A Bayesian Approach For CART

Hugh Chipman, Edward I. George and Robert E. McCulloch

The University of Chicago and The University of Texas at Austin

Abstract

A Bayesian approach for finding classification and regression tree (CART) models is proposed. By putting an appropriate prior distribution on the space of CART models, the resulting posterior will put higher probability on the more "promising trees". In particular, priors are proposed which penalize complexity by putting higher probability on trees with fewer nodes. Metropolis-Hastings algorithms are used to rapidly grow trees in such a way that the high posterior probability trees are more likely to be obtained. In effect, the algorithm performs a stochastic search for promising trees. Examples are used to illustrate the potential superiority of this approach over conventional greedy methods.

Keywords: binary trees, hierarchical models, Markov chain Monte Carlo, model selection, model uncertainty, stochastic search, mixture models.

1 Introduction

CART models are a flexible method for specifying the conditional distribution of a variable y, given a vector of predictor values x. Such models use a binary tree to subdivide the predictor space into nonoverlapping rectangular regions where the conditional distribution of y is identical for all x values in a given region. This subdivision is obtained by letting each node of the tree correspond to a rectangular subset of the x space. If a node has children, the children constitute a division of the parent node subset into two parts according to whether a particular component of x is greater than or less than some value. Thus the bottom nodes of the tree correspond to the nonoverlapping rectangular regions. Discussions of the CART model may be found in Brieman, Friedman, Olshen and Stone (1984) and Clark and Pregibon (1992).

Given a data set, finding the "best" tree is a difficult problem. Many current methods specify a greedy algorithm for "growing" a tree and then "pruning" it back to avoid overfitting the data. In this paper we propose a Bayesian approach to finding CART models. The approach begins by specifying a prior distribution on the set of CART models. This entails the specification of a prior on the tree space and of a prior on the parameter space of a data model for each tree. Combining this prior with the tree model likelihood yields a posterior distribution on the set of tree models. A feature of this approach is that the prior specification can be used to downweight undesirable model characteristics such as tree complexity or to express a preference for certain predictor variables. In this way the posterior will put higher probability on the "better trees".

Because the number of tree models will be huge in all but trivially small problems, it will rarely be feasible to compute the entire posterior. However, Metropolis-Hastings algorithms can still be used successfully to explore the posterior. Because such algorithms tend to gravitate towards regions of high posterior probability, this exploration will effectively be a stochastic search for good tree models. As opposed to the conventional growing approaches which restrict the search to a "tree sequence", such algorithms search over a much richer class of candidate trees.

Related Bayesian approaches to CART modeling have been considered by Buntine (1992), Denison, Mallick and Smith (1996), Oliver and Hand (1995), and Wallace and and Patrick (1993). Alternative methods which search for promising trees include Sutton (1991) who uses simulated annealing, Jordan and Jacobs (1994) who use the EM algorithm, and Tibshirani and Knight (1995) who use "bootstrap bumping".

The paper is structured as follows. Section 2 describes general prior specification, and Section 3 describes specific priors for conditionally normal data distributions. Section 4 outlines computational strategy for posterior exploration. Section 5 compares the performance of our approach with conventional methods on a simulated and a real example. A more extensive version of this paper can be found in Chipman, George and McCulloch (1996).

2 General Prior Specification

In this section we discuss general prior specification for the CART model. Section 2.1 discusses the CART model in more detail so that the nature of the spaces on which we must place our prior is understood. Our prior is structured so that we first specify the prior distribution on the tree structure of the rectangular partition, and then given the tree, we specify the conditional prior distribution on the set of model parameters corresponding to the bottom nodes of the tree. Section 2.2 discusses the specification of the prior on the tree structure and Section 2.3 discusses the specification of the prior on the model parameters given the tree. Specific priors for normal data are described in Section 3.

2.1 The Structure of a CART Model

As discussed in the introduction, a CART model describes the conditional distribution of y given x, where x is a vector of predictors ($x = (x_1, x_2, \ldots, x_p)$). Such a model can be identified by two main components: a binary tree T, which subdivides the predictor space into rectangular regions, and a parameter Θ, which specifies the conditional distributions of y given x over the regions.

The binary tree T subdivides the predictor space as follows. Each node of the tree corresponds to a rectangular region of the x space. Except for bottom nodes, each node has an associated split variable x_i, split value s, and two nodes below it called the left and right children. These children further subdivide the rectangular region of the parent node by letting the left child correspond to x values such that $x_i \leq s$ and the right child to x values such that $x_i > s$. The bottom nodes thus identify a rectangular partition of the x space. Note that this formulation is general enough to handle arbitrary splitting functions of the form $h(x) \leq s$ versus $h(x) > s$ by simply treating $h(x)$ as another predictor variable.

The distributional specification of the CART model for a given tree T is completed by specifying a probability distribution for y given x in each rectangular region corresponding to a bottom node. It will usually be convenient to let these distributions be members of a common parametric family indexed by a parameter θ. Letting Θ denote list of parameters values for the bottom nodes, the full conditional distribution of y given x is then specified by the CART model $p(y \mid x, \Theta, T)$. The model is called a classification tree when y is a categorical variable. Otherwise, it is called a regression tree.

For illustration, Figure 1 depicts two perspectives

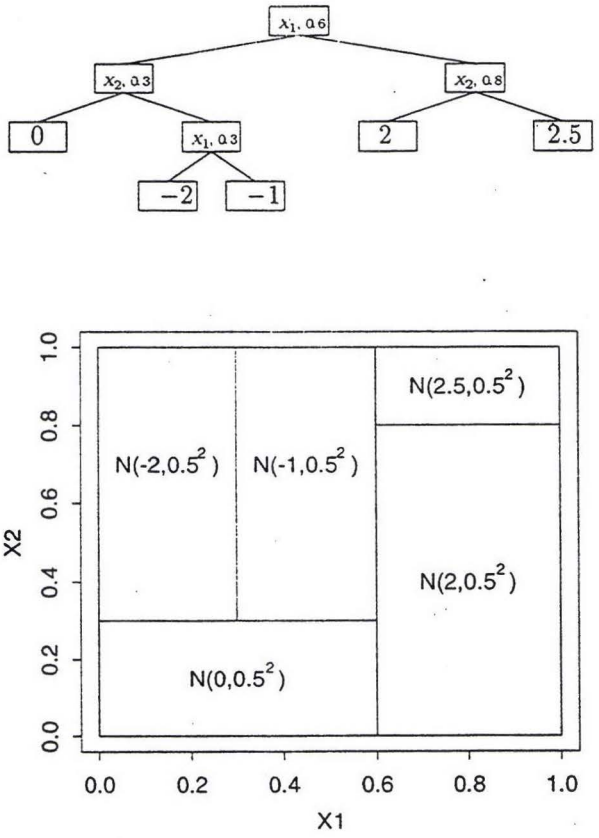

Figure 1: Two perspectives of a CART model

of a CART model $p(y \mid x, \Theta, T)$ for the simple case where $x = (x_1, x_2)$ and $0 \leq x_i \leq 1$. The top figure displays the tree which subdivides the x space, and the bottom figure displays the corresponding subdivision in R^2. This tree has nine nodes of which five are bottom nodes, thus subdividing the x space into five nonoverlapping rectangular regions. At each of the four intermediate nodes we have displayed the split variable x_i and split value s which determine the bottom node regions. For example, the left bottom node is identified with the region of x values for which $x_1 \leq 0.6$ and $x_2 \leq 0.3$. This region corresponds to the lower left box of the bottom figure. Finally, at each bottom node we have displayed the parameter value which identifies the conditional distribution of y given x at that node. In this case, these are values of θ for the simple model $y \sim N(\theta, .5^2)$ which are given as Θ = (0, -2, -1, 2, 2.5). The corresponding distributions are provided in each region of the bottom figure. Note that this is an example of a regression tree model. A classification tree model would be obtained, for example, by using a multinomial distribution at each of the bottom nodes.

Since a CART model is identified by (Θ, T), a Bayesian analysis of the problem proceeds by specifying a prior probability distribution $p(\Theta, T)$. Because Θ indexes the parametric model for each T, it will usually be convenient to use the relationship

$$p(\Theta, T) = p(\Theta \mid T)p(T),$$

and specify $p(T)$ and $p(\Theta \mid T)$ separately. This strategy, which is commonly used for Bayesian model selection (George 1995), offers the advantage that the choice of prior for T does not depend on the form of the parametric family indexed by θ. In particular, the prior on T does not depend on the nature of y. So, for example, the same approach for prior specification of T could be used for y binary or y continuous. Another feature is that conditional specification of the prior on Θ more easily allows for the choice of convenient analytical forms which facilitate posterior computation. Finally, note that $p(\Theta \mid T)$ and $p(T)$ may depend on the input data X, see Buntine (1992), thereby allowing for a richer class of priors.

2.2 Specification of $p(T)$ by a Stochastic Process

Instead of specifying a closed form expression for $p(T)$, the prior on the space of all CART trees, it is more convenient to specify it by a stochastic process which generates trees. Each independent realization of such a process can simply be considered as a random draw from this prior. Furthermore, such a specification still allows for easy evaluation of $p(T)$ for any given tree.

In particular, we consider specification of $p(T)$ by an iterative stochastic process which essentially "grows" trees as follows. Each iteration consists of generating a new tree from the current tree by possibly "splitting" bottom nodes into new bottom nodes. At the end of each iteration, each bottom node of the new tree is also marked *splitable* or *not splitable*. Only nodes marked splitable can split in the next iteration. The process terminates when all bottom nodes have been marked as not splitable.

The tree generating process begins with the trivial tree consisting of a single node marked splitable. At each subsequent iteration, the prior specification is determined by a triplet of probability distributions

$$(p_{split}, p_{var}, p_{val}) \tag{1}$$

which may depend on the tree and X, and may differ at each node. The iteration proceeds by applying the following stochastic process independently to each bottom node marked splitable. First, the node is "split" with probability p_{split}, or is marked not splitable with probability $(1 - p_{split})$. If the node is split, it then becomes a parent node of two new bottom node children, which are marked splitable. Next, the node is stochastically assigned a split variable x_i according to the probability distribution p_{var} on the components of x. Conditionally on the draw of x_i, the node is then stochastically assigned a split value s according to the probability distribution p_{val} on the possible split values of x_i.

As a practical matter, we only consider prior specifications for which the overall set of possible split values is finite. Thus, each p_{val} will always be a discrete distribution. This is hardly a restriction since every data set is necessarily finite, and so can only be partitioned in a finite number of ways. As a consequence, the support of $p(T)$ will always be a finite set of trees.

The stochastic process described above, and hence the prior $p(T)$, is determined entirely by the specification of $(p_{split}, p_{var}, p_{val})$ in (1). There are many interesting possibilities for such a specification. The probability of splitting a node, p_{split}, might depend in a variety of ways upon the complexity of its ancestry or the current tree in general. For example, by setting p_{val} small for nodes with complex ancestry, $P(T)$ can be made to favor simple trees. In choosing a prior for the split variable, p_{var}, we could place higher prior weight on indices corresponding to variables that are thought to be more important. For the prior on the split value, p_{val}, we might expect a region to split more towards its middle than near an edge and so might use a tapered distribution at the extremes.

For illustration, consider the following simple specification of $(p_{split}, p_{var}, p_{val})$ which is applied in Section 5. This specification makes use of the depth of a node which we define as the number of splits above that node. For example, in Figure 1 the five bottom nodes from left to right have depths 2, 3, 3, 2, 2 respectively. Now, for each splitable node, we define the splitting probability by

$$p_{split} \equiv \frac{\alpha}{1 + \beta\gamma^d} \tag{2}$$

When $\beta = 0$, trees with the same number of bottom nodes are assigned the same probability. This is similar to tree prior of DMS for which $P(T)$ is Poisson distribution on the number of bottom nodes. However, when $\beta > 0$, p_{split} will be a decreasing function of node depth d. With such a choice, deeper nodes are less likely to split, so that $p(T)$ will in effect assign lower probability to complex trees. Note that other forms could also be chosen for p_{split} to

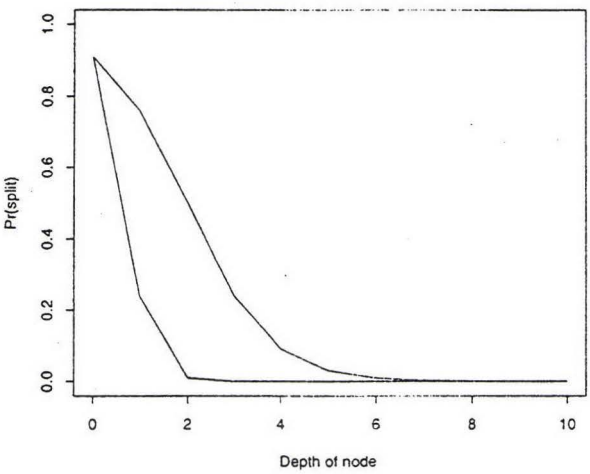

Figure 2: Prior probability that a node will split, as a function of the depth of a node. Upper line is the weak prior with $\gamma = 10^{1/2}$; lower line is the strong prior with $\gamma = 1000^{1/2}$.

allow flexible specification of a decreasing function of d. Figure 2 plots the probability of splitting as a function of d for two sets of parameter values: $(\alpha, \beta, \gamma) = (1, 0.1, 10^{1/2})$ and $(1, 0.1, 1000^{1/2})$. Note that the lower curve was obtained with the larger γ value.

Next, we define the split variable distribution at the split node, p_{var}, to be a discrete uniform distribution on the set of all possible split variables at the split node. Finally, we define the split value distribution p_{val} to be a discrete uniform distribution on the set of possible split values for x_i. Note that the set of possible split variables and values will depend on a node's ancestry. For example, suppose we wanted to iterate our process on the tree in Figure 1, and possible split values for x_1 and x_2 were selected to be the nine values $i/10$, $i = 1, 2, \ldots, 9$. If we considered splitting the rightmost bottom node of that tree, we could split on either x_1 or x_2. If we split on x_1 the possible split values would be (0.7, 0.8, 0.9), whereas if we split on x_2 the only possible split value would be 0.9.

Figures 3a and 3b display the prior distributions on the number of bottom nodes obtained with the weak and strong splitting priors in Figure 2, respectively. Comparing Figures 3a and 3b we see perhaps the basic feature of our prior. By changing the split probability function for p_{split}, we change the prior on the number of bottom nodes which is the number of regions in the final partition.

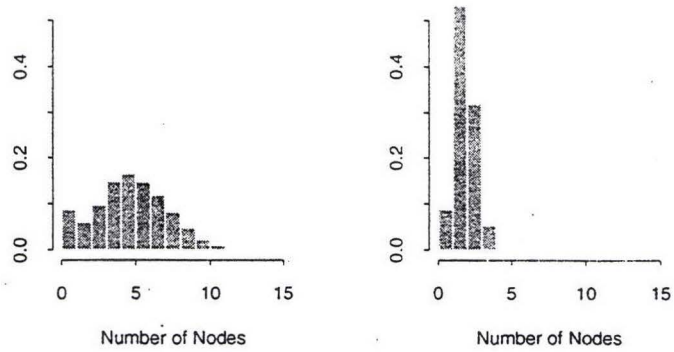

Figure 3: Prior number of nodes, weak (a) and strong (b) priors.

2.3 Specification of $p(\Theta \mid T)$

In this section we discuss the choice of the prior for $\Theta \mid T$. For a given tree T with b bottom nodes, this model is specified by $\Theta = (\theta_1, \theta_2, \ldots, \theta_b)$ which associates the parameter value θ_i with the i^{th} bottom node. If x lies in the region corresponding to the i^{th} bottom node then $y \mid x$ has distribution $f(y \mid \theta_i)$, where we use f to represent a parametric family indexed by θ_i.

Now let y_{ij} denote the j^{th} observation of y in the i^{th} partition (corresponding to the i^{th} bottom node), $i = 1, 2, \ldots, b$, $j = 1, 2, \ldots, n_i$. Define

$$Y \equiv (Y_1, \ldots, Y_b)', \text{ where } Y_i \equiv (y_{i1}, \ldots, y_{in_i})',$$

and define X and X_i analogously. For CART models it is typically assumed that, conditionally on (Θ, T), y values with x values in the same region are iid, and y values across regions are independent. Thus, the CART model distribution for the data will be of the form

$$p(Y \mid X, \Theta, T) = \prod_{i=1}^{b} f(Y_i \mid \theta_i) = \prod_{i=1}^{b} \prod_{j=1}^{n_i} f(y_{ij} \mid \theta_i).$$

Although we emphasize the iid case, note that more general models can be considered at the bottom nodes. For example, one might use regression relationships such as $E(y_{ij} \mid x_{ij}, \theta_i) = x_{ij}\theta_i$ at the ith node. This would allow for modeling the mean of Y by piecewise linear or quadratic functions rather than by constant functions as is implied by the iid assumption.

In choosing a prior, it is crucial to be aware that many choices will be useless in practice because of

the great difficulty of posterior calculation. In Section 4, where we discuss strategies for posterior exploration, it is seen that priors which at least allow for some analytical simplification, can offer tremendous computational advantages. We shall be especially interested in prior forms for $p(\Theta \mid T)$ under which it is possible to analytically integrate out Θ:

$$p(T \mid Y, X) \propto p(T) \int p(Y \mid X, \Theta, T) p(\Theta \mid T) d\Theta. \quad (3)$$

3 Prior Structures for Normal Data

In this section we focus our discussion of prior choice for $\Theta \mid T$ to the case where each y is normal so that $f(y \mid \theta) = N(\mu, \sigma^2)$ with $\theta = (\mu, \sigma)$. Thus,

$$y_{i1}, \ldots, y_{in_i} \mid \mu_i, \sigma_i, T \ \text{ iid} \sim N(\mu_i, \sigma_i^2), \quad (4)$$

$i = 1, \ldots, b$. We propose a variety of prior specifications. In Section 3.1, we discuss independence priors using conjugate forms. In Sections 3.2 and 3.3, we present hierarchical prior specifications which capture prior beliefs about dependence among the μ_i.

3.1 Independence Priors

As discussed in Section 2.3, a simple prior choice is to let the set of $\theta_i = (\mu_i, \sigma_i)$ be iid with the standard conjugate form:

$$\mu_i \mid \sigma_i \sim N(\bar{\mu}, \sigma_i^2/a) \quad (5)$$

and

$$\sigma_i^2 \sim \text{IG}(\nu/2, \nu\lambda/2) \quad (6)$$

(which is equivalent to $\nu\lambda/\sigma_i^2 \sim \chi^2_\nu$). Note that this choice of prior corresponds to a model which is different from the usual CART model in that the value of σ varies as well as that of μ. This specification allows us to model mean *and* variance changes. In practice we may use the observed y values to guide our choices for the prior parameters $(\nu, \lambda, \bar{\mu}, a)$. We choose ν and λ so that some multiple of s_y, the sample standard deviation of the y values, is in the right tail of our prior for σ and some fraction of of s_y is in the left tail.

Of course, it may be inappropriate to use an iid conjugate prior which does not depend on T. For example, if we believe the model is primarily fitting changes in the mean of y, we might expect that for more complex trees (finer partitions of the predictor space) we should have smaller values of σ. In this case, we could let the values of ν and λ depend on T.

Since this prior is of the conjugate form a standard calculation gives:

$$\int f(Y_i \mid X_i, \theta_i) p(\theta_i) d\theta_i = \frac{(\lambda\nu)^{\nu/2}}{\pi^{n_i/2}} \frac{\sqrt{a}}{\sqrt{n_i + a}} \frac{\Gamma((n_i + \nu)/2)}{\Gamma(\nu/2)} (s_i + d_i + \nu\lambda)^{-(n_i+\nu)/2}$$

where $\bar{y}_i$ is the average value in Y_i, s_i is $(n_i - 1)$ times the sample variance of the Y_i values, and $d_i = \frac{n_i a}{n_i + a}(\bar{y}_i - \bar{\mu})^2$.

We may also wish to impose the restriction that σ be the same in each partition in which case our model is more similar to the standard CART approach. In this case the simple prior:

$$\mu_i \mid \sigma \sim N(\bar{\mu}, \sigma^2/a) \quad (7)$$

and

$$\sigma^2 \sim \text{IG}(\nu/2, \nu\lambda/2) \quad (8)$$

may be used. Here, the μ_i are iid given σ. With this choice it is still straightforward to integrate out σ and the set of μ_i obtaining,

$$p(Y \mid X, T) \propto \frac{a^{b/2}}{\prod_{i=1}^b (n_i + a)^{1/2}} \left(\sum_{i=1}^b (s_i + d_i) + \nu\lambda \right)^{-(n+\nu)/2}$$

where s_i and d_i are as above.

3.2 A Hierarchical Prior

Although they are easy to describe and implement, the simple independence priors for $\Theta \mid T$ described in the previous section may not provide enough structure. For example, the independence choice makes the prior on larger sets (large b) of θ values much more diffuse than the prior on smaller sets (small b). This builds into our posterior calculation a preference for smaller trees beyond that expressed in our prior for T. Also, there is a natural intuition that may lead us to believe that there should be prior dependence in the θ values. We may feel that a pair of θ values that correspond to regions which are nearby in the predictor space should be more similar than a pair corresponding to regions which are far apart. Put another way, we may want to incorporate local smoothness of the model surface through our prior.

In the normal case, we may want the μ_i values to be dependent in that μ_i values corresponding to regions in the x space which are it nearby are expected to be *similar*. Thus, for a given tree T, it may be

reasonable to consider that the means $\mu_1, \ldots, \mu_b$ are related across bottom nodes. This idea of local similarity is developed in a non-Bayesian framework by Hastie and Pregibon (1990).

A natural way to model this similarity is to consider the means as arising from a hierarchical Bayesian model. To specify this model, we use the following notation. For the end node with mean μ_i, let $\delta_{i1}, \ldots, \delta_{id(i)}$ be sequence of real-valued mean shifts such that

$$\mu_i = \mu_0 + \sum_{j=1}^{d(i)} \delta_{ij}.$$

The idea is that δ_{ij} represents the additive contribution of the depth j node on the tree path leading to μ_i. Note that the depth of the final node leading to μ_i is $d(i)$. Because of the binary tree structure leading to the bottom nodes, many of the mean shift values δ_{ij} will be identical. Indeed, $\delta_{ij} = \delta_{i'j}$ whenever the paths leading to means μ_i and $\mu_{i'}$ share a node at depth j.

For the equal variance case of (4), where $\sigma_1 = \cdots = \sigma_b = \sigma$, a conjugate prior form for this hierarchical Bayesian model is obtained by putting a zero-mean, normal prior on each of the mean shifts, namely

$$\delta_{ij} \mid T \sim N(0, \sigma^2 v_{ij}), \tag{9}$$

and assuming that for all i, j, i', j', δ_{ij} and $\delta_{i'j'}$ are independent unless $\delta_{ij} = \delta_{i'j}$. It is also assumed that the grand mean μ_0 is independently distributed as $\mu_0 \mid T \sim N(\bar{\mu_0}, \tau_0^2)$. It will usually be convenient to center the values of Y around 0, and set $\bar{\mu_0} = 0$.

Although it may be desirable to set v_{ij} large when little is known about the size of δ_{ij}, we recommend setting v_{ij} to be less than one since δ_{ij} is unlikely to vary by more than $y_1, \ldots, y_n$. Finally, setting v_{ij} small will have the effect of downweighting the posterior probability associated with the corresponding node. One might want to do this for deep nodes in order to put less weight on complex trees.

The mean shift prior structure above induces a multivariate normal prior on the bottom node means, namely

$$\mu \equiv (\mu_1, \ldots, \mu_b)' \mid T \sim N_b(0, \sigma^2 \Sigma_v). \tag{10}$$

The iith diagonal element of Σ_v is $\left(v_0 + \sum_{j=1}^{d(i)} v_{ij}\right)$, and the ii'th off-diagonal element of Σ_v is $\left(v_0 + \sum_{j=1}^{d(i,i')} v_{ij}\right)$, where $d(i,i')$ is the largest value of j for which $\delta_{ij} = \delta_{i'j}$.

To complete the prior specification, we consider the bottom node variance to be a realization from an inverse gamma prior $\sigma^2 \mid T \sim \mathrm{IG}(\nu/2, \nu\lambda/2)$ as in (8). Specification of ν and λ may be guided by the same considerations as for the independence priors. Finally, we assume that μ and σ are a priori independent.

Analytical elimination of μ and σ from

$$p(\mu, \sigma, T \mid Y) \propto p(Y \mid \mu, \sigma, T) p(\mu \mid \sigma, T) p(\sigma \mid T) P(T)$$

for this hierarchical model is feasible. Integrating out μ yields

$$p(\sigma, T \mid y) \propto \sigma^{-(n+\nu+1)} |D_n|^{1/2} |\Sigma_v + D_n|^{-1/2} \exp\left\{-\frac{1}{2\sigma^2}(\nu\lambda + S_y^2)\right\} p(T), \tag{11}$$

where

$$S_y^2 = \sum_{i=1}^{N} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2 + \bar{Y}'(\Sigma_v + D_n)^{-1} \bar{Y}$$

and D_n is the diagonal matrix with iith diagonal element $1/n_i$. Finally, integrating out σ from (11) yields

$$p(T|Y) \propto |D_n|^{1/2} |\Sigma_v + D_n|^{-1/2} (\nu\lambda + S_y^2)^{-(n+\nu)/2} p(T). \tag{12}$$

Note that (12) only gives $p(T \mid Y)$ up to a normalizing constant. To calculate this constant, it would be necessary to evaluate the sum of the right hand side of (12) over all possible values, which is only feasible in trivially small problems. Nevertheless, this expression can provide the basis for fast Monte Carlo search for high posterior probability trees.

4 Extracting Posterior Information

In this section we outline our strategy for evaluating the information in the posterior distribution. In Section 4.1 we discuss our general approach using Metropolis-Hastings algorithms for generating the Markov chains. Section 4.2 describes how we use our chain to search for promising trees.

4.1 The General Approach

The information about the CART model provided by the data is contained in the posterior distribution

$$p(\Theta, T \mid Y, X) \propto p(Y \mid X, \Theta, T) p(\Theta \mid T) p(T). \tag{13}$$

Given that we employ a finite number of split values for each variable, there are a finite number of possible trees. However, there are a great many possible

trees so that is simply not feasible, except in trivially small problems, to integrate and sum the right side of (13) to obtain the normalizing constant. Even if we had a formula for the exact value of $p(\Theta, T \mid Y, X)$ it is not clear how to use it to identify the trees having high posterior probability.

To simplify our problem we note that for certain choices of $p(\Theta \mid T)$, it is possible to analytically integrate out Θ to obtain

$$p(T \mid Y, X) \propto p(Y \mid X, T)p(T). \tag{14}$$

Indeed, the priors discussed in Section 3.1 and 3.3 allow us to do this and still leave us with a rich class of reasonable priors from which to choose. Although evaluation of the norming constant in (14) will rarely be feasible, as long as $p(Y \mid X, T)p(T)$ can be evaluated it will be possible to use a Metropolis-Hastings algorithm to simulate a Markov chain

$$T^0, T^1, T^2, \ldots \tag{15}$$

with limiting distribution $p(T \mid Y, X)$.

To construct such a Metropolis-Hastings algorithm, one needs to specify a Markov transition kernel $q(T, T^*)$ from which trees can be generated, and then $p(Y \mid X, T)p(T)$ evaluated. Starting with an initial tree T^0, the algorithm proceeds by iteratively generating the transition from T^i to T^{i+1} by the two steps:

1. Generate a candidate value T^* with probability distribution $q(T^i, T^*)$.

2. Set $T^{i+1} = T^*$ with probability $\alpha(T^i, T^*) =$

$$\min\left\{\frac{q(T^*, T^i)}{q(T^i, T^*)}\frac{p(Y \mid X, T^*)p(T^*)}{p(Y \mid X, T^i)p(T^i)}, 1\right\}. \tag{16}$$

   Otherwise, set $T^{i+1} = T^i$.

Under weak conditions (see Tierney 1994), the sequence (15) obtained by this algorithm will be a Markov chain with limiting distribution $p(T \mid Y, X)$.

Analogously to our specification of $p(T)$ in Section 2.2, it is useful to specify a transition kernel $q(T, T^*)$ by a stochastic process which can be easily simulated. We have found it useful to use a kernel $q(T, T^*)$ which generates T^* by randomly performing one of the following four modifications to T.

- CHANGE: randomly pick an intermediate node and randomly assign it a new splitting rule

- GROW: randomly pick a bottom node and split it into two new ones by randomly assigning it a new splitting rule

- PRUNE: randomly pick an intermediate node and turn it into a bottom node by collapsing all nodes below it

- SWAP: randomly pick two adjoining intermediate node and swap their splitting rules

The key to an effective specification of a kernel $q(T, T^*)$ based on these modifications are the probabilities with which CHANGE, GROW, PRUNE, SWAP are chosen, and the probabilities with which the implementation of each of these modifications are made. For example, because the CHANGE, GROW, PRUNE steps more often lead to improvements, we have obtained better results by using a kernel which chooses the SWAP step with relatively small probability. We have obtained better results by with kernels which assign higher probability to picking deeper nodes with the CHANGE and PRUNE steps. Modifications at higher nodes are less likely to lead to improvements. Running a Metropolis chain with such modifications also requires much more substantial computational effort because $\alpha(T^i, T^*)$ in (16) requires that we compute $q(T^{i+1}, T^i)$ as well as $q(T^i, T^{i+1})$ for any modification. Finally, we have found that random assignment of the splitting rules in the CHANGE and GROW steps according the same distribution (1) in the prior $p(T)$ leads more rapidly moving Metropolis chains due to higher acceptance rates.

4.2 Searching for Trees

We can run the Metropolis-Hastings algorithm described above to generate a sequence of trees. We start the chain at the trivial initial tree T^0 which consists of a single node. Typically, we hope that as the chain is run, we will move into regions of the parameter space (in this case trees T) which have high posterior probability.

In usual applications of Markov Chain Monte Carlo methods the frequency with which events occur as the chain is run is used to estimate posterior probabilities. In this case, our approach is different because the algorithm tends to get "stuck" in regions near local modes. This happens because q only modifies the lower nodes of a tree, and leaves the higher node structure alone. As a result the chance of completely collapsing a tree, and starting a new one is very low.

Our approach has been to repeatedly restart the chain at T^0, the single node tree. Each time we restart the chain, the algorithm tends to grow trees in a direction of higher posterior probability until the changes in posterior probability begin to stabilize,

suggesting that it is stuck near a local mode. To avoid wasting time waiting for the chain to move away, we instead intervene so that we might find another local mode more quickly. Although we have obtained excellent results by always restarting at the single node tree, it may also be fruitful to restart at different trees. For example, one might restart with the top few levels of other high probability trees, or even at trees found by other heuristic methods. We are currently investigating the potential of restarting at such alternatives.

¿From each run of the chain, the values of $p(Y \mid X, T)p(T)$ allow us to identify trees which have high posterior probability compared to those previously found. We keep track of which visited trees have relatively high posterior probability, but do not keep track of how often they are visited. Our method is thus really a Markov Chain stochastic search rather than a Markov Chain Monte Carlo in that we do not use frequency based estimates of posterior quantities.

Given a set of trees found by repeated runs of our chain, we then rank the trees according their posterior probabilities $p(T \mid Y, X)$. In general our approach only provides a ranking of the trees since we can only compute a quantity which is proportional to $p(T \mid Y, X)$. In practice, if we find a small set of trees that seem to "represent" the support of the posterior, we renormalize the values $p(T \mid Y, X)$ as a way of at least roughly gauging our uncertainty.

5 Examples

In this section, we illustrate the potential of Bayesian CART on a simulated and a real example. For the normal data model of Section 3, we apply the independence priors $P(\Theta \mid T)$ described in Section 3.1, and the prior $p(T)$ described in Section 2.2. To search for trees, we use repeated runs of the Metropolis-Hastings algorithm as described in Section 4. We implemented this algorithm using C++ which, because of its object oriented features, is well suited to handle the computations.

5.1 A small simulated example

This example was obtained by generating 200 iid observations from the tree in Figure 1. This data is sufficient to identify the true tree as a possible model. The bottom nodes with means -2, -1, and 0 are all at least two standard deviations apart from each other and the other nodes. There should be little posterior uncertainty about the corresponding regions. The nodes with means 2 and 2.5 are only one standard deviation apart. Thus we expect more posterior uncertainty about the nature of these two nodes.

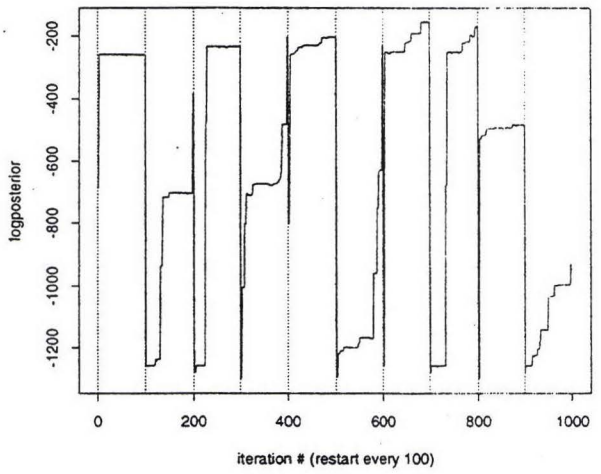

Figure 4: Log posteriors from 10 chains.

The trees we consider may have splits at the values $i/10, i = 1, \ldots, 9$ for either variable. Our prior used p_{split} in (1), with $(\alpha, \beta, \gamma) = (1.0, 0.1, 10^{1/2})$, and the uniform choices for p_{var} and p_{val} described in Section 2.2. This choice of p_{split} corresponds to the prior in Figures 2 and 3 with mass on larger models. We put the (7) prior on μ with mean and variance equal to the sample mean and variance of the response, fixed σ at the true value.

To search through the CART model space we used a simple Metropolis algorithm with a kernel based on the CHANGE, GROW and PRUNE steps. The CHANGE and PRUNE modifications were restricted to nodes which were parents of bottom nodes. The assignment of splitting rules for the CHANGE and GROW steps was identical to that used for the prior $p(T)$. We ran this Metropolis chain 1000 times, restarting each run at the trivial tree T^0. Each run was terminated after 1000 iterations.

To illustrate the behavior of the Metropolis chain, the log posteriors of trees resulting from ten different starts of the chain are displayed in Figure 4. The log posteriors indicate that some runs are dead ends; for example run nine seems to be stuck on a subset of trees having relatively low posterior probabilities. Runs seven and eight appear to be finding relatively interesting trees.

The two most probable models are the true model and the same model except that the nodes with means 2 and 2.5 are combined. The third most prob-

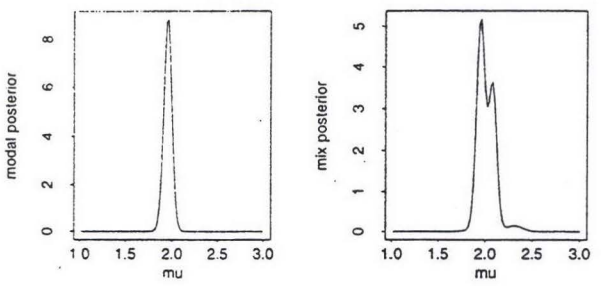

Figure 5: Posterior distribution of the mean of Y at $X_1 = 0.8, X_2 = 0.8$. Left panel is for the modal tree, right panel mixes over the best three trees.

able model is a variant on the true model, with a split on X_2 at 0.7 instead of the true value of 0.8. All remaining trees found by our chain have log posteriors far less than the three top trees. Hence we approximate the posterior distribution by conditioning on these three trees. To illustrate this posterior, we compute the conditional mean of y given $x_1 = 0.8, x_2 = 0.8$. This quantity is a mixture of normal distributions, with weights given by the posterior model probabilities. Figure 5 presents the posterior distribution of μ for this pair of X_1, X_2 values. We present both the distribution conditional on the modal tree and the distribution obtained by mixing over the three most probable trees. The distribution obtained from the mixture reflects the uncertainty associated with the split value between nodes with means 2 and 2.5.

5.2 Mileage example

The second data set is a more substantial and realistic problem. It contains mileage and various other characteristics of 392 cars (after missing values are removed). The data were obtained from the world wide web at the address `http://lib.stat.cmu.edu/datasets/cars.data`. The response is mileage, in miles per gallon. In this example, we use five predictors: number of cylinders, displacement, horsepower, model year, and weight. Here, the predictors have 4, 10, 10, 10, and 12 equally spaced split points, respectively.

For this example we used the same tree prior $p(T)$ as in the previous example. For the parameters, we used the simple independence prior (7) and (8). We chose $\nu = 10, \lambda = 9, \overline{\mu}$ equal to the sample mean of Y, and a equal to λ divided by the sample variance of Y. We chose these values so that the distribution on μ would be roughly the same as the sample distribution of Y.

We also used the same Metropolis algorithm as in the previous example. Here we ran the Metropolis chain 30,000 times, restarting each run at the trivial tree T^0. Each run was terminated after 3000 iterations. The 100 trees with largest posteriors were stored, and are considered here.

When the posteriors for the 100 trees are renormalized, the modal tree accounted for 25% of the posterior mass. The top three trees accounted for 50% of the mass, and were the only trees with individual posterior probabilities of greater than 10%. In fact, the second most probable tree differed only at one node from the most probable tree. The third most probable tree differed more substantially.

The greedy algorithm was used to grow a 23 node tree, which was then pruned back to produce an 11 node tree. For comparison, the greedy algorithm was allowed to choose from the same set of splits used in the Metropolis algorithm. The modal and greedy trees turned out to be quite different. For example the greedy tree never used the predictor displacement to split, while the modal tree used it four times. Even the first split variable of the greedy tree differed from those of the trees found by our procedure. Trees with a first split on the number of cylinders are found in some of the 100 most probable trees, but they receive little mass. The greedy tree was slightly less "bushy", in the sense that it had a wider range of terminal node depths. The trees found by our procedure were more bushy because the prior penalizes growth at a deeper level more heavily.

How can we numerically compare the greedy tree to our trees? One choice would be the residual sum of squares (RSS), the criterion used to fit the greedy tree. This greedy tree had RSS = 3521 compared with 3141 for the modal tree (both have the same number of nodes). Figure 6 plots RSS against the number of nodes in the tree. The line represents the nested sequence of greedy trees, obtained by starting with the largest tree, and pruning the node that reduces the RSS the least. The dots represent the RSS for the trees in our posterior. Notice that all 100 trees in our posterior beat the greedy tree of the same size. The large number of trees that appear better than the greedy tree suggest that this search technique can be far more effective than the greedy method. Some of these trees are quite different from each other, with different first split points. This suggests that not only can the greedy tree be beat, but a number of alternate models provide a better fit to the data.

One challenge that arises from this approach to identifying tree models is the comparison of differ-

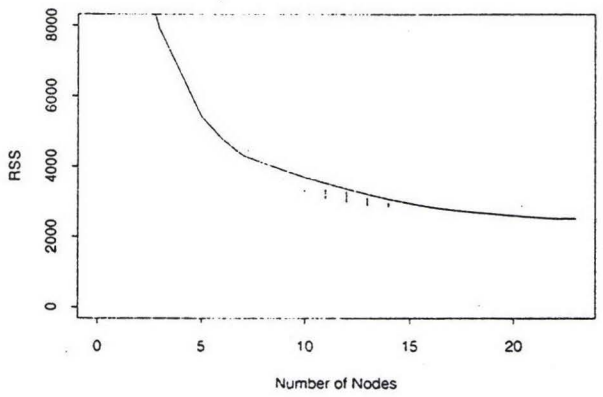

Figure 6: Residual sums of squares for trees of different sizes. Dots represent the trees in the posterior, and the line represents a nested sequence of greedy trees.

ent trees. Now that different trees can be used to describe the same data, a measure of "closeness" for two trees is necessary. For example, the two modal trees are close in the space of trees, because they differ only at one node.

It may also be useful to consider differences in predictions for the given data as a metric on trees. The prediction for a given observation will be the posterior mean for the corresponding node. Two trees will be close if all their predictions are close. Let $\hat{Y}_{i1}$ and $\hat{Y}_{i2}$ be the predictions corresponding to the data point Y_i using trees 1 and 2 respectively. The average absolute difference between these two predictions,

$$D_{12} = \sum_{i=1}^{n} |\hat{Y}_{i1} - \hat{Y}_{i2}|$$

provides a measure of closeness whose units are those of the response. Applied to this example, we found that the two most probable trees were very similar, and all ten posterior trees are quite different from the greedy tree.

6 Discussion

There are two basic and related issues in the development of Bayesian CART models: prior specification and posterior computation. The two issues are related because the effectiveness of our stochastic search for trees with high posterior probability depends on how concentrated the posterior is in the enormous space of possible trees. Because the CART model is so flexible, without prior information, the posterior will be too diffuse for the search to be effective.

In specifying our prior our basic goal is to be able to put high prior probability on simple models in a simple way. We achieve this by specifying a tree generating process in which the conditional probability that a node splits depends on the complexity of its ancestry. In the examples in this paper the complexity of a node's ancestry is measured by its depth. There are many other possible ways to measure ancestor complexity. For example, we could let the probability that a node split be inversely related to the area of the region which is to be split. Both this prior and the one described in Section 2.2 have the common property that the probability of a node splitting depends only on its ancestry. Such priors put relatively high probability on bushy trees, in which nodes have similar depths. As an alternative one might allow the tree to be deep in some areas of the explanatory space. For example, the prior probability could be inversely related to the total number of bottom nodes.

The choice of the prior on Θ is also important. We have proposed a rich and flexible classes of priors which also allow for some posterior simplification by integrating out Θ. This results in substantial computational advantages for posterior exploration.

The Metropolis-Hastings algorithm seems to provide an effective and promising search mechanism. The examples in Section 5 illustrate that it is capable of finding a set of (possibly quite different) trees that fit the data well, and often outperform those trees of similar size found by greedy methods.

We are exploring modifications of our computational approach. For example, it may be more efficient to have our transition probabilities be data dependent. We could compute the posterior of all "nearby" trees using a simple prior and then draw from this posterior to generate a candidate tree. There is also the possibility of considering multiple trees in our current state so that transitions can occur which do just involve changes at the bottom of the tree. We are also exploring models for other response types in addition to the normal model of Section 3.1. Specifically we are adapting the procedure to categorical responses.

Finally, we are confronted with yet another basic issue. How do we report our results? Just reporting the modal tree ignores the goal of capturing the uncertainty. On the other hand you cannot report a large number of trees. The posterior may be spread out over a large number of trees which are quite similar to the modal tree or it may give support to trees

which are quite different from the modal tree. We would like to be able to distinguish between these two cases. This calls for a tree "metric" to tell us when two distinct trees are substantially different from each other. A metric based on predictions, like that proposed in Section 5.2 is one possibility. Others metrics might be defined on the x space only.

Acknowledgements

The authors would like to thank Wray Buntine, Mark Glickman, Augustine Kong, Rob Tibshirani, and Alan Zaslavsky for helpful suggestions. This work was partially supported by NSF grant DMS 94.04408 and Texas ARP grant 003658130.

References

Breiman, L., Friedman, J. Olshen, R. and Stone, C. (1984), *Classification and Regression Trees*, Wadsworth.

Buntine, W. (1992), "Learning Classification Trees", *Statistics and Computing*, 2, 63–73.

Chipman, H., George, E.I. and McCulloch, R.E. (1996), "Bayesian CART", Techinical Report, Graduate School of Business, University of Chicago.

Denison, D., Mallick, B. and Smith, A.F.M. (1996) "Bayesian CART", Technical Report, Department of Mathematics, Imperial College, London.

Clark, L., and Pregibon, D. (1992), "Tree-Based Models" in *Statistical models in S*, J. Chambers and T. Hastie, Eds., Wadsworth.

George, E.I. (1995), "Bayesian Model Selection", Preliminary draft for the *Encyclopedia of Statistical Sciences.*

Green, P. (1995), "Reversible Jump MCMC Computation and Bayesian Model Determination", University of Bristol technical report.

Hastie, T., and Pregibon, L. (1990), "Shrinking Trees", AT&T Bell Laboratories Technical Report.

Jordan, M.I. and Jacobs, R.A. (1994), "Hierarchical Mixtures of Experts and the EM Algorithm", *Neural Computation*, 6, 181-214.

Oliver, J.J. and Hand, D.J. (1995). "On Pruning and Averaging Decision Trees", *Proceedings of the International Machine Learning Conference*, 430-437.

Sutton, C. (1991), "Improving Classification Trees with Simulated Annealing", *Proceedings of the 23rd Symposium on the Interface*, E. Keramidas, Ed., Interface Foundation of North America.

Tibshirani, R., and Knight, K. (1995), "Model Search and Inference by Bootstrap 'Bumping' ", University of Toronto technical report.

Tierney, L. (1994), "Markov Chains for Exploring Posterior Distributions," *Annals of Statistics*, 22, 1701–1762.

Wallace, C.C. and Patrick, J.D. (1993). "Coding decision trees", *Machine Learning*, 11, 7-22.