4 ± 0.0	4 ± 0.0	4 ± 0.0	4 ± 0.0	$\mathbf{1} \pm 0.0$
`spambase`	14 ± 0.1	14 ± 0.1	14 ± 0.1	14 ± 0.1	$\mathbf{7} \pm 0.1$
`digits024`	15 ± 0.1	15 ± 0.1	15 ± 0.1	13 ± 0.1	$\mathbf{6} \pm 0.1$
`digits`	17 ± 0.1	17 ± 0.1	17 ± 0.1	16 ± 0.1	$\mathbf{8} \pm 0.1$
`newsgroups`	14 ± 0.1	14 ± 0.1	14 ± 0.1	13 ± 0.1	$\mathbf{7} \pm 0.1$

Table 1.4 Log predictive probabilities on 10% missing data and standard errors.

Data set	BHC-DP	BRT-DP	BHC-γ	BRT-γ	BRT
`toy`	-14.7 ± 0.7	-14.7 ± 0.7	-14.4 ± 0.6	-14.6 ± 0.6	$\mathbf{-14.3} \pm 0.6$
`spambase`	-190 ± 1.7	$\mathbf{-187} \pm 1.9$	-192 ± 1.7	-192 ± 1.7	-190 ± 1.6
`digits024`	-347 ± 2.1	-345 ± 2.3	-345 ± 2.2	-345 ± 2.2	$\mathbf{-343} \pm 2.1$
`digits`	-372 ± 2.6	-371 ± 2.7	$\mathbf{-369} \pm 2.7$	-371 ± 2.6	-370 ± 2.8
`newsgroups`	-1122 ± 11	-1122 ± 11	$\mathbf{-1114} \pm 11$	$\mathbf{-1114} \pm 11$	$\mathbf{-1114} \pm 11$

digits data set, and `digits024` is the same data set with only samples corresponding to the digits 0, 2, and 4.

Each data set consists of 120 data vectors split equally among the classes, except for `toy` which has only 48 data vectors. When the original data sets are larger the 120 data vectors are subsampled from the original.

Table 1.2 shows the average log likelihoods of the trees found on these data sets. BRT typically finds a more likely tree; the difference between the DP-approximating models and others is significant, whilst the difference between the $\pi_T = \gamma$ models and BRT is often significant.

Table 1.3 shows the logarithm (base 10) of the number of partitions represented by the maximum likelihood trees. BRT typically finds a tree with far fewer partitions than the other models corresponding to a simpler model of the data. The other rose tree-based models (BRT-DP and BRT-γ) have the same or only slightly fewer partitions than their corresponding binary tree equivalents (BHC-DP and BHC-γ, respectively). This reflects our design of the mixing proportion γ_T of BRT: partitions should only be added to the mixture where doing so produces a more likely model. The resulting BRT model is easier to interpret by the practitioner than the alternatives.

1.6.3 Partially observed data

We compared the predictive probabilities of trees found by BHC-γ, BHC-DP, BRT-γ, BRT-DP, and BRT on the same five binary-valued data sets as in the previous section, but with partially observed data. 10% of the data were removed at random. The predictive probabilities, as calculated as in (1.24), are shown in Table 1.4 along with the standard errors. The predictive performance of all five models is similar, with BRT performing slightly better on `toy` and `digits024`, whilst BRT-DP and BHC-γ perform slightly better on `spambase` and `digits`, respectively.

1.6.4 Psychological hierarchies

The data set of Figure 1.8 is from Cree and McRae (2003). The data set consists of a matrix whose rows correspond to objects and whose columns correspond to features. The elements of the matrix are binary and indicate whether a particular object has a particular attribute. There are 60 objects and 100 attributes (such as used for transportation, has legs, has seeds, is cute). Figure 1.8 shows the trees found by BRT and BHC-γ by clustering the 60 objects.

One noticeable oddity of the hierarchical clustering produced by BRT is that lions and tigers inhabit their own cluster that is divorced of from the other animals. The features of the data set also include is it ferocious?, does it roar? which only lions and tigers have, whilst they share few attributes in common to other animals in this data set: this is why they lie on a distinct branch. Removing the attributes is it ferocious?, does it roar? and lives in the wilderness (shared only with deer) cause lions and tigers to be included along with other animals.

This figure shows how BRT not only finds simpler, easier to interpret hierarchies than BHC-γ but also more probable explanations of the data. BHC-DP, BRT-DP and BRT-γ also find similarly less probable and more complicated hierarchies than BRT.

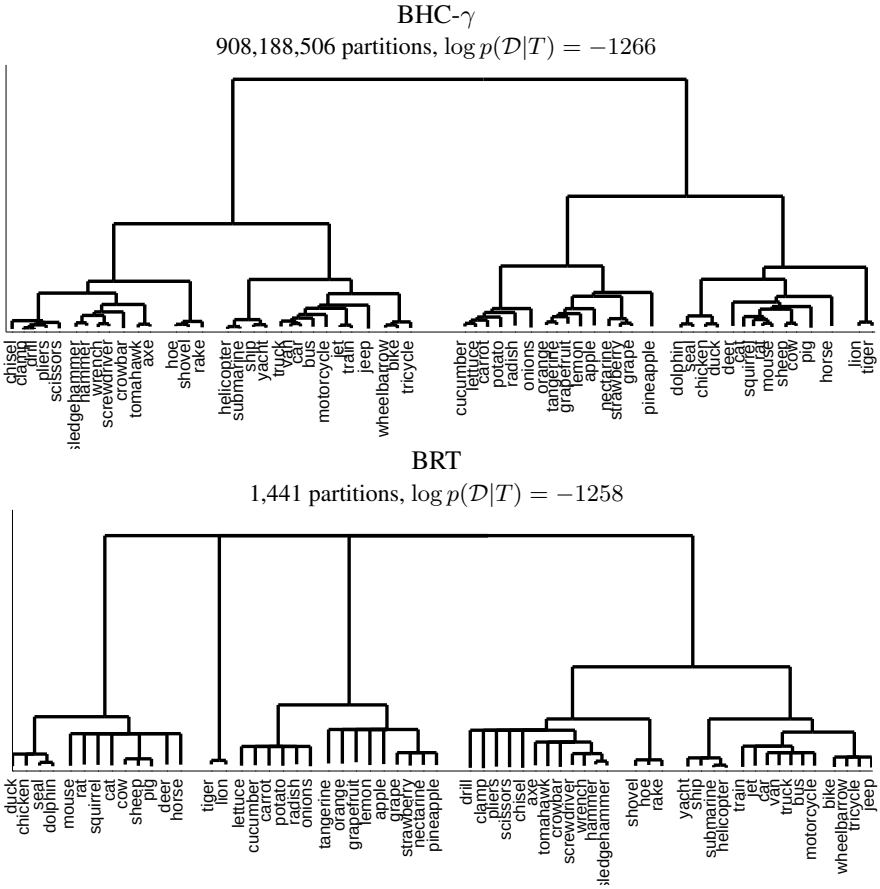


Figure 1.8 Hierarchies found by BHC-γ and BRT on 60 objects (with 100 binary features) data set.

1.6.5 Hierarchies of Gaussian process experts

Figure 1.9 shows fits by a single Gaussian process, a BHC-γ mixture of Gaussian process experts, and a BRT mixture of Gaussian process experts, to multi-modal data consisting of two noisy interlaced sine waves. The background of the figure is coloured according to the predictive density of the model: for BHC-γ and BRT this is calculated as in (1.25). The solid lines on the Gaussian process and BRT plots correspond to the posterior mean of the clusters. BRT finds a simpler and more probable explanation of the data than the alternate models: two primary clusters (in red and yellow) identify with each sine wave, and a root cluster (in green) tying the data together.

1.7 Discussion

We have described a model and developed an algorithm for performing efficient, non-binary hierarchical clustering. Our Bayesian rose tree approach is based on model selection: each

tree is associated with a mixture of partitions of the data set, and a greedy agglomerative algorithm finds trees that have high marginal likelihood under the data.

Bayesian rose trees are a departure from the common binary trees for hierarchical clustering. The flexibility implied by a mixture model over partitions with tree consistency is used in Bayesian rose trees to allow mixture models with fewer components, and thus simpler explanations of the data, than those afforded by BHC. The BRT mixture proportions are designed so that simpler models which explain the data are favoured over more complicated ones: this is in contrast to BHC-DP where forced binary merges create extra, spurious structure which is not supported by the data.

We have demonstrated in our experiments that this algorithm finds simple models which explain both synthetic and real-world data. On all data sets considered, our Bayesian rose tree algorithm found a rose tree with higher marginal likelihood under the data than Bayesian hierarchical clustering (BHC), as well significantly simpler explanations of the data, in terms of the number of partitions. We built BRTs using two likelihood models, a beta-Bernoulli and a Gaussian process expert. In both cases the model yielded reasonable mixtures of partitions of the data.

Our use of BRT for nonparametric conditional density estimation is a proof of concept. BRT offers an attractive means of fitting a mixture of GP experts compared to sampling (Meeds and Osindero 2006; Rasmussen and Ghahramani 2002): with sampling one is never sure when the stationary distribution is attained, while the BRT algorithm is guaranteed to terminate after a greedy pass through the data set, constructing a reasonably good estimate of the conditional density. Note however that the run time of the current algorithm is $O(n^5 \log n)$ where n is the number of data items. The additional $O(n^3)$ factor is due to the unoptimised GP computations. An interesting future project would be to make the computations more efficient using recent advanced approximations.

Another direction for BRT is to adapt the model and inference to produce more than one tree. For example, co-clustering of both data items and their features simultaneously (a probabilistic version of Hartigan (1972)) might allow a richer interpretation to the data. Alternatively, one might imagine that for some data it is possible to extract multiple hierarchical facets, by producing several equally plausible (though different) hierarchies of the same data.

It might also be possible to further simplify the tree produced by BRT: for large amounts of data it can be unwieldy to interpret the hierarchy produced. A common approach is to "cut" the hierarchy to produce a flat clustering but this necessarily removes the hierarchical information found. Instead one might adapt an approach similar to Aldous et al. (2008) where a coarser sub-hierarchy is extracted in a probabilistic fashion.

Nonparametric Bayesian priors also exist for rose trees (such as Pitman (1999) and Bertoin (2001)) and in future these could be used for non-binary hierarchical clustering that is able to take structure uncertainty into account as well as leverage the nonparametric nature to predict unobserved data items directly.

References

Aldous D, Krikun M and Popovic L 2008 Stochastic models for phylogenetic trees on higher-order taxa. *J. Math. Biology* **56**, 525–557.

Barry D and Hartigan JA 1992 Product partition models for change point problems. *Annals of Statistics* **20**(1), 260–279.

Bertoin J 2001 Homogeneous fragmentation processes. *Probability Theory and Related Fields* **121**(3), 301–318.
Bird R 1998 *Introduction to Functional Programming using Haskell* second edn. Prentice Hall.
Blei DM, Griffiths TL and Jordan MI 2010 The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the Association for Computing Machines* **57**(2), 1–30.
Camin JH and Sokal RR 1965 A method for deducing branching sequences in phylogeny. *Evolution*.
Cree GS and McRae K 2003 Analyzing the factors underlying the structure and computation of the meaning of chipmunk, cherry, chisel, cheese and cello (and many other such concrete nouns). *Journal of Experimental Psychology: General* **132**(2), 163–201.
Duda RO and Hart PE 1973 *Pattern Classification And Scene Analysis*. Wiley and Sons, New York.
Felsenstein J 1973 Maximum likelihood estimation of evolutionary trees from continuous characters. *American Journal of Human Genetics* **25**, 471–492.
Felsenstein J 1981 Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution* **17**, 368–376.
Felsenstein J 2003 *Inferring Phylogenies*. Sinauer Associates.
Fitch WM and Margoliash E 1967 Construction of phylogenetic trees. *Science* **155**, 279–284.
Frank A and Asuncion A 2010 UCI machine learning repository.
Friedman N 2003 Pcluster: Probabilistic agglomerative clustering of gene expression profiles. Technical Report Technical Report 2003-80, Hebrew University.
Girvan M and Newman MEJ 2002 Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* pp. 7821–7826.
Hartigan JA 1972 Direct clustering of a data matrix. *Journal of the American Statistical Association* **67**(337), 123–129.
Hartigan JA 1975 *Clustering Algorithms*. Wiley, New York.
Heller KA 2008 *Efficient Bayesian Methods for Clustering* PhD thesis Gatsby Computational Neuroscience Unit, UCL.
Heller KA and Ghahramani Z 2005 Bayesian hierarchical clustering *Proceedings of the International Conference on Machine Learning*, vol. 22.
Huelsenbeck JP and Ronquist F 2001 MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*.
Jain AK, Murty MN and Flynn PJ 1999 Data clustering: a review. *ACM Comput. Surv.* **31**(3), 264–323.
Kaufman L and Rousseeuw PJ 1990 *Finding Groups in Data*. Wiley, New York.
Kemp C, Griffiths TL, Stromsten S and Tenenbaum JB 2004 Semi-supervised learning with trees *Advances in Neural Information Processing Systems*, vol. 16.
McCallum AK 1996 Bow: A toolkit for statistical language modeling, text retrieval, classifcation and clustering. *http://www.cs.cmu.edu/ mccallum/bow*.
McLachlan GJ and Basford KE 1988 *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York.
Meeds E and Osindero S 2006 An alternative infinite mixture of gaussian process experts *Advances In Neural Information Processing Systems*, vol. 18.
Meertens L 1988 First steps towards the theory of rose trees. Working paper 592 ROM-25, IFIP Working Group 2.1.
Müller P and Quintana F In Press Random partition models with regression on covariates. *Journal of Statistical Inference and Planning*.
Murtagh F 1983 A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal* **26**(4), 254–259.
Neal RM 2003 Density modeling and clustering using Dirichlet diffusion trees *Bayesian Statistics*, vol. 7, pp. 619–629.
Pitman J 1999 Coalescents with multiple collisions. *Annals of Probability* **27**, 1870–1902.
Rasmussen CE and Ghahramani Z 2002 Infinite mixtures of Gaussian process experts *Advances in Neural Information Processing Systems*, vol. 14.
Rogers JS 1997 On the consistency of maximum likelihood estimation of phylogenetic trees from nucleotide sequences. **42**(2), 354–357.
Rosch E, Mervis C, Gray W, Johnson D and Boyes-Braem P 1976 Basic objects in natural categories. *Cognitive Psychology 8* pp. 382–439.
Roy DM, Kemp C, Mansinghka V and Tenenbaum JB 2007 Learning annotated hierarchies from relational data *Advances in Neural Information Processing Systems*, vol. 19.
Saitou N and Nei M 1987 The neighor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution* **4**, 406–425.
Segal E, Koller D and Ormoneit D 2002 Probabilistic abstraction hierarchies *Advances in Neural Information Processing Systems*, vol. 14.
Studier JA and Keppler KJ 1988 A note on the neighbor-joining algorithm of Saitou and Nei. *Molecular Biology and Evolution* **14**, 210–211.
Teh YW, Daume III H and Roy DM 2008 Bayesian agglomerative clustering with coalescents *Advances in Neural Information Processing Systems*, vol. 20, pp. 1473–1480.

Vaithyanathan S and Dom B 2000 Model-based hierarchical clustering *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, vol. 16.

Williams C and Seeger M 2001 Using the Nyström method to speed up kernel methods *Advances in Neural Information Processing*, vol. 13.

Williams CKI 2000 A MCMC approach to hierarchical mixture modelling *Advances in Neural Information Processing Systems*, vol. 12.

Xu Y, Heller KA and Ghahramani Z 2009 Tree-based inference for dirichlet process mixtures *Conference on AI and Statistics*.

Yang Z 1994 Statistical properties of the maximum likelihood method of phylogenetic estimation and comparison with distance matrix methods. *Systematic Biology* **43**, 329–342.

Yang Z and Rannala B 1997 Bayesian phylogenetic inference using DNA sequences: A Markov Chain Monte Carlo method. *Molecular Biology and Evolution* **14**, 717–724.

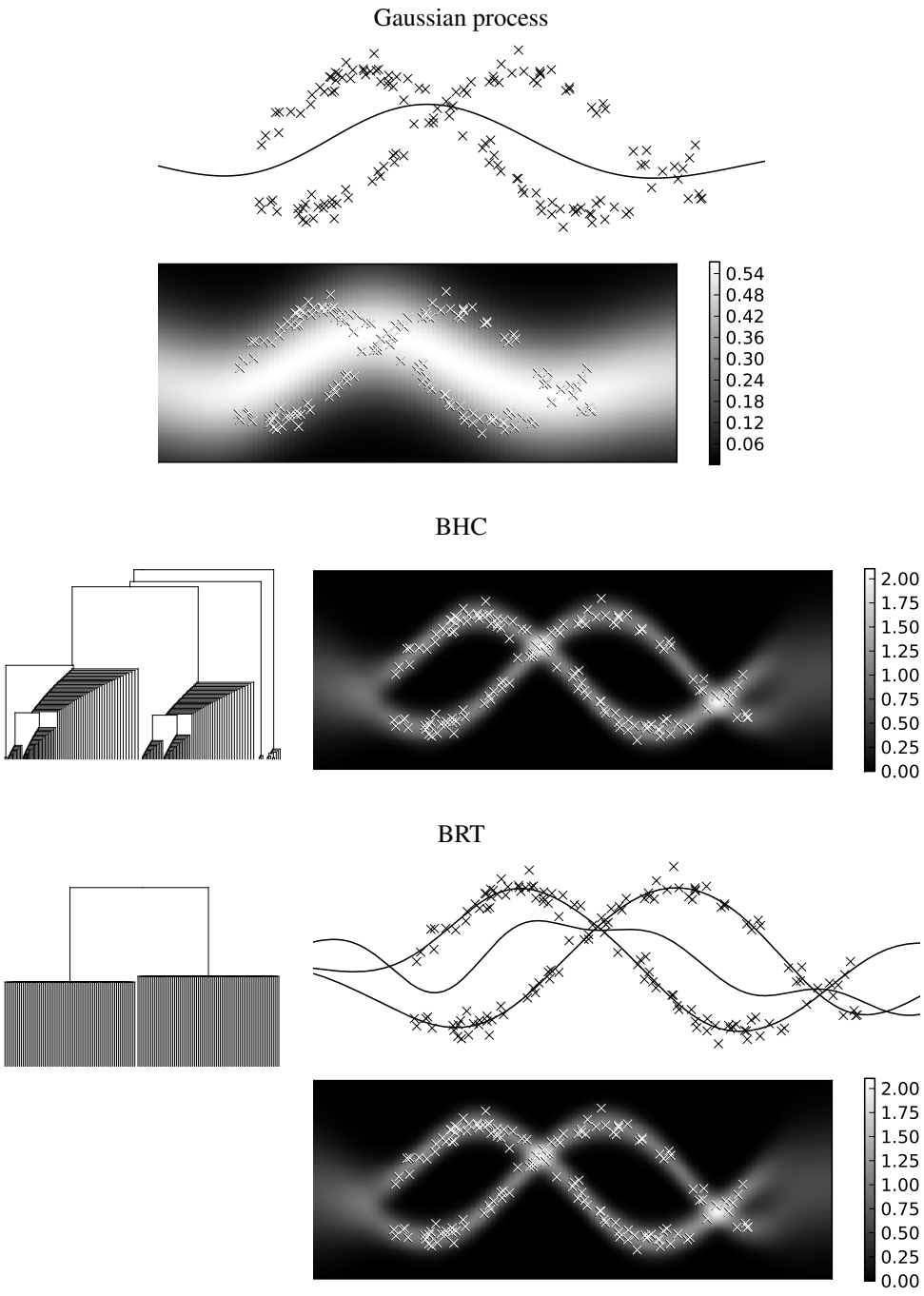


Figure 1.9 A Gaussian process (top) with marginal log likelihood: -1037, on synthetic data of two interlaced curves. Observations are crosses, and line is the posterior mean function of the GP. Background grey scale indicates the predictive density of the GP (scale on left of predictive plot). A BHC mixture of GP experts (middle) with marginal log likelihood: -801, consisting of 149 non-singleton clusters in $7,527,281$ partitions. To the left is the tree found by BHC. Finally, a BRT mixture of GP experts (bottom) with marginal log likelihood: 59, consisting of 3 non-singleton clusters in 5 partitions. Background grey scale indicates corresponding density $p(\mathcal{D}^*|\mathcal{D})$ (as defined in Heller (2008)) via scale on corresponding left of plot. The posterior mean functions passing through the data points correspond to the two subtrees found by BRT, whilst the third posterior mean function corresponds to the GP at root of the tree.