Bayesian Additive Regression Trees, Computational Approaches

Hugh Chipman, Edward George, Richard Hahn, Robert McCulloch, Matthew Pratola, and Rodney Sparapani [*]

April, 2021

Abstract

Methods based on binary trees play a fundamental role in modern data science. In this paper we give a very focused review of basic Bayesian approaches to tree modeling. Bayesian approaches have some fundamental advantages. Complex models are enhanced with meaningful prior specifications and Markov chain Monte Carlo provides a framework for useful stochastic search of the model space along with some sense of the uncertainty. Bayesian approaches require a specification of a prior in tree space and computation of a high and variable dimension posterior. We provide some computational details that are not readily available in the literature. We also review a few more recent extensions of the of the basic approaches to illustrate the power and potential of the overall approach.

[*]Hugh Chipman is a Professor of Statistics, Department of Mathematics and Statistics, Acadia University, Wolfville, NS, B4P 2R6, `mailto:hugh.chipman@acadiau.ca`. Edward I. George is Professor of Statistics, Department of Statistics, The Wharton School, University of Pennsylvania, Philadelphia, PA, 19104, `mailto:edgeorge@wharton.upenn.edu`. Richard Hahn is Associate Professor of Statistics, The School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ, 85281, `mailto:richard.hahn@asu.edu`. Robert McCulloch is Professor of Statistics, The School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ, 85281, `mailto:robert.mcculloch@asu.edu`. Matthew Pratola is an Associate Professor of Statistics, Department of Statistics, The Ohio State University, Columbus, OH, 43210, `mailto:mpratola@stat.osu.edu`. Rodney Sparapani is an Associate Professor of Biostatistics, Institute for Health and Equity, Medical College of Wisconsin, Milwaukee, WI, 53226, `mailto:rsparapa@mcw.edu`.

Contents

1 Introduction

Our problem is to predict a target variable y give the information in a vector of predictor variables $\mathbf{x}$. Approaches based on trees play a large role in the development of predictive methodology. The classic CART work ([3]), which uses a single tree, is still a very important part of our toolkit. Ensemble methods, which use many trees, such as random forests ([2]) and boosting ([8]) have proven remarkably effective. The XGBoost approach to boosting ([4]) is heavily used in applications. In this paper we explore some of the modeling and computational issues involved in an approach to a Bayesian analysis of tree based models.

The Bayesian approach offers some attractive features. Perhaps most fundamentally priors can be used to express interesting beliefs about complex models. Computation of the posterior motivates interesting explorations of the model space and helps us assess our inferential uncertainty. Multiple Bayesian tree based models may be embedded in larger models using the standard hierarchical modeling framework.

A Bayesian approach requires us formulate a tree model as a parameter, place a prior on the parameter, and define a computable likelihood. Section 2 reviews the approach developed in [6] (hereafter CGM98). Section 3 reviews the Markov chain Monte Carlo (MCMC) algorithm of CGM98 for computing the posterior. The presentation in this article spells out some important details not readily discernible from CGM98. These algorithmic details underlie the implementations in the `R` packages `BART` and `BayesTree` that are both available on the Comprehensive R Archive Network (CRAN): `https://cran.r-project.org`. Section 3 reviews the more recent advances in tree model MCMC due to Pratola ([21]). These algorithms are used in the `R` package `rbart` that is also on CRAN. All three of these `R` packages are based on code written in C++ which is then called from `R`.

Building upon CGM98 and the boosting (in particular [9]), Section 4 reviews the Bayesian approach to ensemble tree modeling developed in [5] (hereafter CGM10). The model developed in CGM10 is known as BART for Bayesian Additive Regression Trees. BART is applied to the classic Boston housing values and air pollution data set in Section 5. Section 6 reviews the MCMC algorithm for posterior computation.

In Section 7 we review two modeling approaches to illustrate the power of Bayesian tree modeling beyond the basic development of CGM98 and CGM10. We make no attempt at a comprehensive review of Bayesian tree models but highlight two examples which we find compelling. Section 7.1 explores the fundamental issues of sparsity and variable selection. Section 7.2 develops computational and modeling approaches which dramatically improve the computational speed of Bayesian approaches making inference with large numbers of observations and predictors feasible. Note the two papers that use multiple BART models to address issues in causal inference are [10] and [19]. Section 8 concludes.

2 Bayesian CART

We begin by laying out the structure of a tree model. This follows the general structure of the usual thing seen in a CART type model but our notation and discussion is targeted towards our ultimate goal of a Bayesian analysis. Our Bayesian approach requires a specification of a prior on tree based model and a Markov chain Monte Carlo (MCMC) algorithm for posterior computation.

Note that while we first cover the basics of modeling and computation for a model based on a single tree, (Sections 2 and 3) this methodology underlies the more powerful BART approach so that a complete understanding of the single tree material is needed to understand BART (Sections 4 and 6).

2.1 A Single Tree Model

A binary tree consists of a set of internal nodes and a set of terminal nodes. We also call the terminal nodes bottom nodes. Each internal node has a binary decision rule associated with it. Internal nodes spawn left and right children, each of which in turn may be internal nodes with decision rules or terminal nodes. Each terminal node has a parameter associated with it. We let $\mathcal{T}$ denote the tree including the decision rules at the interior nodes. Let $\Theta = (\theta_1, \theta_2, \ldots, \theta_b)$ denote the set of parameters at the b bottom nodes.

Given a predictor vector $\mathbf{x}$, you "drop it down the tree" using each decision rule to send $\mathbf{x}$ left or right to the left child node or the right child node. When $\mathbf{x}$ finally lands in a terminal node there is a parameter value awaiting it.

Figure 1 depicts a simple example; the rest of this section will discuss this illustration. The tree has four internal nodes with labels $\{1, 2, 3, 5\}$ and five terminal nodes with labels $\{4, 10, 11, 6, 7\}$. The left child of node i is labeled $2i$ and the right child is labeled $2i + 1$.

Each decision rule is based on a single predictor x_i. The decision rule in node 1 uses x_2. The form of the decision rule depends on whether x_i is numeric or categorical. With numeric variables, we choose a cutpoint c and then go left if $x_i \le c$ and right otherwise. The decision rule in node 2 uses x_1 and $c = 3$. With categorical variables, a decision rule specifies which categories go left and the rest go right. For example, x_2 is categorical with possible values $\{A, B, C, D\}$. The decision rule in node 1 of Figure 1 sends categories $\{C, D\}$ left and categories $\{A, B\}$ right.

It is convenient to have a linear integer index for the bottom node parameters. Our convention is that we number the bottom nodes "left to right". For example, $\Theta = (1, 5, 8, 8, 2)$. We have $\theta_2 = 5$ even though this corresponds to the bottom node with integer label 10. Each predictor vector $\mathbf{x}$ has a corresponding bottom node and we let $\zeta(x)$ be the linear index of the bottom node corresponding to $\mathbf{x}$. So, if $\mathbf{x} = (x_1, x_2) = (4, B)$ then $\zeta(x) = 4$.

2.2 Tree Model Likelihood

The parameter of our model is $(\mathcal{T}, \Theta)$. To obtain our likelihood, we start from a parametric model $Y \sim f(y \mid \theta)$. The idea is that given $(\mathcal{T}, \Theta)$ and $\mathbf{x}$, we drop $\mathbf{x}$ down the tree and then use the θ value in the terminal node $\mathbf{x}$ lands in. If we let $\zeta(\mathbf{x})$ be the index in Θ of the terminal node corresponding to $\mathbf{x}$, then

$$Y \mid \mathbf{x}, (\mathcal{T}, \Theta) \sim f(y \mid \theta_{\zeta(\mathbf{x})}).$$

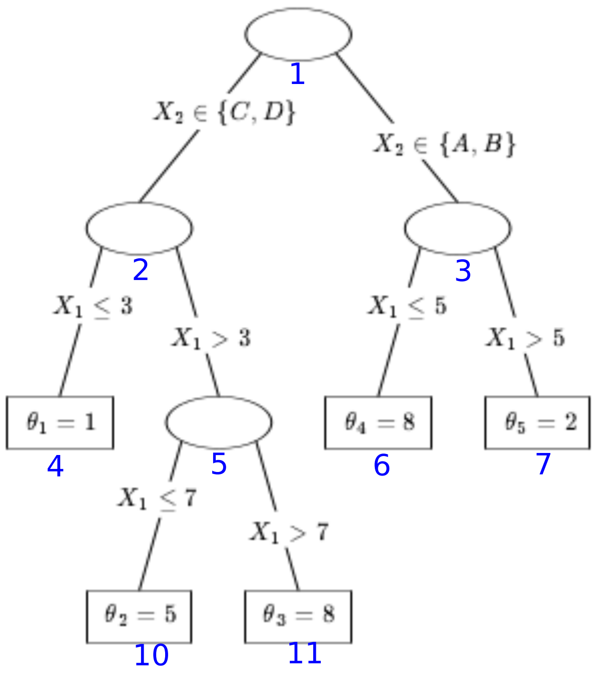

Figure 1: A Bayesian tree.

Given data $(y^k, \mathbf{x}_k)$, $k = 1, 2, \ldots, n$, we can let $\theta^k = \theta_{\zeta(\mathbf{x}_k)}$ so that for $y = (y^1, y^2, \ldots, y^n)$ and $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$,

$$f(y \,|\, \mathbf{x}, (\mathcal{T}, \Theta)) = \prod_{k=1}^{n} f(y^k \,|\, \theta^k)$$

where we assume that the Y^k are independent give the $\{x_k\}$.

It is convenient to let $y_i = \{y^k : \theta^k = \theta_i\}$, the set of y such that x assigns to the i^{th} terminal node. Then we can write our likelihood by multiplying across terminal nodes,

$$f(y \,|\, \mathbf{x}, (\mathcal{T}, \Theta)) = \prod_{i=1}^{b} f(y_i \,|\, \theta_i).$$

Using $y_i = (y_{i1}, y_{i2}, \ldots, y_{iv}, \ldots, y_{in_i})$, where n_i is the number of observations assigned to the i^{th} terminal node, we can also write our likelihood for a given terminal node by multiplying across observations assigned to that node,

$$f(y_i \,|\, \theta_i) = \prod_{v=1}^{n_i} f(y_{iv} \,|\, \theta_i),$$

where we again assume conditional independence.

Three basic examples of such a model are

1. the binary response model $\theta = p$ with $f(y \,|\, p) \sim \text{Bernoulli}(p)$,

2. the mean-variance shift model $\theta = (\mu, \sigma)$ with $f(y \mid \mu, \sigma) \sim N(\mu, \sigma^2)$,

3. the mean-shift model $\theta = \mu$ and $f(y \mid \mu, \sigma) \sim N(\mu, \sigma^2)$, with a common σ across all terminal nodes.

These examples are discussed in CGM98.

2.3 Tree Model Prior

To complete our Bayesian approach, we need to specify a prior on the model parameter $(\mathcal{T}, \Theta)$. Fundamental to our prior is the decomposition

$$p(\mathcal{T}, \Theta) = p(\mathcal{T})\, p(\Theta \mid \mathcal{T}).$$

This decomposition greatly facilitates the prior choice. In particular, note that the dimension of Θ depends on $\mathcal{T}$. Since $\mathcal{T}$ captures the partitioning, and Θ captures parameters within partitions, it seems reasonable to think about $\mathcal{T}$ first and then Θ conditional on $\mathcal{T}$.

2.3.1 $p(\mathcal{T})$

We specify $p(T)$ be describing the process by which a tree $\mathcal{T}$ may be drawn from $p(\mathcal{T})$. We start with a tree which consists of a single node. We then recursively grow the tree by specifying:

- $p_{SPLIT}(\eta, \mathcal{T})$: the probability we *split* terminal node η of tree $\mathcal{T}$ so that it gains left and right node children.

- $p_{RULE}(\eta, \mathcal{T})$: a distribution over the decision rules assignable to the current terminal node η of tree $\mathcal{T}$, should we decide to split it into left and right children.

Given $p_{SPLIT}(\eta, \mathcal{T})$ and $p_{RULE}(\eta, \mathcal{T})$, we randomly grow the tree, recursively, until we have decided not to split each bottom node. Each time we split, we assign the rule by drawing from $p_{RULE}(\eta, \mathcal{T})$. We choose $p_{SPLIT}(\eta, \mathcal{T})$ and $p_{RULE}(\eta, \mathcal{T})$ so that they only depend on the part of $\mathcal{T}$ above the current terminal node η. This ensures that our bottom node splitting process does not depend on the order in which we consider the bottom nodes for splitting.

We let $p_{SPLIT}(\eta, \mathcal{T})$ have the form

$$p_{SPLIT}(\eta, \mathcal{T}) = \frac{\alpha}{(1 + d_\eta)^\beta} \tag{1}$$

where d_η is the depth of node η in tree $\mathcal{T}$, and hyperparameters are $0 < \alpha < 1$ and $\beta > 0$. A single node tree has depth zero. This allows us to express the idea that it gets harder to split as the tree grows. This plays a crucial role in the BART model where we need to express a prior preference for smaller trees. In a single tree model, a value of $\beta = 0.5$ would be reasonable, while in BART $\beta = 2$ is a common default. Interesting alternative enhancements of these choice for $p(\mathcal{T})$ have been proposed by [16, 22, 23].

We now turn to the choice of $p_{RULE}(\eta, \mathcal{T})$. Essentially, the basic default choice is uniform, but taking in account which variables and rules are *available* given η and $\mathcal{T}$. Recall that a predictor is considered to be either numeric or categorical.

For a given categorical x and current bottom node η, the set of available categories are all the categories that have been sent to that bottom node. For example in Figure 1 the categories $\{C, D\}$ are available in bottom nodes 4, 10, and 11. A categorical variable is said to be available in a bottom node if there are at least two categories available in the node.

For a numeric x, a rule is determined by the choice of cutpoint. For each x_i, we initially choose a discrete set of possible cutpoints. Typically we base our choice on the observed values in the training data. Basic choices would be a set of unique x values or quantiles or a uniform grid of value between the min and max. At a bottom node η a subset of the possible splits are available for forming a new rule. For example in Figure 1, you would not consider split value greater less than or equal to 5 for x_1 in terminal node 7, since observations in that bottom node are already restricted to have $x_1 > 5$. Given our initial set of discrete cut points, a choice of numeric predictor, and a bottom node, we can determine the set of available cutpoints. The numeric variable is said to be available if the the set of available cutpoints is non-empty.

We can now define $p_{RULE}(\eta, \mathcal{T})$ by drawing uniformly from the set of available predictors and then uniformly from the set of available rules given the choice of predictor. The `R` package `BayesTree` uses this prior specification for numeric and categorical predictors and much of detail in the underlying C++ code is devoted to determining the availability of variables and rules. The `R` packages `BART` and `rbart` only allow numeric predictors. With only numeric predictors, a categorical variable must be encoded with dummy variables with consequences for the implied prior. Note that unlike in the linear model, K dummies are included for a variable with K categories.

There are many interesting alternative specifications. See for example Section 7.1.

With a discrete set of cutpoints for each numeric variable, $\mathcal{T}$ belongs to a large but discrete set. MCMC steps involving draws of $\mathcal{T}$ will be sampling from a discrete parameter space, and will rely on Metropolis-Hastings proposals (Section 3).

2.3.2 $p(\Theta \mid \mathcal{T})$

Recall that $\Theta = (\theta_1, \theta_2, \ldots, \theta_b)$ where b is the number of bottom nodes in the tree $\mathcal{T}$. A simplifying assumption is prior independence across bottom nodes,

$$p(\Theta \mid \mathcal{T}) = \prod_{i=1}^{b} p(\theta_i).$$

The θ values for the bottom nodes are IID $\theta_i \sim p(\theta)$. With this assumption, we only have to choose the distribution $p(\theta)$.

Our model is (suppressing $\mathbf{x}$)

$$p(y, \mathcal{T}, \Theta) = p(T)\, [\prod_{i=1}^{b} p(\theta_i)]\, [\prod_{i=1}^{b} p(y_i \mid \theta_i)] = p(T)\, \prod_{i=1}^{b} p(\theta_i)\, p(y_i \mid \theta_i).$$

Basic computations are then simplified by choosing $p(y \mid \theta)$ from a standard family and $p(\theta)$ from the corresponding conjugate prior. For example, in BART, θ is just a normal mean so that the (conditionally) conjugate prior is just a univariate normal.

3 Tree MCMC

In this section we outline MCMC approaches for drawing from

$$p(\mathcal{T}, \Theta \mid y) \propto p(\mathcal{T})\, p(\Theta \mid \mathcal{T})\, p(y \mid \mathcal{T}, \Theta)$$

where we have again suppressed $\mathbf{x}$.

Our basic strategy is to integrate out Θ and then use a variety of Metropolis-Hastings (MH) transitions to propose changes to $\mathcal{T}$.

To integrate out Θ, first note

$$p(\mathcal{T} \mid y) \propto p(\mathcal{T})\, p(y \mid \mathcal{T}).$$

Then $p(y \mid \mathcal{T})$ can be computed as

$$\begin{aligned} p(y \mid \mathcal{T}) &\propto \int p(\Theta \mid \mathcal{T})\, p(y \mid \mathcal{T}, \Theta)\, d\Theta \\ &= \int [\prod_{i=1}^{b} p(\theta_i)\, p(y_i \mid \theta_i)] d\theta_1 d\theta_2 \ldots d\theta_b \\ &= \prod_{i=1}^{b} \int p(\theta_i)\, p(y_i \mid \theta_i)\, d\theta_i \\ &= \prod_{i=1}^{b} p(y_i). \end{aligned}$$

With the choice of a conjugate prior, each $p(y_i)$ is computable in closed form. It is just the joint predictive density (or probability mass function) for the subset of observations assigned to bottom node i of the tree $\mathcal{T}$.

This decomposition has important computational benefits. We will draw from $p(\mathcal{T} \mid y)$ using various MH schemes each of which propose changes to a current tree $\mathcal{T}$. When just a part of $\mathcal{T}$ changes, some individual observations will move from one terminal node to another. That is, only a subset of the y_i will change and only the corresponding subset of the integrals $\int p(\theta_i)\, p(y_i \mid \theta_i)\, d\theta_i$ have to be recomputed.

Below, we detail the MH proposals used in CGM98. We will have

- A pair of complementary BIRTH/DEATH moves. In a BIRTH move we propose adding a rule and pair of children to a terminal node. In DEATH move we propose killing a pair of children so that their parent become a terminal node.

- CHANGE Rule move. We propose changing the rule at an interior node.
- SWAP Rule move. We propose swapping the rules for a parent/child pair of interior nodes.

These moves are used in the `R` package `BayesTree`. At each MCMC iteration the BIRTH/DEATH move is chosen with probability 0.5, the CHANGE Rule move is chosen with probability 0.4, and the SWAP Rule is chosen with probability 0.1. Within a BIRTH/DEATH move, BIRTH or DEATH are chosen at random with equal probability, unless one of these moves is not possible (e.g. DEATH for a tree with a single bottom node). Probabilities of BIRTH, DEATH, CHANGE and SWAP are hard coded into the procedure.

Notably, the `R` package `BART` only uses the BIRTH/DEATH move in the marginal $\mathcal{T}$ space and and redraws each θ_i at each MCMC step and still works remarkably well. This is because the BART MCMC works much better than then single tree MCMC.

All of our moves construct a proposed Markov transition. Let $\mathcal{T}_0$ be the current tree and $\mathcal{T}^*$ be the proposed tree which is some modification of $\mathcal{T}_0$. We accept the proposal with Metropolis-Hastings probability

$$\alpha = \min\left\{1, \frac{P(\mathcal{T}^* \mid y)\, P(\mathcal{T}^* \to \mathcal{T}_0)}{P(\mathcal{T}_0 \mid t)\, P(\mathcal{T}_0 \to \mathcal{T}^*)}\right\} \tag{2}$$

where $P(\mathcal{T}_0 \mid y)$ and $P(\mathcal{T}^* \mid y)$ are the posterior probabilities of trees $\mathcal{T}_0$ and $\mathcal{T}^*$ respectively. Thus $P(T \mid y) \propto p(T)\, p(y \mid T)$. $P(P \to \mathcal{T}_0)$ is the probability of proposing $\mathcal{T}_0$ while at $\mathcal{T}^*$, and $P(\mathcal{T}_0 \to \mathcal{T}^*)$ is the probability of proposing $\mathcal{T}^*$ while at $\mathcal{T}_0$. $P(\mathcal{T}_0 \mid y)$ and $P(\mathcal{T}^* \mid y)$ will depend on both the likelihood and our prior, while the transition probabilities depend on the mechanics of our proposal.

Given $\mathcal{T}$ we can easily draw Θ using

$$p(\Theta \mid \mathcal{T}, y) \propto \prod_{i=1}^{b} p(y_i \mid \theta_i) p(\theta_i).$$

Hence, each θ_i may be drawn independently. With the choice of a standard likelihood and conjugate prior, methods for making these draws are typically readily available.

Clearly, the fundamental moves are the BIRTH/DEATH moves. These moves allow trees to grow and shrink in size.

3.1 The BIRTH/DEATH Move

In a BIRTH proposal, a bottom node of the current tree is chosen and we propose to give it a pair of children. A *nog* node of a tree is a node which has children, but no grandchildren. Thus, both children of a nog node are bottom nodes. In a DEATH proposal, we choose a nog node from the current tree and we propose "killing its children". In Figure 1 we might propose a BIRTH at any of the bottom nodes 4,10,11,6, and 7. We could propose a DEATH move at the two nog nodes 5 and 3.

We first describe the BIRTH move in detail. Let $\mathcal{T}_0$ denote the *current* tree and $\mathcal{T}^*$ denote the *proposed* tree. Thus, $\mathcal{T}^*$ differs from $\mathcal{T}_0$ only in that one of the bottom nodes of $\mathcal{T}_0$ has given birth to a pair of children in $\mathcal{T}^*$.

First we discuss the likelihood contribution. As noted above,

$$p(y \,|\, \mathcal{T}) = \prod_{i=1}^{b} p(y_i \,|\mathcal{T}). \tag{3}$$

Thus the contribution of the likelihood to the ratio $P(\mathcal{T}^* \,|\, y)/P(\mathcal{T}_0 \,|\, y)$ in (2) is just

$$\frac{p(y_l, y_r \,|\, \mathcal{T}^*)}{p(y_l, y_r \,|\, \mathcal{T}_0)} = \frac{p(y_l \,|\, \mathcal{T}^*)\, p(y_r \,|\, \mathcal{T}^*)}{p(y_l, y_r \,|\, \mathcal{T}_0)} \tag{4}$$

where y_l denotes the observations in the new left child in $\mathcal{T}^*$, y_r denotes the observations in the new right child in $\mathcal{T}^*$. All other contributions to the likelihoods cancel out because of the product form of (3).

As with the likelihood, much of the prior contributions to the posterior ratio cancel out since the trees differ only at the 2 new bottom nodes and our stochastic tree growing prior draws tree components independently at different "places" of the tree. Hence the prior contribution to the $P(\mathcal{T}^* \,|\, y)/P(\mathcal{T}_0 \,|\, y)$ ratio is

$$\frac{(PG)\,(1 - PGl)\,(1 - PGr)\,P(rule)}{(1 - PG)}, \tag{5}$$

where (the following three bullets are from (1)):

- PG: prior probability of growing at chosen bottom node of $\mathcal{T}_0$,
- PGl: prior probability of growing at new left child in $\mathcal{T}^*$,
- PGr: prior probability of growing at new right child in $\mathcal{T}^*$, and
- $P(rule)$: prior probability of choosing the rule defining the new children in $\mathcal{T}^*$, given by p_{RULE}.

We draw the candidate rule for $\mathcal{T}^*$ by drawing from the prior so that $P(rule)$ is given by $p_{RULE}(\eta, \mathcal{T}_0)$ where η is the bottom node we have randomly chosen for a potential birth.

Finally, the ratio $P(\mathcal{T}^* \to \mathcal{T}_0)/P(\mathcal{T}_0 \to \mathcal{T}^*)$, is given by

$$\frac{(PD)\,(Pnog)}{(PB)\,(Pbot)\,P(rule)}, \tag{6}$$

where

- PD: probability of choosing the death proposal at tree $\mathcal{T}^*$.

- *Pnog*: probability of choosing the nog node that gets you back $\mathcal{T}_0$.
- *PB*: probability of choosing a birth proposal at $\mathcal{T}_0$.
- *Pbot*: probability of choosing the $\mathcal{T}_0$ bottom node such that a birth gets you to $\mathcal{T}^*$.
- $P(rule)$: probability of drawing the new splitting rule to generate $\mathcal{T}^*$'s children.

Our proposal draw of the new rule generating the two new bottom nodes is a draw from the prior. It is in this draw that variable selection (or, perhaps, variable proposal) occurs! Note that since our proposal for the *rule* is a draw from the prior, it cancels out in the ratio (2).

The final MH ratio used for BIRTH is

$$\min\left\{1, \frac{(PG)(1-PGl)(1-PGr)}{(1-PG)}\,\frac{(PD)(Pnog)}{(PB)(Pbot)}\,\frac{p(y_l \mid \mathcal{T}^*)\,p(y_r \mid \mathcal{T}^*)}{p(y_l, y_r \mid \mathcal{T}_0)}\right\}.$$

The formulas given above correspond exactly to the C++ source code in the `R` packages `BayesTree` and `BART`.

For a DEATH move, we choose a nog node of $\mathcal{T}_0$ and propose killing the two children to create $\mathcal{T}^*$. The MH acceptance probability is

$$\min\left\{1, \frac{(1-PG)(PB)(Pbot)}{(PG)(1-PGl)(1-PGr)(PD)(Pnog)}\,\frac{p(y_l, y_r \mid \mathcal{T}^*)}{p(y_l \mid \mathcal{T}_0)\,p(y_r \mid \mathcal{T}_0)}\right\},$$

where

- *PG*: prior probability of spawning children at the proposed new bottom node of $\mathcal{T}^*$ (nog node of $\mathcal{T}_0$).
- *PB*: probability of a BIRTH move at $\mathcal{T}^*$.
- *Pbot*: probability of choosing the the bottom node of $\mathcal{T}^*$ such that a birth gets you back to $\mathcal{T}_0$.
- *PGl*: prior probability of adding children at the proposed left child of $\mathcal{T}_0$.
- *PGr*: prior probability of adding children at the proposed right child of $\mathcal{T}_0$.
- *PD*: probability of a DEATH move at $\mathcal{T}_0$.
- *Pnog*: probability the choosing the nog node at $\mathcal{T}_0$.

3.2 CHANGE Rule

The CHANGE Rule move picks an interior node and then modifies the current tree by changing the decision rule at the chosen node. Our transition $P(\mathcal{T}_0 \to \mathcal{T}^*)$ is made up of the steps:

1. Draw node η from $\mathcal{T}_0$ by drawing uniformly from the set of interior nodes.
2. Draw a rule from $p_{RULE}(\eta, \mathcal{T}_0)$.
3. Replace the decision rule at node η of $\mathcal{T}_0$ with the rule drawn in the second step to obtain $\mathcal{T}^*$.

After we draw $\mathcal{T}^*$, we check that the resulting tree has nonzero prior probability. For example, our prior does not allow logically empty bottom nodes since rules are always checked to be drawn using available variables. If $\mathcal{T}^*$ is such that $p(\mathcal{T}^*)$ is 0, then we can immediately reject the move without further computation.

The number of interior nodes in $\mathcal{T}_0$ and $\mathcal{T}^*$ are the same and each interior node of each tree clearly has available variables (otherwise it could not have a splitting rule). Also recall that $p_{RULE}(\eta, \mathcal{T})$ only depends on the part of $\mathcal{T}$ above η in $\mathcal{T}$. Hence we have the property that $P(\mathcal{T}_0 \to \mathcal{T}^*) = P(\mathcal{T}^* \to \mathcal{T}_0)$ so that the ratio cancels in the MH acceptance ratio.

$$\alpha = \min\left\{1, \frac{p(\mathcal{T}^*)\, p(y \mid \mathcal{T}^*)}{p(\mathcal{T}_0)\, p(y \mid \mathcal{T}_0)}\right\}.$$

To compute $p(y \mid \mathcal{T})$ for either of $\mathcal{T}_0$ or $\mathcal{T}^*$, we only have to consider observations in bottom nodes below η since the contributions for other bottom nodes will cancel.

3.3 SWAP Rule

In the SWAP Rule step, we randomly pick a parent-child pair that are both internal nodes. We then swap their splitting rules. If both children have the identical rule, we swap the splitting rule of the parent with both children.

Similar to the CHANGE Rule proposal, a key observation is that the proposal step for SWAP is symmetric. The general expresssion of the MH acceptance probability is as in (2). For SWAP, the proposal distributions $P(T_0 \to T^*)$ and $P(T^* \to T_0)$ will cancel in (2). Only the likelihood and prior terms need to be calculated.

The proposal for SWAP is a draw (with equal probability) from the list of interior nodes having at least one child that is nonterminal. This list constitutes the parents of the swap. For each parent there will be at least one child with a rule that could be swapped with the parent. Once a parent is chosen, the 2 children are inspected. If only 1 child is nonterminal, that child will be the one chosen for the SWAP. If both children are nonterminal and they have different rules, then 1 of the 2 children will be chosen (with equal probability) for the swap. If both children have identical rules, then the parent rule and the child rules are swapped, and both children get the parent rule.

Once the proposal has chosen a parent-child pair to swap, the rules are swapped and the resulting tree checked to determine if the swap produces any necessarily empty terminal nodes. If there are necessarily empty terminal nodes, this corresponds to a proposed tree T^* with prior probability 0 and thus the MH step will not accept. This check can be carried out without referring to the data, since only the rules of T_0 and T^* need to be checked.

Assuming that the proposal does not have 0 prior probability, then the prior probabilities for T^* and T_0 are calculated for the entire trees. Although there is cancellation in the ratio of prior terms for parts of the tree that do not change, the prior computation is relatively quick and so is simply carried out for the full trees.

The calculation of likelihood for T^* requires re-assignment of data among all bottom nodes that are below the parent. The likelihoods can be calculated for subsets of T^* and T_0, for all bottom nodes below the parent of the proposal. The 2 likelihood values and 2 prior values are sufficient to evaluate α in (2).

If the SWAP proposal is not accepted, then the tree is restored to T_0. If the proposal is accepted, the change to the tree has already been made (to allow computation of prior and likelihood at T^*).

3.4 Improved Tree Space Moves

As is well known, the proposal distribution is a key user-specified parameterization of the Metropolis-Hastings MCMC algorithm that has a large effect on how well, and how efficient, MH sampling can be performed. In the best case scenario, draws from the true posterior are directly available giving an acceptance ratio of 1. In practice, a distribution that is simple to draw from is used as the proposal. This leads to an algorithm that is practically implementable, but uses a proposal having only moderate accuracy (often only locally) to the true posterior, leading to many rejected (i.e. wasted) samples and slower convergence. Nonetheless, the practical usefulness of MH has lead to its widespread adoption.

The situation becomes more challenging in the modern setting where one is interested in performing Bayesian inference for complex, high-dimensional models. In CGM98 and CGM10, a pragmatic approach for the case of Bayesian regression trees (a complex, high-dimensional model) was taken by designing the proposals described above that explore tree-space by incrementally making the model just slightly more or less complex (via BIRTH or DEATH at a single terminal or nog node respectively) or just slightly adjusting an existing tree's ruleset (via CHANGE or SWAP at a single node or pair of nodes respectively). However, in some settings, it has been recognized that this proposal distribution may lead to slow convergence and/or inaccurate sampling – an issue of eminent practical relevance even if required properties for the asymptotic convergence of MH sampling are satisfied.

Good alternatives to the CGM98 algorithm are not neccesarily obvious since one would like to retain the simplicity, locality and efficiency of the algorithm. Recent work has provided some alternatives and refinements at moderate increases in computational cost when a problem demands more effective sampling of the posterior. [21] introduces a new ROTATE proposal, defines a PERTURB proposal as a refined version of CHANGE, and also revises the basic MCMC loop, as described in Algorithm 1 below. Algorithm 1 is for the mean-shift model, which will be a building block for BART in Section 4.

Algorithm 1 Updated Bayesian CART MCMC Algorithm

```
procedure BAYESIAN CART-ITERATION($\mathbf{y}, \mathbf{X}$, num_trees)
output An approximate draw from the tree posterior
    Draw $\mathcal{T}|\sigma^2, \mathbf{y}$ via BIRTH, DEATH or ROTATE at one random eligible internal node
    Set num_internal = number of internal nodes of tree $\mathcal{T}$
    Set num_terminal = number of terminal nodes of tree $\mathcal{T}$
    for $j$ in 1 to num_internal do
        Draw rule $(v_j, c_j)|\mathcal{T}, \sigma^2, \mathbf{y}$ via PERTURB
    for $j$ in 1 to num_terminal do
        Draw $\mu_j|\mathcal{T}, \mathbf{y}$ via Gibbs
    Draw $\sigma^2|\mathcal{T}, \Theta, \mathbf{y}$ via Gibbs
return
```

3.4.1 Rotate

Like SWAP, ROTATE maps the existing internal structure of a tree into a plausible alternative (i.e. one that could have been generated by a longer sequence of BIRTH/DEATH proposals). But while SWAP only considers 1 or 2 possible alternatives, ROTATE generates a larger (stochastic) set of possible transitions. Unlike SWAP, ROTATE also considers the descendants of the ROTATE node in forming the possible transitions, and the further up (down) the tree, the more (less) possible ROTATE transitions there are. If one thinks of BIRTH/DEATH as the simplest possible 'rearrangement' of a tree, ROTATE can then be thought of as generalizing the ideas of SWAP, BIRTH and DEATH in an elegant way to arbitrary internal locations of a tree. For instance, while BIRTH/DEATH involves the likelihood contributions for y_l, y_r, ROTATE involves the likelihood contributions for the data involved in the left/right subtrees of the ROTATE proposal, say y_{T_l}, y_{T_r}. Heuristically then, ROTATE is a less local proposal than BIRTH/DEATH and more diverse than SWAP, but not so global nor so diverse as to be too inefficient. Finally, ROTATE is its own inverse, making application of this algorithmically-generated proposal distribution practically tangible.

3.4.2 Perturb

Similar to CHANGE, PERTURB aims to update the rules in an existing tree. This is done in two ways: updating the cutpoints, or updating the (variable,cutpoint) pairs. Note that PETURB is applied to all nodes in a tree, leading to more efficient exploration of this aspect of the posterior distribution. This is made possible by more efficient generation of cutpoint proposals, which are conditioned on both the ancestral and descendant parts of the tree for the node being updated. Similarly, variable proposals are made more efficient by using a preconditioned proposal distribution; [21] suggest using a correlation metric such as Spearman rank correlation to form the preconditioned transition matrix, although other choices are possible. Note also that both variants of PETURB can simultaneously update all internal nodes that are at the same tree depth, thereby exploiting parallelism to make such computations more efficient.

3.4.3 The Complex Mixtures that are Tree Proposals

Modifying the individual proposals as described above only goes part of the way to ameliorating Bayesian Tree MCMC algorithms. Part of the tale is in how smartly these propoals are used. Recall that for BIRTH/DEATH, the particular proposal selected from either of these choices is determined by the flip of an equally weighted coin. And the corresponding terminal or nog node selected for the chosen move is also randomly drawn with equal weight. But why not prefer a BIRTH in shallower parts of tree-space, or a DEATH in deeper parts of tree space? Similarly, in the BIRTH/DEATH/ROTATE mixture, should these be equally weighted or should one proposal be preferred depending on the state of the tree? Such issues are very much non-trivial, and would lead away from the simple, pragmatic, proposal distributions that have seen so much success. One alternative is to leverage parallel computation to explore a large set of possible transitions to avoid devising a clever strategy to determine what the mixture ought to be at any given iteration of the algorithm. Such is the strategy of [20], who use the BD-MCMC algorithm to select amongst all possible BIRTH/DEATH moves (or BIRTH/DEATH/ROTATE moves) at a rate that is proportional to their posterior probability rather than the default (weighted) mixture. While this increases the number of required computations needed at each step of the MCMC, such computations can be largely hidden via effective parallelization, resulting in more efficient sampling of the posterior per unit time.

4 The BART Model

BART (Bayesian Additive Regression Trees, CGM10) builds on the Bayesian analysis of a single tree to consider an ensemble of trees. BART is inspired by Friedman's work ([9]) on boosting but uses the power of the Bayesian machinery.

To transition from the single tree development of Section 2, we start with a single tree but let $\theta = \mu$ be a single mean parameter. Rather than using Θ to denote the collection of bottom node parameters, we will switch notation to $\mathcal{M} = (\mu_1, \mu_2, \ldots, \mu_b)$, a collection of mean parameters for the bottom nodes.

We then define the function $g(\mathbf{x}; \mathcal{T}, \mathcal{M})$ to be $\mu_{\zeta(\mathbf{x})}$ where ζ is as in Section 2. That is, we drop $\mathbf{x}$ down the tree $\mathcal{T}$ until it lands in a bottom node and finds a μ_i awaiting it, which is then the value of g. Clearly g looks like a step function corresponding to the classic regression tree of classic CART.

We can turn a single tree model indexed by parameter $(\mathcal{T}, \mathcal{M})$ into a probability model with a likelihood by adding an error term,

$$Y = g(\mathbf{x}; \mathcal{T}, \mathcal{M}) + \epsilon, \;\; \epsilon \sim N(0, \sigma^2).$$

BART follows Friedman (and more generally the boosting literature) by replacing the single tree mean model $g(\mathbf{x}; \mathcal{T}, \mathcal{M})$ with a sum of m trees $f(\mathbf{x}) = \sum_{j=1}^{m} g(\mathbf{x}; \mathcal{T}_j, \mathcal{M}_j)$.

$$Y = f(\mathbf{x}) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2) \qquad \text{where } f \overset{\text{prior}}{\sim} \text{BART}. \tag{7}$$

As in Section 2, each $\mathcal{T}_j$ is a recursive binary regression tree. $\mathcal{M}_j$ contains the terminal node constants μ_{ij}, for which $g(\mathbf{x}; \mathcal{T}_j, \mathcal{M}_j)$ is the step function which assigns $\mu_{ij} \in \mathcal{M}_j$ to $\mathbf{x}$ according to the sequence of splitting rules in $\mathcal{T}_j$.

For each value of $\mathbf{x}$, under (7), $E(Y \mid \mathbf{x})$ is equal to the sum of all the terminal node μ_{ij}'s assigned to $\mathbf{x}$ by the $g(x; \mathcal{T}_j, \mathcal{M}_j)$'s. Thus, the sum-of-trees function is flexibly capable of approximating a wide class of functions from R^n to R, especially when the number of trees m is large. Note also that the sum-of-trees representation is simply the sum of many simple multidimensional step functions from R^n to R, namely the $g(\mathbf{x}; \mathcal{T}_j, \mathcal{M}_j)$, rendering it much more manageable than basis expansions with more complicated elements such as multidimensional wavelets or multidimensional splines.

The BART model specification is completed by introducing a prior distribution over all the parameters of the sum-of-trees model, namely $(\mathcal{T}_1, \mathcal{M}_1), \ldots, (\mathcal{T}_m, \mathcal{M}_m)$ and σ. Note that $(\mathcal{T}_1, \mathcal{M}_1), \ldots, (\mathcal{T}_m, \mathcal{M}_m)$ entail all the bottom node parameters as well as the tree structures and splitting rules, a very large number of parameters, especially when m is large. To cope with this parameter explosion, we use a "regularization" prior that effectively constrains the fit by keeping each of the individual tree effects from being unduly influential. Without such a regularizing influence, large tree components would overwhelm the rich structure of (7), thereby limiting its scope of fine structure approximation.

4.1 Specification of the BART Regularization Prior

To simplify the specification of this regularization prior, we restrict attention to symmetric independence priors of the form

$$p((\mathcal{T}_1, \mathcal{M}_1), \ldots, (\mathcal{T}_m, \mathcal{M}_m), \sigma) = \left[\prod_j \left(\prod_i p(\mu_{ij} \mid \mathcal{T}_j) \right) p(\mathcal{T}_j) \right] p(\sigma), \tag{8}$$

where $\mu_{ij} \in \mathcal{M}_j$, thereby reducing prior specification to the choice of prior forms for $p(\mathcal{T}_j), p(\mu_{ij} \mid \mathcal{T}_j)$ and $p(\sigma)$. To simplify matters further we use identical prior forms for every $p(\mathcal{T}_j)$ and for every $p(\mu_{ij} \mid \mathcal{T}_j)$. As detailed below, each of these prior forms are controlled by just a few interpretable hyperparameters that can be calibrated to yield surprisingly effective default specifications for regularization of the sum-of-trees model.

For $p(\mathcal{T}_j)$, we use the prior developed in Section 2. Note however that the values for α and β are typically very different in BART. In BART we often use $\alpha = .95$ and $\beta = 2$ whereas with a single tree we use a much smaller β. This expresses the idea that we do not expect the individual trees to be large.

For $p(\mu_{ij} \mid \mathcal{T}_j)$, we use the conjugate normal distribution $\mathcal{N}(\mu_\mu, \sigma_\mu^2)$ which allows μ_{ij} to be marginalized out as in Section 3, vastly simplifying MCMC posterior calculations. To guide the specification of the hyperparameters μ_μ and σ_μ, we note that under (7), it is highly probable that $E(Y \mid \mathbf{x})$ lies between y_{min} and y_{max}, the minimum and maximum of the observed values of Y in the data, and that the prior distribution of $E(Y \mid \mathbf{x})$ is $\mathcal{N}(m\,\mu_\mu, m\,\sigma_\mu^2)$, (because $E(Y \mid \mathbf{x})$ is the sum of m independent μ_{ij}'s under the sum-of-trees model). Based on these facts, we use the informal empirical Bayes strategy of choosing μ_μ and σ_μ so that $\mathcal{N}(m\,\mu_\mu, m\,\sigma_\mu^2)$ assigns substantial probability to the interval (y_{min}, y_{max}). This is conveniently done by choosing μ_μ and σ_μ so that $m\,\mu_\mu - k\,\sqrt{m}\,\sigma_\mu = y_{min}$ and $m\,\mu_\mu + k\,\sqrt{m}\,\sigma_\mu = y_{max}$ for some preselected value of k such as 1, 2 or 3. For example, $k = 2$ would yield a 95% prior probability that $E(Y \mid \mathbf{x})$ is in the interval (y_{min}, y_{max}). The goal of this specification strategy for μ_μ and σ_μ is to ensure that the implicit prior for $E(Y \mid \mathbf{x})$ is in the right "ballpark" in the sense of assigning substantial probability to the entire region of plausible values of $E(Y \mid \mathbf{x})$ while avoiding overconcentration and overdispersion of the prior with respect to the likelihood. As long as this goal is met, BART seems to be very robust to variations of these specifications.

For $p(\sigma)$, we also use a conjugate prior, here the inverse chi-square distribution $\sigma^2 \sim \nu\,\lambda/\chi^2_\nu$. Here again, we use an informal empirical Bayes approach to guide the specification of the hyperparameters ν and λ, in this case to assign substantial probability to the entire region of plausible values of σ while avoiding overconcentration and overdispersion of the prior. Essentially, we calibrate the prior df ν and scale λ with a "rough data-based overestimate" $\hat{\sigma}$ of σ. Two natural choices for $\hat{\sigma}$ are (i) a "naive" specification, the sample standard deviation of Y, or (ii) a "linear model" specification, the residual standard deviation from a least squares linear regression of Y on all the predictors. We then pick a value of ν between 3 and 10 to get an appropriate shape, and a value of λ so that the qth quantile of the prior on σ is located at $\hat{\sigma}$, that is $P(\sigma < \hat{\sigma}) = q$. We consider large values of q such as 0.75, 0.90 or 0.99 to center the distribution below $\hat{\sigma}$.

5 BART Example: Boston housing values and air pollution

Here, we demonstrate BART with the classic Boston housing example [12]. This data is based on the 1970 US Census where each observation represents a Census tract in the Boston Standard Metropolitan Statistical Area. For each tract, there was a localized air pollution estimate, the concentration of nitrogen oxides, `nox`, based on a meteorological model that was calibrated to monitoring data. Restricted to tracts with owner-occupied homes, there are $N = 506$ observations. We'll predict the median value of owner-occupied homes (in thousands of dollars), `mdev`, by thirteen covariates including `nox` which is our primary interest.

However, BART does not directly provide a summary of the effect of a single covariate, or a subset of covariates, on the outcome. Friedman's partial dependence function [9] can be employed with BART to summarize the marginal effect due to a subset of the covariates, $\boldsymbol{x}_S$, by aggregating over the complement covariates, $\boldsymbol{x}_C$, i.e., $\boldsymbol{x} = [\boldsymbol{x}_S, \boldsymbol{x}_C]$. The marginal dependence function is defined by fixing $\boldsymbol{x}_S$ while aggregating over the observed settings of the complement covariates in the data set: $f(\boldsymbol{x}_S) = N^{-1}\sum_{i=1}^{N} f(\boldsymbol{x}_S, \boldsymbol{x}_{iC})$. For example, suppose that we want to summarize `mdev` by `nox` while aggregating over the other twelve covariates in the Boston housing data. In Figure 2, we demonstrate the marginal estimate and its 95% credible interval: notice that BART has discerned a complex non-linear relationship between `mdev` and `nox` from the data. N.B. this example including data and source code can be found in the `BART R` package [24] as the `nox.R` demonstration program.

6 BART MCMC

Combining the regularization prior with the likelihood, $L((\mathcal{T}_1, \mathcal{M}_1), \ldots, (\mathcal{T}_m, \mathcal{M}_m), \sigma \mid y)$ induces a posterior distribution

$$p((\mathcal{T}_1, \mathcal{M}_1), \ldots, (\mathcal{T}_m, \mathcal{M}_m), \sigma \mid y) \tag{9}$$

over the full sum-of-trees model parameter space. Here y is the observed $n \times 1$ vector of Y values in the data which are assumed to be independently realized. Note also that here and below we suppress explicit dependence on $\mathbf{x}$ as we assume $\mathbf{x}$ to be fixed throughout. Although analytically intractable, the following backfitting MCMC algorithm can be used to very effectively simulate samples from this posterior.

This algorithm is a Gibbs sampler at the outer level. Let $\mathcal{T}_{(j)}$ be the set of all trees in the sum *except* $\mathcal{T}_j$, and similarly define $\mathcal{M}_{(j)}$, so that $\mathcal{T}_{(j)}$ will be a set of $m-1$ trees, and $\mathcal{M}_{(j)}$ the associated terminal node parameters. A Gibbs sampling strategy for sampling from (9) is obtained by m successive draws of $(\mathcal{T}_j, \mathcal{M}_j)$ conditionally on $(\mathcal{T}_{(j)}, \mathcal{M}_{(j)}, \sigma)$:

$$(\mathcal{T}_j, \mathcal{M}_j) \mid T_{(j)}, \mathcal{M}_{(j)}, \sigma, y, \tag{10}$$

$j = 1, \ldots, m$, followed by a draw of σ from the full conditional:

$$\sigma \mid \mathcal{T}_1, \ldots \mathcal{T}_m, \mathcal{M}_1, \ldots, \mathcal{M}_m, y. \tag{11}$$

The draw of σ in (11) is simply a draw from an inverse gamma distribution, which can be straightforwardly obtained by routine methods. More subtle is the implementation of the m draws of $(\mathcal{T}_j, \mathcal{M}_j)$ in (10). This can be done by taking advantage of the following simplifying reduction. First, observe that the conditional distribution $p(\mathcal{T}_j, \mathcal{M}_j \mid \mathcal{T}_{(j)}, \mathcal{M}_{(j)}, \sigma, y)$ depends on $(\mathcal{T}_{(j)}, \mathcal{M}_{(j)}, y)$ only through $\mathbf{R}_j = (r_{j1}, \ldots, r_{jn})'$, the $n \times 1$ vector of partial residuals

$$r_{ji} \equiv y_i - \sum_{k \neq j} g(\mathbf{x}_i; T_k, \mathcal{M}_k), \tag{12}$$

obtained from a fit that excludes the jth tree. Thus, the m draws of $(\mathcal{T}_j, \mathcal{M}_j)$ given $(\mathcal{T}_{(j)}, M_{(j)}, \sigma, y)$ in (10) are equivalent to m draws from

$$(\mathcal{T}_j, \mathcal{M}_j) \mid \sigma, \mathbf{R}_j, \tag{13}$$

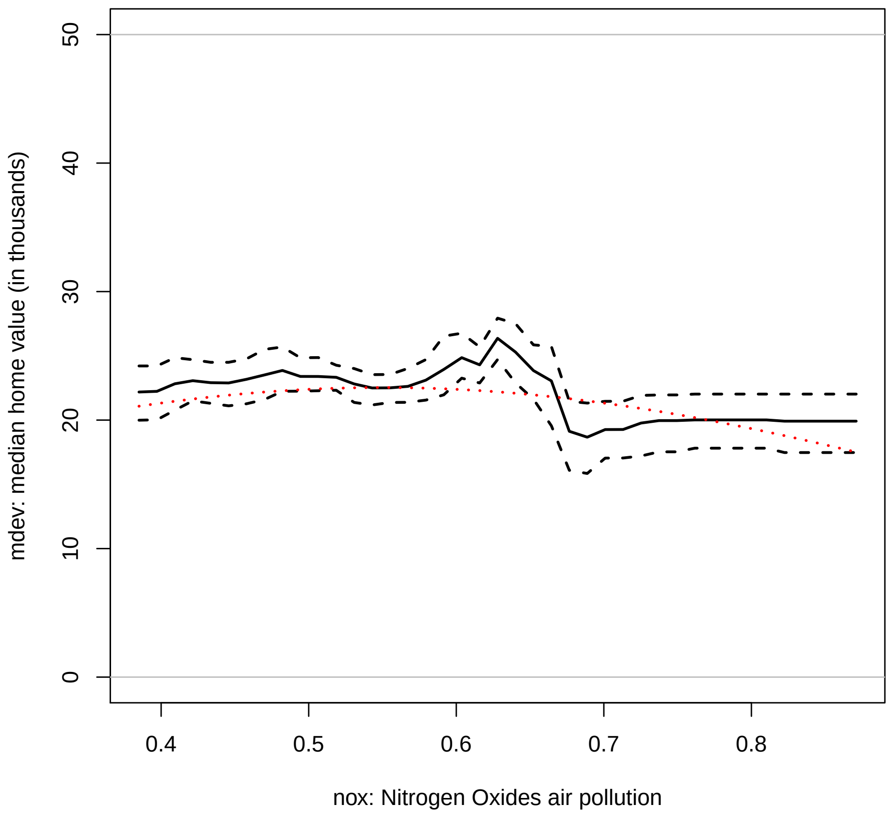

Figure 2: The Boston housing data was compiled from the 1970 US Census where each observation represents a Census tract in Boston with owner-occupied homes. For each tract, we have the median value of owner-occupied homes (in thousands of dollars), `mdev`, and thirteen other covariates including a localized air pollution estimate, the concentration of nitrogen oxides `nox`, which is our primary interest. We summarize the marginal effect of `nox` on `mdev` while aggregating over the other covariates with Friedman's partial dependence function. The marginal estimate and its 95% credible interval are shown. The red line with short dashes comes from the linear regression model of [12] where a quadratic effect of `nox` with respect to the logarithm of `mdev` is assumed.

$j = 1, \ldots, m$. Each of these draws is then done using the methods along the lines of those discussed in Section 3. We margin out $\mathcal{M}_j$ and then use MH proposals to modify $\mathcal{T}_j$. Given $\mathcal{T}_j$ we can draw $\mathcal{M}_j$.

The `R` package `BayesTree` uses all MH tree proposals in CGM98 and Section 3 for BART estimation. The `R` package `BART` just uses the BIRTH/DEATH step and redraws all the μ_{ij} at each MCMC iteration. This very simple approach works remarkably well in practice. The `R` package `rbart` implements BART (and a heteroskedastic version) using the more sophisticated tree moves of Section 3.4.

We initialize the chain with m simple single node trees, and then repeat iterations until satisfactory convergence is obtained. Fortunately, this backfitting MCMC algorithm appears to mix very well, as we have found that different restarts give remarkably similar results even in difficult problems. At each iteration, each tree may increase or decrease the number of terminal nodes by one, or change one or two splitting rules. The sum-of-trees model, with its abundance of unidentified parameters, allows the "fit" to glide freely from one tree to another. Because each move makes only small incremental changes to the fit, we can imagine the algorithm as analogous to sculpting a complex figure by adding and subtracting small dabs of clay.

For inference based on our MCMC sample, we rely on the fact our backfitting algorithm is *ergodic*. Thus, the induced sequence of sum-of-trees functions

$$f^*(\cdot) = \sum_{j=1}^{m} g(\cdot\,; \mathcal{T}_j^*, \mathcal{M}_j^*), \tag{14}$$

from the sequence of draws $(\mathcal{T}_1^*, \mathcal{M}_1^*), \ldots, (\mathcal{T}_m^*, \mathcal{M}_m^*)$, is converging to $p(f \mid y)$, the posterior distribution of the "true" $f(\cdot)$. Thus, by running the algorithm long enough after a suitable burn-in period, the sequence of f^* draws, say $f_1^*, \ldots, f_K^*$, may be regarded as an approximate, dependent sample of size K from $p(f \mid y)$. Bayesian inferential quantities of interest can then be approximated with this sample as follows.

To estimate $f(\mathbf{x})$ or predict Y at a particular $\mathbf{x}$, in-sample or out-of-sample, a natural choice is the average of the after burn-in sample $f_1^*, \ldots, f_K^*$,

$$\frac{1}{K} \sum_{k=1}^{K} f_k^*(\mathbf{x}), \tag{15}$$

which approximates the posterior mean $E(f(\mathbf{x}) \mid y)$. Posterior uncertainty about $f(\mathbf{x})$ may be gauged by the variation of $f_1^*(\mathbf{x}), \ldots, f_K^*(\mathbf{x})$. For example, a natural and convenient $(1-\alpha)\%$ posterior interval for $f(\mathbf{x})$ is obtained as the interval between the upper and lower $\alpha/2$ quantiles of $f_1^*(\mathbf{x}), \ldots, f_K^*(\mathbf{x})$.

7 BART Extentions

In this section we mention some BART extensions. The Bayesian formulation and corresponding MCMC approaches provide a rich environment for model and algorithm enhancement. We do not attempt to survey developments in Bayesian trees, but point to two very powerful examples of extending or modifying the BART approach. In Section 7.1, the BART prior is modified to enhance search for models that use a small number of predictors. In Section 7.2 the computational and modeling approach is extensively modified to enable a "BART like" inference for which is much faster and can handle much larger data sets.

7.1 The DART sparsity prior

Various Bayesian variable selection techniques applicable to BART have been studied [5, 7, 1, 11, 18, 16, 17]. Here we focus on the sparse variable selection prior of Linero [16] for which we use the acronym DART (where "D" stands for the Dirichlet distribution). Let's represent the variable selection probabilities by s_j where $j = 1, \dots, P$. Now, replace the uniform variable selection prior in BART with a Dirichlet prior as $[s_1, \dots, s_P] \mid \theta \overset{\text{prior}}{\sim} \mathrm{D}\left(\theta/P, \dots, \theta/P\right)$. The prior for θ is induced via $\theta/(\theta+\rho) \overset{\text{prior}}{\sim} \mathrm{Beta}\left(a,\ b\right)$. Typical settings are $b = 1$ and $\rho = P$. The distribution of θ controls the sparsity of the model: $a = 0.5$ induces a sparse posture while $a = 1$ is not sparse and similar to the uniform prior with probability $s_j = P^{-1}$. If additional sparsity is desired, then you can set ρ to a value smaller than P.

The key to understanding the inducement of sparsity is the distribution of the arguments to the Dirichlet prior: θ/P. It can be shown that $\theta/P \sim F(a, b, \rho/P)$ where $F(.)$ is the beta prime distribution scaled by ρ/P [15]. The non-sparse setting is $(a, b, \rho/P) = (1, 1, 1)$. As you can see in the Figure 3 [24], sparsity is promoted by reducing ρ, reducing a, or even further by reducing both.

Now, let's turn our attention to the posterior computation of the Dirichlet sparse prior. For a Dirichlet prior placed on the variable splitting probabilities, $\boldsymbol{s}$, its posterior samples are drawn via Gibbs sampling with conjugate Dirichlet draws. The Dirichlet parameter is updated by adding the total variable branch count over the ensemble, m_j, to the prior setting, $\frac{\theta}{P}$, i.e., $\left[\frac{\theta}{P} + m_1, \dots, \frac{\theta}{P} + m_P\right]$. In this way, the Dirichlet prior induces a "rich get richer" variable selection strategy. The sparsity parameter, θ, is drawn on a discrete grid of values [16]: this draw only depends on $[s_1, \dots, s_P]$.

7.1.1 Grouped variables and the DART prior

Here we take the opportunity to address a common pitfall of a Dirichlet prior for variable selection with a so-called grouped variable. Suppose that we have P variables, but Q of the covariates correspond to a grouped variable such as a series of dummy indicators encoded for a single categorical variable (suppose that these are the first Q variables without loss of generality): $x_1, \dots, x_Q$. N.B. Obviously, these developments apply to multiple grouped variables; however, for brevity, a single grouped variable will suffice to elucidate the problem and a solution. We denote the variable selection probabilities for all covariates as $\boldsymbol{s} = [s_1, \dots, s_P]$. There are two other probabilities of interest: the collapsed probabilities, $\boldsymbol{p} = [s_1 + \cdots + s_Q, s_{Q+1}, \dots, s_P]$ and the re-scaled probabilities $\boldsymbol{q} = [\tilde{s}_1, \dots, \tilde{s}_Q]$ where $\tilde{s}_j \propto s_j$

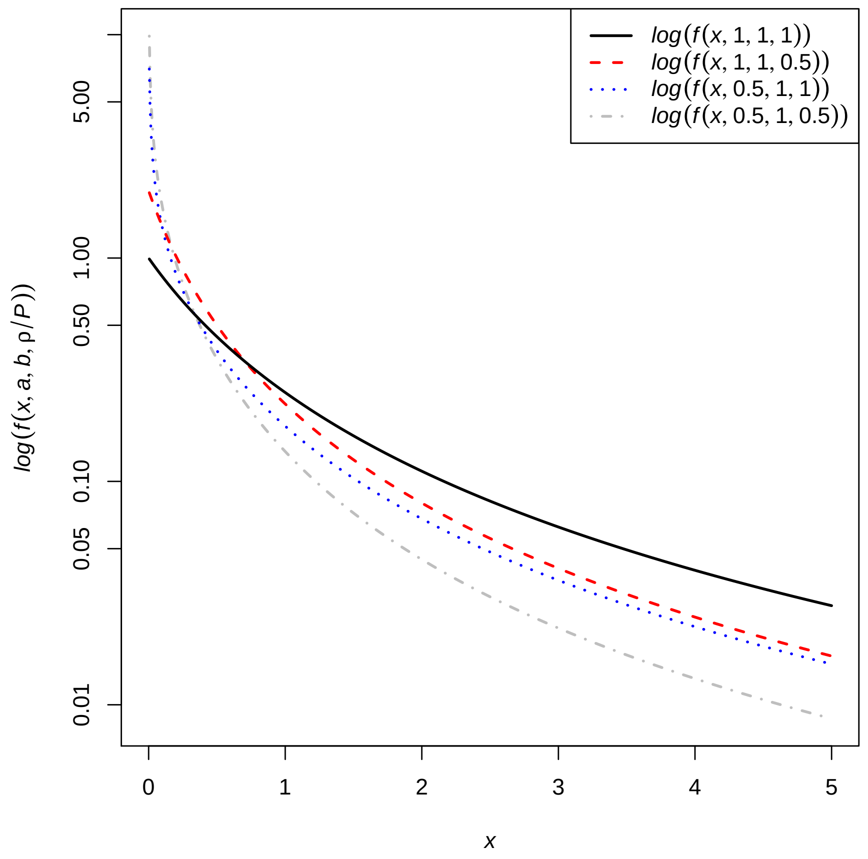

Figure 3: The distribution of θ/P and the sparse Dirichlet prior [24]. The key to understanding the inducement of sparsity is the distribution of the arguments to the Dirichlet prior: $\theta/P \sim F(a, b, \rho/P)$ where $F(.)$ is the beta prime distribution scaled by ρ/P. Here we plot the natural logarithm of the scaled beta prime density, $f(.)$, at a non-sparse setting and three sparse settings. The non-sparse setting is $(a, b, \rho/P) = (1, 1, 1)$ (solid black line). As you can see in the figure, sparsity is promoted by reducing ρ (long dashed red line), reducing a (short dashed blue line), or even further by reducing both (mixed dashed gray line).

such that $\sum_{j=1}^{Q} \tilde{s}_j = 1$. If we blindly use Dirichlet variable selection probabilities on data such as this, then we arrive at the following.

$$\begin{aligned}
\boldsymbol{s}|\theta &\overset{\text{prior}}{\sim} \mathrm{D}_P\left(\theta/P, \ldots, \theta/P\right) \\
&\quad \text{where the subscript } P \text{ is the order of the Dirichlet} \\
\boldsymbol{p}|\theta &\overset{\text{prior}}{\sim} \mathrm{D}_{\tilde{P}}\left(Q\theta/P, \theta/P, \ldots, \theta/P\right) \\
&\quad \text{where } \tilde{P} = P - Q + 1 \\
\boldsymbol{q}|\theta &\overset{\text{prior}}{\sim} \mathrm{D}_Q\left(\theta/P, \ldots, \theta/P\right)
\end{aligned}$$

The distribution of p_1, the first element of p, puts more prior weight on the grouped variable than the others. And now, the solution to the problem is trivial: re-scale $\boldsymbol{q}$ by Q^{-1} while naturally re-defining $\boldsymbol{p}$ and $\boldsymbol{s}$ as follows.

$$\begin{aligned}
\boldsymbol{p}|\theta &\overset{\text{prior}}{\sim} \mathrm{D}_{\tilde{P}}\left(\theta/\tilde{P}, \ldots, \theta/\tilde{P}\right) \\
\boldsymbol{q}|\theta &\overset{\text{prior}}{\sim} \mathrm{D}_Q\left(Q^{-1}\theta/\tilde{P}, \ldots, Q^{-1}\theta/\tilde{P}\right) \\
\boldsymbol{s}|\theta &\overset{\text{prior}}{\sim} \mathrm{D}_P\left(Q^{-1}\theta/\tilde{P}, \ldots, Q^{-1}\theta/\tilde{P}, \theta/\tilde{P}, \ldots, \theta/\tilde{P}\right) \\
&\overset{\text{prior}}{\sim} \mathrm{D}_P\left((\boldsymbol{q}|\theta), (\boldsymbol{p}|\theta)\right)
\end{aligned}$$

7.2 XBART

Markov chain algorithms based on independent local modifications to individual trees, or even just nodes of trees, are potentially slow to explore the immense space of binary trees. In some respects, it is remarkable that randomly selecting a variable to split on and a cut-point to split at work as well as it does! Greedy procedures based on recursive partitioning and exhaustive search, like those used in CART, may be able to more rapidly converge to local modes, especially when sample sizes are large and deep trees are required to approximate the response surface. However, optimization-based procedures produce a single output even when quite different trees fit the data essentially equally well. The XBART algorithm (for "Xcellerated", or "accelerated", BART) is a hybrid approach, which borrows elements of recursive partitioning by exhaustive search with elements of stochastic likelihood-weighted posterior sampling. The result is a stationary Markov chain that can be used to define its own estimator of the response surface, or draws from which can be used to initialize BART MCMC algorithms, reducing burn-in time. This section describes the XBART algorithm, with a special focus on the computational innovations this hybrid approach facilitates. For theoretical discussion and extensive simulation evidence, see [14] and [13].

7.2.1 The XBART algorithm and `GrowFromRoot`

At a high level, the XBART algorithm proceeds according to a series of iterative parameter updates, much like the original BART Gibbs sampler. Indeed, the sampling steps for σ and the leaf parameters μ are exactly the same as the full conditional updates from BART back-fitting. Likewise, XBART's tree updates are based on the residualized response, given the other trees in the *collection* and their parameters. Where XBART differs is that individual trees are re-grown anew at each update, rather than being modified incrementally. That is, rather than making a single transition to each tree, the current tree is deleted and regrown in full according to a recursive, but stochastic, growing process (individual branches stop growing stochastically). The main algorithm is presented in 2; the key subroutine, `GrowFromRoot`, is shown in 3. Although samples from this algorithm do not constitute draws from a bona fide Bayesian posterior, Monte Carlo averages may still be computed to define various estimators, specifically, predictions for new observations.

Algorithm 2 Accelerated Bayesian Additive Regression Trees (XBART)

```
procedure XBART($\mathbf{y}, \mathbf{X}, C, L$, num_samples)
output Samples of forest
    $p \leftarrow$ number of columns of $\mathbf{X}$
    $N \leftarrow$ number of rows of $\mathbf{X}$
    Initialize $r_l^{(0)} \leftarrow \mathbf{y}/L$.
    for $k$ in 1 to num_samples do
        for $l$ in 1 to $L$ do
            Calculate partial residual $r_l^{(k)}$ as shown in CGM10.
            if $k < I$ then
                GrowFromRoot($r_l^{(k)}$,$\mathbf{X}$)
            else
                GrowFromRoot($r_l^{(k)}$,$\mathbf{X}$)
            $\sigma^2 \sim$ Inverse-Gamma($N + \alpha, r_l^{(k)t} r_l^{(k)} + \eta$)
    return
```

The `GrowFromRoot` subroutine can be conceptualized as a sequence of draws from the posterior of "local Bayesian agents". At each node of the tree, the local Bayesian agent who "lives" at that node is given the data from the node above and updates her prior over a finite set of parameters, corresponding to partitions of the data. The likelihood used by these agents is the same as that from the BART model, but the local parameter set consists only of the available local partitions, irrespective of the previous or subsequent structure of the tree. Accordingly, the "local posterior" at each node is computed as a simple application of Bayes rule to a discrete parameter set. All available divisions are considered at each step, making the XBART algorithm comparatively fast at isolating partitions that are strongly indicated by the data. Formally, each local agent is tasked with partitioning the data into two parts (or leave it unpartitioned). Observations in the same partition are assumed to have the same, unknown, location parameter; therefore, the prior predictive distribution — obtained by integrating out the partition-specific mean — is a mean-zero multivariate

Algorithm 3 GrowFromRoot

```
procedure GrowFromRoot(r, $\mathbf{X}$)        ▷ Fit a tree to response vector r with predictors $\mathbf{X}$.
output A tree $T_l$.
    $N \leftarrow$ number of rows of $\mathrm{r}, \mathbf{X}$
    $p \leftarrow$ number of columms of $\mathrm{r}, \mathbf{X}$
    Evaluate expression 16 for $C$ evenly spaced cut-points for each of $p$ predictors.
    Sample a cut-point with probabilities given in expression 17.
    if sample no-split option then
        Sample leaf parameter according to $\mu \sim N\left(\sum \mathrm{r}/\left[\sigma^2\left(\frac{1}{\tau}+\frac{N}{\sigma^2}\right)\right], 1/\left[\frac{1}{\tau}+\frac{N}{\sigma^2}\right]\right)$. return
    else
        Partition data according to the selected cut-point.
        GrowFromRoot($y_{\text{left}}$,$\mathbf{X}_{\text{left}}$)
        GrowFromRoot($y_{\text{right}}$,$\mathbf{X}_{\text{right}}$)
```

normal distribution with covariance

$$\mathbf{V} = \tau \mathrm{J}\mathrm{J}^t + \sigma^2 \mathbf{I},$$

where τ is the prior variance of the leaf-specific mean parameter, σ^2 is the variance of the additive error, and J is a column vector of all ones. The prior predictive density of $\mathrm{y} \sim \mathcal{N}(0, \mathbf{V})$ is

$$p(\mathrm{y} \mid \tau, \sigma^2) = (2\pi)^{-n/2} \det(\mathbf{V})^{-1/2} \exp\left(-\frac{1}{2}\mathrm{y}^t \mathbf{V}^{-1} \mathrm{y}\right),$$

which can be simplified, using the matrix inversion lemma, to:

$$\mathbf{V}^{-1} = \sigma^{-2}\mathbf{I} - \frac{\tau}{\sigma^2(\sigma^2+\tau n)}\mathrm{J}\mathrm{J}^t.$$

Sylvester's determinant theorem applied to $\det \mathbf{V}^{-1}$ yields a log-predictive likelihood of

$$\begin{aligned}-\frac{n}{2}\log(2\pi) - n\log(\sigma) + \frac{1}{2}\log\left(\frac{\sigma^2}{\sigma^2+\tau n}\right)\\ - \frac{1}{2}\frac{\mathrm{y}^t\mathrm{y}}{\sigma^2} + \frac{1}{2}\frac{\tau}{\sigma^2(\sigma^2+\tau n)}s^2,\end{aligned}$$

where $s \equiv \mathrm{y}^t\mathrm{J} = \sum_i y_i$ so that $\mathrm{y}^t\mathrm{J}\mathrm{J}^t\mathrm{y} = (\sum_i y_i)^2 = s^2$. Considering both partitions, $b \in \{\text{left}, \text{right}\}$, gives a combined log-predictive likelihood of

$$\begin{aligned}&\sum_b \left\{-\frac{n_b}{2}\log(2\pi) - n_b\log(\sigma) + \frac{1}{2}\log\left(\frac{\sigma^2}{\sigma^2+\tau n_b}\right)\right.\\ &\qquad \left. - \frac{1}{2}\frac{\mathrm{y}_b^t\mathrm{y}_b}{\sigma^2} + \frac{1}{2}\frac{\tau}{\sigma^2(\sigma^2+\tau n_b)}s_b^2\right\}\\ &= -n\log(2\pi) - n\log(\sigma) - \frac{1}{2}\frac{\mathrm{y}^t\mathrm{y}}{\sigma^2}\\ &\qquad + \frac{1}{2}\sum_b\left\{\log\left(\frac{\sigma^2}{\sigma^2+\tau n_b}\right) + \frac{\tau}{\sigma^2(\sigma^2+\tau n_b)}s_b^2\right\}.\end{aligned}$$

The first three terms are not functions of the partition yielding a "local likelihood" proportional to

$$\sum_b \left\{ \log\left(\frac{\sigma^2}{\sigma^2+\tau n_b}\right) + \frac{\tau}{\sigma^2(\sigma^2+\tau n_b)} s_b^2 \right\}, \tag{16}$$

where n_b and s_b are functions of the partition (which is defined by the cut-point). These formulae have been written in terms of data y to emphasize the "local" interpretation/justification of the model. In the implementation, however, the data are the partial residuals.

Selection of a variable to split on, and a cut-point to split at, are then sample according to Bayes rule:

$$\pi(v,c) = \frac{\exp\left(\ell(c,v)\right)\kappa(c)}{\sum_{v'=1}^{p}\sum_{c'=0}^{C}\exp\left(\ell(c',v')\right)\kappa(c')} \tag{17}$$

where

$$\begin{aligned}\ell(v,c) = &\frac{1}{2}\left\{\log\left(\frac{\sigma^2}{\sigma^2+\tau n(\leq,v,c)}\right)\right.\\ &\qquad \left.+ \frac{\tau}{\sigma^2(\sigma^2+\tau n(\leq,v,c))}s(\leq,v,c)^2\right\}\\ &+\frac{1}{2}\left\{\log\left(\frac{\sigma^2}{\sigma^2+\tau n(>,v,c)}\right)\right.\\ &\qquad \left.+ \frac{\tau}{\sigma^2(\sigma^2+\tau n(>,v,c))}s(>,v,c)^2\right\}\end{aligned}$$

for $c \neq 0$. The partition size is denoted $n(\leq,v,c)$, which is the number of observations such that $x_v \leq c$; similarly, $s(\leq,v,c)$ is the sum of the residual $r_l^{(k)}$ of those same observations. The complement quantities, $n(>,v,c)$ and $s(>,v,c)$, are defined analogously. A uniform prior is applied to the cut-points so that, $\kappa(c \neq 0) = 1$.

Stochastic termination of the growing process is achieved by including a "no split" option in the local agents' parameter sets, effectively corresponding to a cut location that lies outside of the range of the data. The prior on this parameter can be chosen such that the XBART prior predictive (the algorithm applied to no data) corresponds to the usual BART prior predictive. Formally, for $c = 0$, which corresponds to no split,

$$\ell(v,c) = \frac{1}{2}\left\{\log\left(\frac{\sigma^2}{\sigma^2+\tau n}\right) + \frac{\tau}{\sigma^2(\sigma^2+\tau n)}s^2\right\}$$

and $\kappa(0) = \frac{1-\alpha(1+d)^{-\beta}}{\alpha(1+d)^{-\beta}}$. With this weight, the probability of *splitting* is the complement set of not splitting:

$$p_{SPLIT} = 1 - \frac{|\mathcal{C}|(\alpha^{-1}(1+d)^{\beta}-1)}{|\mathcal{C}|(\alpha^{-1}(1+d)^{\beta}-1)+|\mathcal{C}|} = \alpha(1+d)^{-\beta},$$

just as in the original BART prior.

Relative to BART MCMC samplers, XBART has higher per-iteration cost because it must evaluate the likelihood at $|\mathcal{C}|$ points at each node during `GrowFromRoot`. The benefits of this higher cost are (usually) improved posterior exploration leading to dramatically fewer required iterations. Still, any improvement to the per iteration computational burden are beneficial and the recursive structure of XBART permits a number of helpful improvements.

Two particular innovations deserve to be highlighted: pre-sorting the predictor variables and using cut point values based on local quantiles (as opposed to using all valid cut points at each node).

Pre-sorting predictor variables Because the BART marginal likelihood depends only on partition sums, the sufficient statistics for *all* cut points at a given node can be calculated with a single pass through the data at each variable by computing a cumulative sum, provided that the response values (in the form of the partial residual) are accessible in sorted order (for each predictor). More formally, define the cumulative sums in terms of a matrix of indices, $\mathbf{O}$, with elements o_{vh} denoting the index of the h^{th} largest observation of the x_v^{th} variable in the original data matrix. In terms of $\mathbf{O}$, the partition sums can be expressed as

$$s(\leq, v, c) = \sum_{h \leq c} r_{o_{vh}} \tag{18}$$

and

$$s(>, v, c) = \sum_{h=1}^{n} r_{lh} - s(\leq, v, c). \tag{19}$$

where r denotes the vector of partial residuals from the other trees. These sums are the inputs to the `GrowFromRoot` split criterion. To perform a similar operation at the subsequent node, the variable sorting must be maintained; fortunately this can be achieved efficiently by "sifting" the variables. After a variable v and cut point c are drawn, the algorithm partitions $\mathbf{O}$ into two matrices $\mathbf{O}^{\leq}$ and $\mathbf{O}^{>}$ which are populated sequentially by evaluating each element of $\mathbf{O}$ in turn and sending it to the next element of either $\mathbf{O}^{\leq}$ and $\mathbf{O}^{>}$, according to whether the corresponding element has $x_j \leq c$ or not. By populating each row of $\mathbf{O}^{\leq}$ and $\mathbf{O}^{>}$ by sequentially scanning the rows of $\mathbf{O}$, the ordering is preserved for the next step of the recursion.

Adaptive nested cut points The discrete Bayes rule calculation at the heart of the stochastic `GrowFromRoot` procedure is computationally intensive when sample sizes are large (especially at early stages of the growing process, such as the split at the root), because each data point defines a valid cutting location. In some measure, this is why the BART MCMC implementations favor a pre-defined grid of candidate cut locations (perhaps based on marginal quantiles or a uniform grid). The recursive nature of the `GrowFromRoot` algorithm permits an "adaptive" alternative, where a non-exhaustive set of quantiles can be considered at each node, where the quantiles are computed relative to the available data at the present node. Conveniently, these quantiles need never be computed explicitly; instead, one simply evaluates the likelihood at "strides" by skipping a fixed number of observations (in sorted order) when calculating the marginal likelihood split criterion. All of the cumulative sums must still be computed, but the sampling is performed among a much smaller subset of cut points, saving significant computational effort on both the likelihood evaluations as well as the random variable generation. This approach does not reduce the expressivity of the model, as any cut point can eventually be selected, just perhaps further down the tree.

Thus there is a trade-off between coarser/sparser cut point candidates and eventual tree depth. In practice, using tens or hundreds of cut points (rather than thousands or more) seems to gives good performance. Intuitively, the adaptive cut point strategy will work well when there are large regions of covariate space where the function is relatively flat and others where it is comparatively variable. Coarser cut point sets permit rapid identification of the flat regions, while simultaneously growing deeper trees in regions of high response surface variation. A function which oscillates rapidly uniformly over the predictor space may be more efficiently fit with a denser set of candidate cut points.

7.2.2 Warm-start XBART

An especially appealing aspect of the XBART algorithm is its use in conjunction with traditional MCMC BART, by initializing independent Markov chains at draws from XBART. This approach combines XBART's ability to rapidly find potentially large trees that fit the data well with the valid posterior uncertainty assessment that MCMC provides. Provided that each draw from XBART is from a starting location in a high probability region of the BART posterior, burn-in times are negligible for each chain, leading to substantially lower run times. Meanwhile, the diversity of the various starting locations results in wider credible intervals for quantities of interest, such as point-wise predictions. Nearness of the tree draws (according to various metrics) from the separate chains may also be used as a gauge of mixing, although in practice simply appending the separate draws appears to yield conservatively wide intervals, which has its own appeal. Simulation results indicate that warm-start XBART is faster and has better point-wise coverage of the mean function compared to either XBART or MCMC BART [13].

8 Conclusion

Bayesian tree modeling is a rich area of ongoing research, with challenges ranging from fundamental modeling to the construction of computational algorithms. The Bayesian approach offers many advantages, for example, BART infers the depth of each tree rather than having to tune it using cross-validation as in most non-Bayesian boosting approaches. But there is cost to the Bayesian advantages. Not everyone wants to choose a prior, and not everyone wants MCMC draws.

As empirical analysis continues to take center stage today, we see a growing variety of applications in data science with many different kinds of objectives. We believe that the fundamentals of Bayesian thinking will continue to play in role in the development of methodology that is relevant to real world decision making, and Bayesian tree models will continue to be a useful part of that bigger picture.

References

[1] Justin Bleich, Adam Kapelner, Edward I George, and Shane T Jensen. Variable selection for BART: An application to gene regulation. *The Annals of Applied Statistics*, 8(3):1750–1781, 2014.

[2] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001.

[3] Leo Brieman, Jerome Friedman, Richard Olshen, and Charles Stone. *Classification and Regression Trees*. Chapman & Hall, 1993.

[4] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[5] H.A. Chipman, E.I. George, and R.E. McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

[6] Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

[7] Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian regression structure discovery. *Bayesian Theory and Applications,(Eds, P. Damien, P. Dellaportas, N. Polson, D. Stephens), Oxford University Press, Oxford, UK*, 2013.

[8] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[9] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[10] P. Richard Hahn, Jared S. Murray, and Carlos M. Carvalho. Bayesian Regression Tree Models for Causal Inference: Regularization, Confounding, and Heterogeneous Effects (with Discussion). *Bayesian Analysis*, 15(3):965 – 1056, 2020.

[11] PR Hahn and CM Carvalho. Decoupling shrinkage and selection in Bayesian linear models: a posterior summary perspective. *Journal of the American Statistical Association*, 110(509):435–448, 2015.

[12] David Harrison Jr and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978.

[13] Jingyu He and P Richard Hahn. Stochastic tree ensembles for regularized nonlinear regression. *Journal of the American Statistical Association*, (ahead of print):1–61, 2021.

[14] Jingyu He, Saar Yalov, and P Richard Hahn. XBART: Accelerated Bayesian additive regression trees. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1130–1138, 2019.

[15] Norman Lloyd Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Continuous Univariate Distributions*, volume 2. John Wiley & Sons, New York, 2nd edition, 1995.

[16] A. Linero. Bayesian regression trees for high dimensional prediction and variable selection. *Journal of the American Statistical Association*, 113(522):626–36, 2018.

[17] Yi Liu and Veronika Ročková. Variable selection via Thompson sampling. *Journal of the American Statistical Association*, (ahead of print):1–41, 2021.

[18] RE McCulloch, C Carvalho, and R Hahn. A general approach to variable selection using Bayesian nonparametric models, 2015. Joint Statistical Meetings, Seattle, 08/09/15-08/13/15.

[19] Robert E. McCulloch, Rodney A. Sparapani, Brent R. Logan, and Purushottam W. Laud. Causal inference with the instrumental variable approach and Bayesian nonparametric machine learning. *arXiv preprint*, 2102.01199, 2021.

[20] Reza Mohammadi, Matthew Pratola, and Maurits Kaptein. Continuous-time birth-death MCMC for Bayesian regression tree models. *Journal of Machine Learning Research*, 21(201):1–26, 2020.

[21] M. T. Pratola. Efficient metropolis-hastings proposal mechanisms for Bayesian regression tree models. *Bayesian Analysis*, 11:885–911, 2016.

[22] Veronika Ročková and Enakshi Saha. On theory for BART. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89, pages 2839–2848. PMLR, 16–18 Apr 2019.

[23] Veronika Ročková and Stephanie van der Pas. Posterior concentration for Bayesian regression trees and forests. *Annals of Statistics*, 48(4):2108–2131, 08 2020.

[24] Rodney Sparapani, Charles Spanbauer, and Robert McCulloch. Nonparametric machine learning and efficient computation with Bayesian additive regression trees: The BART R package. *Journal of Statistical Software*, 97(1):1–66, 2021.