BART:
Bayesian Additive Regression Trees

Hugh A. Chipman, Edward I. George, Robert E. McCulloch [*]

July 2005, Revision June 2006

Abstract

We develop a Bayesian "sum-of-trees" model where each tree is constrained by a regularization prior to be a weak learner, and fitting and inference are accomplished via an iterative Bayesian backfitting MCMC algorithm that generates samples from a posterior. Effectively, BART is a nonparametric Bayesian regression approach which uses dynamic random basis elements that are dimensionally adaptive. BART is motivated by ensemble methods in general, and boosting algorithms in particular. However, BART is defined by a statistical model: a prior and a likelihood, while boosting is defined by an algorithm. This model-based approach enables a full assessment of prediction uncertainty while remaining highly competitive in terms of prediction accuracy. The potential of BART is illustrated on examples where it compares favorably with competing methods including gradient boosting, neural nets and random forests. It is also seen that BART is remarkably effective at finding low dimensional structure in high dimensional data.

KEY WORDS: Bayesian backfitting; Boosting; CART; MCMC; Random basis; Regularization; Sum-of-trees model; Weak learner.

[*] Hugh Chipman is Associate Professor and Canada Research Chair in Mathematical Modelling, Department of Mathematics and Statistics, Acadia University, Wolfville, Nova Scotia, B4P 2R6. Edward I. George is Professor of Statistics, The Wharton School, University of Pennsylvania, 3730 Walnut St, 400 JMHH, Philadelphia, PA 19104-6304, edgeorge@wharton.upenn.edu. Robert E. McCulloch is Professor of Statistics, Graduate School of Business, University of Chicago, 5807 S. Woodlawn Ave, Chicago, IL 60637, Robert.McCulloch@gsb.uchicago.edu. This work was supported by NSF grant DMS-0130819.

1 Introduction

We consider the fundamental problem of making inference about an unknown function f that predicts an output Y using a p dimensional vector of inputs x when

$$Y = f(x) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2). \tag{1}$$

To do this, we consider modelling or at least approximating $f(x) = E(Y \mid x)$, the mean of Y given x, by a sum of m regression trees $f(x) \approx g_1(x) + g_2(x) + \ldots + g_m(x)$ where each g_i denotes a binary regression tree. Replacing f in (1) by this approximation, we obtain

$$Y = g_1(x) + g_2(x) + \cdots + g_m(x) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2). \tag{2}$$

The sum-of-trees model (2) is fundamentally an additive model with multivariate components. It is vastly more flexible than a single tree model which does not easily incorporate additive effects. For example, Hastie, Tibshirani & Friedman (2001) (p.301) illustrate how a sum-of-trees model can realize half the prediction error rate of a single tree model when the true underlying f is a thresholded sum of quadratic terms. And because multivariate components can easily account for high order interaction effects, a sum-of-trees model is also much more flexible than typical additive models that use low dimensional smoothers as components.

Various methods which combine a set of tree models, so called ensemble methods, have attracted much attention for prediction. These include boosting (Freund & Schapire (1997), Friedman (2001)), random forests (Breiman 2001) and bagging (Breiman 1996), each of which use different techniques to fit a linear combination of trees. Boosting fits a sequence of single trees, using each tree to fit data variation not explained by earlier trees in the sequence. Bagging and random forests use data randomization and stochastic search to create a large number of independent trees, and then reduce prediction variance by averaging predictions across the trees. Yet another approach that results in a linear combination of trees is Bayesian model averaging applied to the posterior arising from a Bayesian single-tree model as in Chipman, George & McCulloch (1998) (hereafter CGM98), Denison, Mallick & Smith (1998) and Wu, Tjelmeland & West (2005). Such model averaging uses posterior probabilities as weights for averaging the predictions from individual trees.

In this paper we propose a Bayesian approach called BART (Bayesian Additive Regression Trees) which uses a sum-of-trees model to estimate and draw inference about $f(x) = E(Y\,|x)$. The essential idea is to enhance the sum-of-trees model (2) by treating each of the tree components g_i as random. We do this by imposing a regularization prior which keeps the individual tree effects small. In effect, the g_i's become a dimensionally adaptive random basis of "weak learners", to borrow a phrase from the boosting literature. In sharp contrast to previous Bayesian single-tree methods, BART treats the sum of trees as the model itself, rather than let it arise out of model-averaging over a set of single-tree models. As opposed to ending up with a weighted sum of separate single tree attempts to fit the entire function f, BART ends up with a sum of trees, each of which explains a small and different portion of f.

To fit the sum-of-trees model, BART uses a tailored version of Bayesian backfitting MCMC (Hastie & Tibshirani (2000)) that iteratively constructs and fits successive residuals. Although similar in spirit to the gradient boosting approach of Friedman (2001), BART differs in both how it weakens the individual trees by instead using a prior, and how it performs the iterative fitting by instead using Bayesian backfitting on a fixed number of trees. Conceptually, BART can be viewed as a Bayesian nonparametric approach that fits a parameter rich model using a strongly influential prior distribution.

Another novel feature of BART is that it produces an MCMC sample from the induced posterior over the sum-of-trees model space, a sample that can be readily used for enhanced inference. For example, a single posterior mean estimate of $f(x) = E(Y|x)$ at any input value x is simply obtained by averaging the successive MCMC draws of f evaluated at x. Moreover, pointwise uncertainty intervals for $f(x)$ are easily obtained by the corresponding quantiles. As will be seen, these uncertainty intervals behave sensibly, for example by widening for predictions at test points far from the training set.

The remainder of the paper is organized as follows. In Section 2, the BART model is outlined. This consists of the sum-of-trees model combined with a regularization prior. In Section 3, the Bayesian backfitting MCMC algorithm for fitting and inference is described. In Section 4, examples, both simulated and real, are used to demonstrate the potential of BART. Section 5 concludes with a discussion.

2 The BART Model

2.1 A Sum-of-Trees Model

To elaborate the form of a sum-of-trees model, we begin by establishing notation for a single tree model. Let T denote a binary tree consisting of a set of interior node decision rules and a set of terminal nodes, and let $M = \{\mu_1, \mu_2, \ldots, \mu_b\}$ denote a set of parameter values associated with each of the b terminal nodes of T. The decision rules are binary splits of the predictor space of the form $\{x \in A\}$ vs $\{x \notin A\}$ where A is a subset of the range of x. Each x value is associated with a single terminal node of T by the sequence of decision rules from top to bottom, and is then assigned the μ_i value associated with this terminal node. For a given T and M, we use $g(x; T, M)$ to denote the function which assigns a $\mu_i \in M$ to x. Thus,

$$Y = g(x; T, M) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2) \tag{3}$$

is a single tree model of the form considered by CGM98. Note that here, the terminal node mean given by $g(x; T, M)$ is simply $E(Y \mid x)$, the conditional mean of Y given x.

Using this notation, the sum-of-trees model (2) can more explicitly be expressed as

$$Y = g(x; T_1, M_1) + g(x; T_2, M_2) + \cdots + g(x; T_m, M_m) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2). \quad (4)$$

Unlike the single tree model (3), when $m > 1$ the terminal node parameter μ_i given by $g(x; T_j, M_j)$ is merely part of the conditional mean of Y given x. Such terminal node parameters will represent interaction effects when their assignment depends on more than one component of x (i.e., more than one variable). And because (4) may be based on trees of varying sizes, the sum-of-trees model can incorporate both main effects and interaction effects of varying orders. In the special case where every terminal node assignment depends on just a single component of x, the sum-of-trees model reduces to a simple additive function of the individual components of x.

With a large number of trees, a sum-of-trees model gains increased representation flexibility which, as we'll see, endows BART with excellent predictive capabilities. Indeed, in the examples in Section 4, we set m as large as 200. However, this representational flexibility is obtained at the cost of a rapidly increasing number of parameters. Indeed, for fixed m, each sum-of-trees model (4) is determined by $(T_1, M_1), \ldots, (T_m, M_m)$ and σ, which includes all the bottom node parameters as well as the tree structures and decision rules. Further, the representational flexibility of each individual tree leads to substantial redundancy across the tree components. Indeed, one can regard $\{g(x; T_1, M_1), \ldots, g(x; T_m, M_m)\}$ as an "overcomplete basis" for the vector space of possible $E(Y \mid x)$ values.

The downside of the rich structure of (4) is that single large trees can overwhelm the model, thereby limiting the advantages of the additive representation both in terms of function approximation and computation. To guard against this possibility, we complete the BART model specification by introducing a strong regularization prior over $(T_1, M_1), \ldots, (T_m, M_m)$ and σ that keeps each of the individual tress from being unduly influential, thereby restraining the overall fit.

2.2 A Regularization Prior

The BART model consists of two parts: a sum-of-trees model (4) and a regularization prior on $((T_1, M_1), \ldots, (T_m, M_m), \sigma)$ that we now proceed to describe. (As discussed in Section 2.2.4 below, we treat m as a fixed constant). The complexity of such prior specification is vastly simplified by restricting attention to priors for which

$$\begin{aligned} p((T_1, M_1), \ldots, (T_m, M_m), \sigma) &= \left[\prod_j p(T_j, M_j)\right] p(\sigma) \\ &= \left[\prod_j p(M_j \mid T_j)\, p(T_j)\right] p(\sigma) \end{aligned} \tag{5}$$

and

$$p(M_j \mid T_j) = \prod_i p(\mu_{i,j} \mid T_j), \tag{6}$$

where $\mu_{i,j}$ is the ith component of M_j. Under such priors, the tree components are independent of each other and of σ, and the bottom node parameters of every tree are independent.

The independence restrictions above simplify the prior choice problem to the specification of prior forms for just $p(T_j), p(\mu_{i,j} \mid T_j)$ and $p(\sigma)$. As described in the ensuing subsections, we use the same tree prior form proposed by CGM98 for all $p(T_j)$, we use the same normal conjugate form for all $p(\mu_{i,j} \mid T_j)$ and we use the inverse gamma conjugate form for $p(\sigma)$. In addition to their valuable computational benefits, these forms are controlled by just a few interpretable hyperparameters which can be calibrated to yield effective default specifications.

2.2.1 The Tree Prior

For $p(T_j)$, we follow CGM98 which is easy to specify and dovetails nicely with calculations for the backfitting MCMC algorithm described in Section 3. This prior form is specified by three aspects: (i) the probability that a node at depth d is nonterminal, given by

$$\alpha(1+d)^{-\beta}, \qquad \alpha \in (0,1), \beta \in [0,\infty), \tag{7}$$

(ii) the distribution on the splitting variable assignments at each interior node, and (iii) the distribution on the splitting rule assignment in each interior node, conditional on splitting variable. For (ii) and (iii) we use the simple defaults used by CGM98, namely the uniform prior on available variables for (ii) and the uniform prior on the discrete set of available splitting values for (iii).

In a single tree model, (i.e. $m = 1$), a tree with many terminal nodes may be needed to model complicated structure. However, for a sum-of-trees model, especially with m large, we want the regularization prior to keep the individual tree components small. In our examples in Section 4, we do so by using $\alpha = .95$ and $\beta = 2$ in (7). With this choice, trees with 1, 2, 3, 4, and ≥ 5 terminal nodes receive prior probability of 0.05, 0.55, 0.28, 0.09, and 0.03, respectively. Note that even with this prior, trees with many terminal nodes can be grown if the data demands it. For example, in one of our simulated examples with this prior, we observed considerable posterior probability on trees of size 17 when we set $m = 1$.

2.2.2 The $\mu_{i,j}$ Prior

In the sum-of-trees model (4), $E(Y \mid x)$ is the sum of m $\mu_{i,j}$'s. Given our independence assumptions (5) and (6), we need only specify the prior for a single scalar $\mu_{i,j}$. As mentioned previously, we prefer a conjugate prior, here the normal distribution $N(\mu_\mu, \sigma_\mu^2)$. The essence of our strategy is then to choose the prior mean μ_μ and standard deviation σ_μ so that a sum of m independent realizations gives a reasonable range for the conditional mean of Y.

For convenience we start by simply shifting and rescaling Y so that we believe the prior probability that $E(Y \mid x) \in (-.5, .5)$ is very high. Unless reliable information about the range of Y is available, we usually do this by using the observed y values, shifting and rescaling so that the observed y values range from -.5 to .5. Such informal empirical Bayes methods can be very useful to ensure that prior specifications are at least in the right ballpark.

For the transformed Y, we then center the prior at zero $\mu_\mu = 0$ and choose the standard deviation σ_μ so that the mean of Y falls in the interval $(-.5, .5)$ with "high" probability. Now the standard deviation of the sum of m independent $\mu_{i,j}$'s is $\sqrt{m}\,\sigma_\mu$. Thus, we choose σ_μ so that $k\sqrt{m}\sigma_\mu = .5$ for a suitable value of k. For example, $k = 2$ would yield a 95% prior probability that the expected value of Y is in the interval $(-.5, .5)$. In summary, our prior for each $\mu_{i,j}$ is simply

$$\mu_{i,j} \sim N(0, \sigma_\mu^2) \text{ where } \sigma_\mu = .5/k\sqrt{m}. \tag{8}$$

This prior has the effect of shrinking the tree parameters $\mu_{i,j}$ towards zero, limiting the effect of the indiviual tree components of (4) by keeping them small. Note that as k and/or the number of trees m is increased, this prior will become tighter and apply greater shrinkage to the $\mu_{i,j}$'s. Prior shrinkage on $\mu_{i,j}$'s is the counterpart of the shrinkage parameter in Friedman's (2001) gradient boosting algorithm. The prior variance σ_μ of $\mu_{i,j}$ here and the gradient boosting shrinkage parameter there, both serve to "weaken" the individual trees so that each is constrained to play a smaller role in the overall fit. For the choice of k, we have found that values of k between 1 and 3 yield good results, and we recommend $k = 2$ as an automatic default choice. Alternatively the value of k may be chosen by cross-validation.

Although the calibration of this prior is based on a simple transformation of Y, it should be noted that there is no need to transform the predictor variables.

This is a consequence of the fact that the tree splitting rules are invariant to monotone transformations of the x components. The utter simplicity of using our prior for $\mu_{i,j}$ is an appealing feature of BART. In contrast, methods like neural nets that use linear combinations of predictors require standardization choices for each predictor.

2.2.3 The σ Prior

We again use a conjugate prior, here the inverse chi-square distribution $\sigma^2 \sim \nu\,\lambda/\chi^2_\nu$. For the hyperparameter choice of ν and λ, we proceed as follows. We begin by obtaining a "rough overestimate" $\hat{\sigma}$ of σ as described below. We then pick a degrees of freedom value ν between 3 and 10. Finally, we pick a value of q such as 0.75, 0.90 or 0.99, and set λ so that the qth quantile of the prior on σ is located at $\hat{\sigma}$, that is $P(\sigma < \hat{\sigma}) = q$.

Figure 1 illustrates priors corresponding to three (ν, q) settings when the rough overestimate is $\hat{\sigma} = 2$. We refer to these three settings, $(\nu, q) = (10, 0.75)$, $(3, 0.90)$, $(3, 0.99)$, as conservative, default and aggressive, respectively. The prior mode moves towards smaller σ values as q is increased. We recommend against choosing $\nu < 3$ because it seems to concentrate too much mass on very small σ values, which leads to overfitting. In our examples, we have found these three settings work very well and yield similar results. For automatic use, we recommend the default setting $(\nu, q) = (3, 0.90)$ which tends to avoid extremes.

The key to an effective specification above is to come up with a reasonable value of $\hat{\sigma}$. In the absence of real prior information, this can be obtained by either of the following informal empirical Bayes approaches: 1) the "naive" specification, in which we take $\hat{\sigma}$ to be the sample standard deviation of Y, or 2) the "linear model" specification, in which we take $\hat{\sigma}$ as the residual standard deviation from a least squares linear regression of Y on the original X's. The naive specification represents the belief that BART can provide a much better estimate of $E(Y \mid x)$

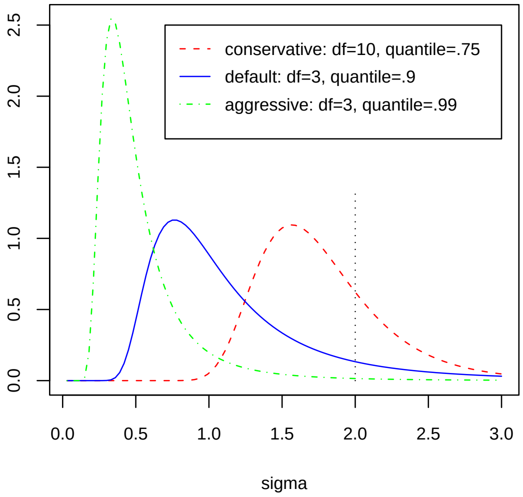

Figure 1: Three priors on σ when $\hat{\sigma} = 2$.

values than the sample mean of Y. The linear model specification represents the belief that BART can fit better than a linear model.

2.2.4 Choosing the Number of Trees m

Our procedure fixes m, the number of trees. Although it appears that using too many trees only slightly degrades predictive performance, it is still useful to have some intuition about reasonable values of m. We have found it helpful to first assess prior beliefs about how many variables are likely to be important. One might then assume that five trees are needed for each important variable. For example, if we believe 10 out of 200 variables are important, we might try $m = 50$ trees. Using more than 50 trees may slow down computation with little benefit, but if there is complicated structure it may help the fit. In applications we have typically used a large number of trees, ($m = 100$ and $m = 200$ in our examples), as we have found that predictive performance suffers more when too few trees are selected rather than too many. Although we find it easier and faster to simply consider and compare our results for various choices of m, it would be straightforward to consider a fully Bayes approach that puts a prior on m.

3 A Bayesian Backfitting MCMC Algorithm

Given the observed data y, our Bayesian setup induces a posterior distribution

$$p((T_1, M_1), \ldots, (T_m, M_m), \sigma |\, y) \tag{9}$$

on all the unknowns that determine a sum-of-trees model (4). Although the sheer size of this parameter space precludes exhaustive calculation, the following backfitting MCMC algorithm can be used to sample from this posterior.

At a general level, our algorithm is a Gibbs sampler. For notational convenience, let $T_{(j)}$ be the set of all trees in the sum *except* T_j, and similarly define $M_{(j)}$. Thus $T_{(j)}$ will be a set of $m-1$ trees, and $M_{(j)}$ the associated terminal node parameters. The Gibbs sampler here entails m successive draws of (T_j, M_j) conditionally on $(T_{(j)}, M_{(j)}, \sigma)$:

$$\begin{gathered}(T_1, M_1)|T_{(1)}, M_{(1)}, \sigma, y \\ (T_2, M_2)|T_{(2)}, M_{(2)}, \sigma, y \\ \vdots \\ (T_m, M_m)|T_{(m)}, M_{(m)}, \sigma, y,\end{gathered} \tag{10}$$

followed by a draw of σ from the full conditional:

$$\sigma|T_1, \ldots T_m, M_1, \ldots, M_m, y. \tag{11}$$

Hastie & Tibshirani (2000) considered a similar application of the Gibbs sampler for posterior sampling for additive and generalized additive models with σ fixed, and showed how it was a stochastic generalization of the backfitting algorithm for such models. For this reason, we refer to our algorithm as backfitting MCMC.

The draw of σ in (11) is simply a draw from an inverse gamma distribution and so can be easily obtained by routine methods. Much more challenging is how to implement the m draws of (T_j, M_j) in (10). This can be done by taking advantage of the following reductions. First, observe that the conditional distribution $p(T_j, M_j|\, T_{(j)}, M_{(j)}, \sigma, y)$ depends on $(T_{(j)}, M_{(j)}, y)$ only through

$$R_j \equiv y - \sum_{k \neq j} g(x; T_k, M_k), \tag{12}$$

the $n-$vector of partial residuals based on a fit that excludes the jth tree. Thus, a draw of (T_j, M_j) given $(T_{(j)}, M_{(j)}, \sigma, y)$ in (10) is equivalent to a draw from

$$(T_j, M_j)|R_j, \sigma. \tag{13}$$

Now (13) is formally equivalent to the posterior of the single tree model $R_j = g(x; T_j, M_j) + \epsilon$ where R_j plays the role of the data y. Because we have used a conjugate prior for M_j,

$$p(T_j|R_j, \sigma) \propto p(T_j) \int p(R_j|M_j, T_j, \sigma) p(M_j|T_j, \sigma) dM_j \tag{14}$$

can be obtained in closed form up to a norming constant. This allows us to carry out the draw from (13), or equivalently (10), in two successive steps as

$$T_j|R_j, \sigma \tag{15}$$

$$M_j|T_j, R_j, \sigma. \tag{16}$$

The draw of T_j in (15), although somewhat elaborate, can be obtained using the Metropolis-Hastings (MH) algorithm of CGM98. This algorithm proposes a new tree based on the current tree using one of four moves. The moves and their associated proposal probabilities are: growing a terminal node (0.25), pruning a pair of terminal nodes (0.25), changing a non-terminal rule (0.40), and swapping a rule between parent and child (0.10). Although the grow and prune moves change the implicit dimensionality of the proposed tree in terms of the number of terminal nodes, by integrating out M_j in (14), we avoid the complexities associated with reversible jumps between continuous spaces of varying dimensions (Green 1995).

Finally, the draw of M_j in (16) is simply a set of independent draws of the terminal node $\mu_{i,j}$'s from a normal distribution. The draw of M_j enables the calculation of the subsequent residual R_{j+1} which is critical for the next draw of T_j. Fortunately, there is again no need for a complex reversible jump implementation.

We initialize the chain with m single node trees, and then iterations are repeated until satisfactory convergence is obtained. At each iteration, each tree may increase or decrease the number of terminal nodes by one, or change one or two decision rules. Each μ will change (or cease to exist or be born), and σ will change. It is not uncommon for a tree to grow large and then subsequently collapse back down to a single node as the algorithm iterates. The sum-of-trees model, with its abundance of unidentified parameters, allows for "fit" to be freely reallocated 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.

Compared to the single tree model MCMC approach of CGM98, our backfitting MCMC algorithm mixes dramatically better. When only single tree models are considered, the MCMC algorithm tends to quickly gravitate towards a single large tree and then gets stuck in a local neighborhood of that tree. In sharp contrast, we have found that restarts of the backfitting MCMC algorithm give remarkably similar results even in difficult problems. Consequently, we run one long chain with BART rather than multiple starts. Although mixing does not appear to be an issue, the recently proposed modifications of Wu et al. (2005) might well provide additional benefits.

In some ways BART backfitting MCMC is a stochastic alternative to boosting algorithms for fitting linear combinations of trees. It is distinguished by the ability to sample from a posterior distribution. At each iteration, we get a new draw of

$$f^* = g(x; T_1, M_1) + g(x; T_2, M_2) + \ldots + g(x; T_m, M_m) \tag{17}$$

corresponding to the draw of T_j and M_j. These draws are a (dependent) sample from the posterior distribution on the "true" f. Rather than pick the "best" f^* from these draws, the set of multiple draws can be used to further enhance inference. In particular, a less variable estimator of f or predictor of Y, namely the posterior mean of f, is approximated by averaging the f^* over the multiple draws. Further, we can gauge our uncertainty about the actual underlying f by the variation across the draws. For example, we can use the 5% and 95% quantiles of $f^*(x)$ to obtain 90% posterior intervals for $f(x)$.

4 Examples

In this section we illustrate the potential of BART on two distinct types of data. The first is simulated data where the mean is the five dimensional test function used by Friedman (1991). The second is the well-known Boston Housing data which has been used to compare a wide variety of competing methods in the literature, and which is part of the machine learning benchmark package in R (mlbench).

4.1 Friedman's Five Dimensional Test Function

To illustrate the potential of multivariate adaptive regression splines (MARS), Friedman (1991) constructed data by simulating values of $x = (x_1, x_2, \ldots, x_p)$ where

$$x_1, x_2, \ldots, x_p \text{ iid } \sim Uniform(0, 1), \tag{18}$$

and y given x where

$$y = f(x) + \epsilon = 10 \sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5 + \epsilon \tag{19}$$

where $\epsilon \sim N(0,1)$. Because y only depends on $x_1, \ldots, x_5$, the predictors $x_6, \ldots, x_p$ are irrelevant. These added variables together with the interactions and nonlinearities make it especially difficult to find $f(x)$ by standard parametric methods. Friedman (1991) simulated such data for the case $p = 10$.

We now proceed to illustrate the potential of BART with this simulation setup. In Section 4.1.1, we begin with a simple application of BART. In Section 4.1.2, we elaborate this application to show BART's effectiveness at detecting low dimensional structure within high dimensional data. In Section 4.1.3, we show that BART outperforms competing methods including gradient boosting and neural nets. In Section 4.1.4, we illustrate BART's robust performance with respect to hyperparameter changes. We also see that the BART MCMC burns in fast, and mixes well.

4.1.1 A Simple Application of BART

We begin by illustrating inference on a single simulated data set of the Friedman function (18) and (19) with $p = 10$ $x's$ and $n = 100$ observations. For simplicity, we applied BART with the default setting $(\nu, q, k) = (3, 0.90, 2)$ and $m = 100$ trees as described in Section 2.2. Using the backfitting MCMC algorithm, we generated 5000 MCMC draws of f^* as in (17) from the posterior after skipping 1000 burn-in iterations. For each value of x, we obtained posterior mean estimates $\hat{f}(x)$ of $f(x)$ by averaging the 5000 $f^*(x)$ values. Endpoints of 90% posterior intervals for each $f(x)$ were obtained as the 5% and 95% quantiles of the f^* values.

Figure 2(a) plots the posterior mean estimates $\hat{f}(x)$ against the true $f(x)$ for the $n = 100$ in-sample values of x from (18) which were used to generate the y values using (19). Vertical lines indicate the 90% posterior intervals for the $f(x)$'s. Figure 2(b) is the analogous plot at 100 randomly selected out-of-sample x values. We see that the means $\hat{f}(x)$ correlate very well with the true $f(x)$

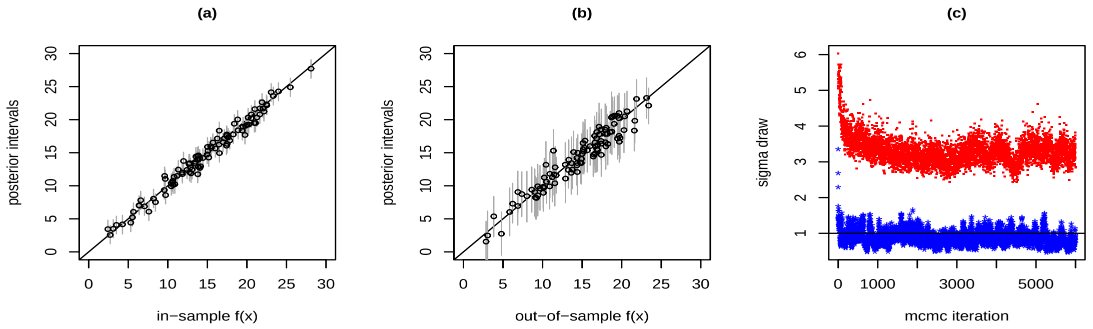

Figure 2: Inference about Friedman's function in $p = 10$ dimensions.

values and the intervals tend to cover the true values. The wider out-of-sample intervals intuitively indicate greater uncertainty about $f(x)$ at new x values.

Although we do not expect the 90% posterior intervals to exhibit 90% frequentist coverage, it may be of interest to note that 89% and 96% of the intervals in Figures 2(a) and (b) covered the true $f(x)$ value, respectively. In fact, in over 200 independent replicates of this example we found average coverage rates of 87% (in-sample) and 93% (out-of-sample). Thus, these posterior intervals may be roughly interpreted in a frequentist sense. However, it should be noted that for extreme x values, the prior may exert more shrinkage towards 0 leading to lower coverage frequencies.

The lower sequence in Figure 2(c) is the sequence of σ draws over the entire 1000 burn-in plus 5000 iterations (plotted with *). The horizontal line is drawn at the true value $\sigma = 1$. The Markov chain here appears to reach equilibrium quickly, and although there is autocorrelation, the draws of σ nicely wander around the true value $\sigma = 1$ suggesting that we have fit but not overfit. To further highlight the deficiencies of a single tree model, the upper sequence (plotted with $\cdot$) in Figure 2(c) is a sequence of σ draws when $m = 1$ is used. The sequence seems to take longer to reach equilibrium and remains substantially above the true value $\sigma = 1$, suggesting that a single tree may be inadequate to fit this data.

4.1.2 Finding Low Dimensional Structure in High Dimensional Data

Of the p variables $x_1, \ldots, x_p$ from (18), f in (19) is a function of only five $x_1, \ldots, x_5$. Thus the problem we have been considering is one of drawing inference about a five dimensional signal embedded in p dimensional space. In the previous subsection we saw that when $p = 10$, the setup used by Friedman (1991), BART could easily detect and draw inference about this five dimensional signal with just $n = 100$ observations. We now consider the same problem with substantially larger values of p to illustrate the extent to which BART can find low dimensional structure in high dimensional data. For this purpose, we repeated the analysis displayed in Figure 2 with $p = 20$, 100 and 1000 but again with only $n = 100$ observations. We used BART with the same default setting of $(\nu, q, k) = (3, 0.90, 2)$ and $m = 100$ with one exception; we used the naive estimate $\hat{\sigma}$ (the sample standard deviation of Y) rather the least squares estimate to anchor the qth prior quantile to allow for data with $p \geq n$. Note that because the naive $\hat{\sigma}$ is very likely to be larger than the least squares estimate, it would also have been reasonable to use the more aggressive prior setting for (ν, q).

Figure 3 displays the in-sample and out-of-sample BART inferences for the larger values $p = 20$, 100 and 1000. The in-sample estimates and 90% posterior intervals for $f(x)$ are remarkably good for every p. As would be expected, the out-of-sample plots show that extrapolation outside the data becomes less reliable as p increases. Indeed the estimates stray further from the truth especially at the boundaries, and the posterior intervals widen (as they should). Where there is less information, it makes sense that BART pulls towards the center because the prior takes over and the μ's are shrunk towards the center of the y values. Nonetheless it remarkable that the BART inferences are at all reliable, at least in the middle of the data, when the dimension p is so large compared to the sample size $n = 100$.

In the third column of Figure 3, it is interesting to note what happens to the

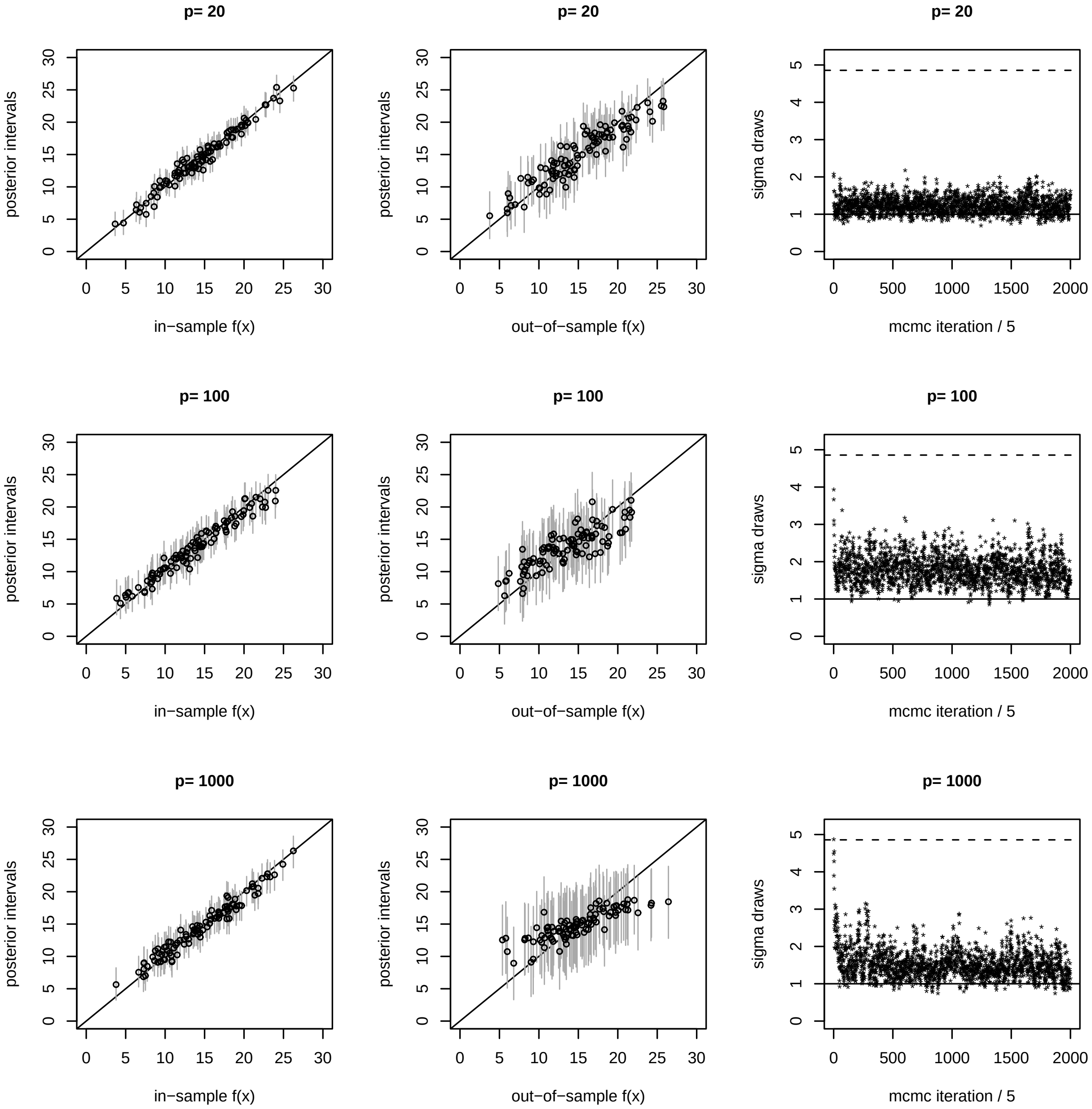

Figure 3: Inference about Friedman's function in p dimensions.

MCMC sequence of σ draws. In each of these plots, the solid line at $\sigma = 1$ is the true value and the dashed line at $\hat{\sigma} = 4.87$ is the naive estimate used to anchor the prior. In each case, the σ sequence repeatedly crosses $\sigma = 1$. However as p gets larger, it increasingly tends to stray back towards larger values, a reflection of increasing uncertainty. Lastly, note that the sequence of σ draws in Figure 3 are systematically higher than the σ draws in Figure 2(c). This is mainly due to the fact that the regression $\hat{\sigma}$ rather than the naive $\hat{\sigma}$ was used to anchor the prior in Figure 2. Indeed if the naive $\hat{\sigma}$ was instead used for Figure 2, the σ draws would similarly rise.

A further attractive feature of BART is that it appears to avoid being misled by pure noise. To gauge this, we simulated $n = 100$ observations from (18) with $f \equiv 0$ for $p = 10, 100, 1000$ and ran BART with the same settings as above. With $p = 10$ and $p = 100$ all intervals for f at both in-sample and out- of-sample x values covered or were close to 0 clearly indicating the absence of a relationship. At $p = 1000$ the data becomes so uninformative that our prior, which suggests that there is some fit, takes over and some in-sample intervals are far from 0. However, the out-of-sample intervals still tend to cover 0 and are very large so that BART still indicates no evidence of a relationship between y and x.

4.1.3 Out-of-Sample Comparisons with Competing Methods

We next proceeded to compare BART with competing methods using the Friedman simulation scenario above with $p = 10$. As plausible competitors to BART in this setting, we considered boosting (Freund & Schapire (1997), Friedman (2001)), implemented as `gbm` by Ridgeway (2004), random forests (Breiman 2001), MARS (Friedman 1991) (implemented as `polymars` by Kooperberg, Bose & Stone (1997), and neural networks, implemented as `nnet` by Venables & Ripley (2002). Least squares linear regression was also included as a reference point. All implementations are part of the R statistical software (R Development Core

Method	Parameter	Values considered
Boosting	# boosting iterations	`n.trees`= $1, 2, \ldots, 2000$
	Shrinkage (multiplier of each tree added)	`shrinkage`= 0.01, 0.05, 0.10, 0.25
	Max depth permitted for each tree	`interaction.depth`= 1,2,3,4
Neural Nets	# hidden units	`size`= 10, 15, 20, 25, 30
	Decay (penalty coef on sum-squared weights)	`decay`= 0.50, 1, 1.5, 2, 2.5
	(Max # optimizer iterations, # restarts)	fixed at `maxit`= 1000 and 5
Random Forests	# of trees	`ntree`= 200, 500, 1000
	# variables sampled to grow each node	`mtry`= 3, 5, 7, 10
MARS	GCV penalty coefficient	`gcv`= 1, 2, ..., 8
BART -cv	Sigma prior: (ν, q) combinations	(3,0.90), (3,0.99), (10,0.75)
	μ Prior: k value for σ_μ	1, 1.5, 2, 2.5, 3
	(# trees m, iterations used, burn-in iterations)	fixed at (200, 1000,500)
BART -default	Sigma prior: (ν, q) combinations	fixed at (3,0.90)
	μ Prior: k value for σ_μ	fixed at 2
	(# trees m, iterations used, burn-in iterations)	fixed at (200, 1000,500)

Table 1: Operational parameters for the various competing models. `Names` in last column indicate parameter names in R.

Team 2004). These competitors were chosen because, like BART, they are black box predictors. Trees, Bayesian CART (CGM98), and Bayesian treed regression (Chipman, George & McCulloch 2002) models were not considered, since they tend to sacrifice predictive performance for interpretability.

With the exception of linear regression, all the methods above are controlled by the operational parameters listed in Table 1. In the simulation experiment described below, we used 10-fold cross-validation for each of these methods to choose the best parameter values from the range of values also listed in Table 1. To be as fair as possible in our comparisons, we were careful to make this range broad enough so that the most frequently chosen values were not at the minimum or maximum of the ranges listed. Table 1 also indicates that some parameters were simply set to fixed values.

We considered two versions of BART in the simulation experiment. In one version, called BART-cv, the hyperparameters (ν, q, k) of the priors were treated as operational parameters to be tuned. For the σ prior hyperparameters (ν, q), the three settings (3,0.90) (default), (3,0.99)(aggressive) and (10,0.75)(conservative) as shown in Figure 1 were considered. For the μ prior hyperparameter k, five values between 1 (little shrinkage) and 3 (heavy shrinkage) were considered. Thus, 3*5 = 15 potential values of (ν, q, k) were considered. In the second version of BART, called BART-default, the operational parameters (ν, q, k) were simply fixed at the default $(3, 0.90, 2)$. For both BART-cv and BART-default, all specifications of the quantile q were made relative to the least squares regression estimate $\hat{\sigma}$. Although tuning m in BART-cv might have yielded some moderate improvement, we opted for the simpler choice of a large number of trees.

In additional to its specification simplicity, BART-default offers huge computational savings over BART-cv. Selecting among the 15 possible hyperparameter values with 10 fold cross-validation, followed by fitting the best model, requires 15*10 + 1 = 151 applications of BART. This is vastly more computationally intensive than BART-default which requires but a single fit.

The models were compared with 50 replications of the following experiment. For each replication, we set $p = 10$ and simulated 100 independent values of (x, y) from (18) and (19). Each method was then trained on these 100 in-sample values to estimate $f(x)$. Where relevant, this entailed using 10-fold cross-validation to select from the operational parameter values listed in Table 1. We next simulated 1000 out-of-sample x values from (18). The predictive performance of each method was then evaluated by the root mean squared error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{f}(x_i) - f(x_i))^2} \tag{20}$$

over the $n = 1000$ out-of-sample values.

Average RMSEs over 50 replicates and standard errors of averages are given in Table 2. All the methods explained substantial variation, since the average

Method	average RMSE	se(RMSE)
Random Forests	2.655	0.025
Linear Regression	2.618	0.016
Neural Nets	2.156	0.025
Boosting	2.013	0.024
MARS	2.003	0.060
BART-cv	1.787	0.021
BART-default	1.759	0.019

Table 2: Out-of-sample performance on 50 replicates of the Friedman data.

RMSE for the constant model ($\hat{y} \equiv \bar{y}$) is 4.87. Both BART-cv and BART-default substantially outperformed all the other methods by a significant amount. The strong performance of BART-default is noteworthy, and suggests that reasonably informed choices of prior hyperparameters may render cross-validation unnecessary. BART-default's simplicity and speed make it an ideal tool for automatic exploratory investigation. Finally, we note that BART-cv chose the default $(\nu, q, k) = (3, 0.90, 2.0)$ most frequently (20% of the replicates).

4.1.4 The Robustness of BART to Hyperparameter Choices

Yet another very appealing feature of BART is that it appears to be relatively insensitive to small changes in the prior and to the choice of m, the number of trees. Returning to the single simulated data set from Section 4.1.1, we illustrate this insensitivity by gauging the robustness of BART's performance to changes in ν, q, k and m.

Figures 4(a) and (b) display the in-sample and out-of-sample RMSE (20) obtained by BART as (ν, q, k, m) are varied. These are based on posterior mean estimates of $f(x)$ from 5000 BART MCMC draws (after skipping 1000 burn-in iterations). In each plot of RMSE versus m, the plotted text indicates the values of (ν, q, k): $k = 1$, 2 or 3 and $(\nu, q) =$ d, a or c (default/agressive/conservative). Three striking features of the plot are apparent: (i) a very small number of trees

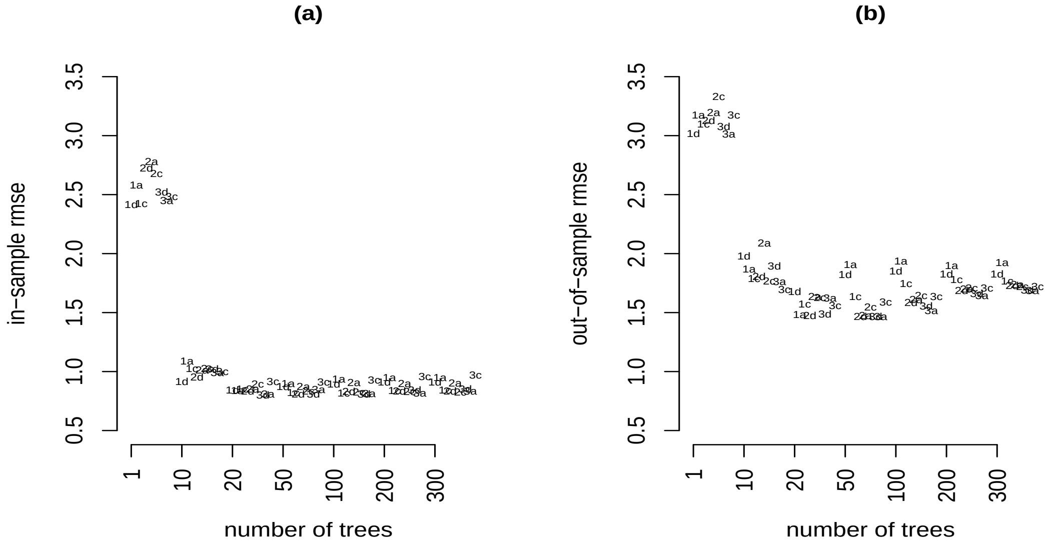

Figure 4: BART's robust RMSE performance as (ν, q, k, m) is varied: (a) in-sample RMSE comparisons and (b) out-of-sample RMSE comparisions.

(m very small) gives poor results, (ii) as long as $k > 1$, very similar results are obtained from different prior settings, and (iii) increasing the number of trees well beyond the number needed to capture the fit, results in only a slight degradation of the performance.

As Figure 4 suggests, the BART fitted values are remarkably stable as the settings are varied. Indeed, in this example, the correlations between out-of-sample fits turn out to be very high, almost always greater than .99. For example, the correlation between the fits from the (ν, q, k, m)=(3,.9,2,100) setting (a reasonable default choice) and the (10,.75,3,100) setting (a very conservative choice) is .9948. Replicate runs with different seeds are also stable: The correlation between fits from two runs with the (3,.9,2,200) setting is .9994. Such stability enables the use of one long MCMC run. In contrast, some models such as neural networks require multiple starts to ensure a good optimum has been found.

4.2 Boston Housing Data

We now proceed to illustrate the potential of BART on the Boston Housing data. This data originally appeared in Harrison & Rubinfeld (1978), and have since been used as a standard benchmark for comparing regression methods. The original study modelled the relationship between median house price for a census tract and 13 other tract characteristics, such as crime rate, transportation access, pollution, etc. The data consist of 506 census tracts in the Boston area. Following other studies, we take log median house price as the response.

4.2.1 Out-of-Sample Predictive Comparisons

We begin by comparing the performance of BART with various competitors on the Boston Housing data in a manner similar to Section 4.1.3. Because this is real rather than simulated data, a true underlying mean is unavailable, and so here we assess performance with a train/test experiment. For this purpose, we replicated 50 random 75%/25% train/test splits of the 506 observations. For each split, each method was trained on the 75% portion, and performance was assessed on the 25% portion by the RMSE between the predicted and observed y values.

As in Section 4.1.3, all the methods in Table 1 were considered, with the exception of MARS because of its poor performance on this data (see, for example, Chipman et al. (2002)). All ranges and settings for the operational parameters in Table 1 were used with the exception of neural networks for which we instead considered `size` = 3, 5, 10 and `decay` = 0.05, 0.10, 0.20, 0.50. Operational parameters were again selected by cross-validation. Both BART-cv and BART-default were considered, again with all specifications of the quantile q relative to the least squares regression estimate $\hat{\sigma}$.

Table 3 summarizes RMSE values for the 50 train/test splits, with smallest values being best. As in Table 2, both BART-cv and BART-default sig-

Method	average RMSE	se(RMSE)
Linear Regression	0.1980	0.0021
Neural Nets	0.1657	0.0030
Boosting	0.1549	0.0020
Random Forests	0.1511	0.0024
BART-default	0.1475	0.0018
BART-cv	0.1470	0.0019

Table 3: Test set performance over 50 random train/test splits of the Boston Housing data.

nificantly outperform all other methods. Furthermore, BART-default, which is trivial to specify and does not require cross-validation, performed essentially as well as BART-cv. Indeed, except for the difference between BART-cv and BART-default, all the differences in Table 3 are statistically significant (by paired t-tests that pair on the splits, at significance level $\alpha = .05$). The most commonly chosen hyperparameter combinations by BART-cv in this example were $(\nu, q, k) = (3, 0.99, 2.5)$ in 20% of the splits, followed by the default choice (3,0.90,2) in 14% of the splits.

4.2.2 Further Inference on the Full Data Set

For further illustration, we applied BART to all 506 observations of the Boston Housing data using the default setting $(\nu, q, k) = (3, 0.90, 2)$, $m = 200$, and the regression estimate $\hat{\sigma}$ to anchor q. This problem turned out to be somewhat challenging with respect to burn-in and mixing behavior: 100 iterations of the algorithm were needed before σ draws stabilized, and the σ draws had autocorrelations of 0.63, 0.54, 0.41 and 0.20 at lags 1, 2, 10, and 100, respectively. Thus, we used 10000 MCMC draws after a burn-in of 500 iterations.

At each of the 506 predictor values x, we used 5% and 95% quantiles of the MCMC draws to obtain 90% posterior intervals for $f(x)$. Not knowing the true mean $f(x)$ here of course makes it difficult to assess their coverage frequency. An appealing feature of these posterior intervals is that they widen when there is less information about $f(x)$. To roughly illustrate this, we calculated Cook's distance diagnostic D_x for each x (Cook 1977) based on a linear least squares regression of y on x. Larger D_x indicate more uncertainty about predicting y with a linear regression at x. To see how the width of the 90% posterior intervals corresponded to D_x, we plotted them together in Figure 5(a). Although the linear model may not be strictly appropriate, the plot is suggestive: all points with large D_x values have wider uncertainty bounds.

A very useful tool for gauging the actual effect of predictors using BART is the partial dependence plot developed by Friedman (2001). Suppose the vector of predictors x can be subdivided into two subgroups: the predictors of interest, x_s, and the complement $x_c = x \setminus x_s$. A prediction $f(x)$ can then be written as $f(x_s, x_c)$. To estimate the effect of x_s on the prediction, Friedman suggests the partial dependence function

$$f_s(x_s) = \frac{1}{n} \sum_{i=1}^{n} f(x_s, x_{ic}), \tag{21}$$

where x_{ic} is the ith observation of x_c in the data. Note that (x_s, x_{ic}) will generally not be one of the observed data points. Using BART it is straightforward to then estimate and even obtain uncertainty bounds for $f_s(x_s)$. A draw of $f_s^*(x_s)$ from the induced BART posterior on $f_s(x_s)$ is obtained by simply computing $f_s^*(x_s)$ as a byproduct of each MCMC draw f^*. The average of these MCMC $f_s^*(x_s)$ draws then yields an estimate of $f_s(x_s)$, and the 5% and 95% quantiles can be used to obtain 90% posterior intervals for $f_s(x_s)$.

We illustrate this by using BART to estimate the partial dependence of log median house value at 10 values of the single variable `crime`. At each distinct `crime` value x_s, $f_s(x_s)$ in (21) is defined using all $n = 506$ values of the other 12 predictors x_c in the Boston Housing data. To draw values $f_s^*(x_s)$ from the induced BART posterior on $f_s(x_s)$ at each `crime` value, we simply applied the calculation in (21) using every tenth MCMC BART draw of f^* above. With these

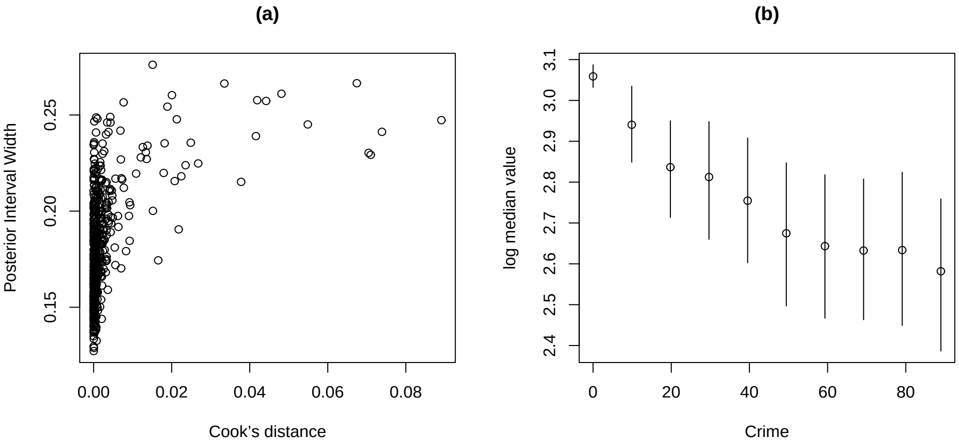

Figure 5: Plots from a single run of BART on the full Boston dataset. (a) Comparison of uncertainty bound widths with Cook's distance measure. (b) Partial dependence plot for the effect of `crime` on the response (log median property value).

1000 draws, we obtained the partial dependence plot in Figure 5(b) which shows the average estimates and 90% posterior intervals for $f_s(x_s)$ at each of 10 values of x_s. Note that the vast majority of data values occur for `crime` < 5, causing the intervals to widen as `crime` increases and the data become more sparse. At the small `crime` values, the plot suggests that the variable does have the anticipated affect on housing values.

Finally, we conclude with some remarks about the complexity of the fitted functions that BART generated to describe this data. For the last iteration, we recorded the distribution of tree sizes across the 200 trees. 7.5%, 61.5%, 26.5% and 4.5% of the trees had 1, 2, 3 and ≥ 4 terminal nodes respectively. A two-node tree indicates a main effect, since there is only a split on a single predictor. Three-node trees involving two variables indicate two-way interactions, etc. The prevalence of trees with three or fewer terminal nodes indicates that main effects and low-level interactions dominate here.

5 Discussion

The essential components of BART are the sum-of-trees model, the regularization prior and the backfitting MCMC algorithm. Although each of these components shares some common features of the Bayesian single tree approaches of CGM98 and Denison et al. (1998), their development for this framework requires substantially more than trivial extension. For example, in a sum-of-trees model, each tree only accounts for part of the overall fit. Thus to limit the creation of overly dominant terms, the regularization prior is calibrated to “weaken” the component trees by shrinking heavily towards a simple fit. This shrinkage is both in terms of the tree size (small trees are weaker learners), and in terms of the fitted values at the terminal nodes. For this purpose, we have chosen a prior that adapts the amount of shrinkage so that as the number of trees m in the model increases, the fitted values of each tree will be shrunk more. This choice helps prevent overfitting when m is large. In simulation and real-data experiments (Section 4), we have demonstrated that excellent predictive performance can still be obtained even using a very large number of trees.

To facilitate prior specification, the prior parameters themselves are expressed in terms of understandable quantities, enabling sensible selection of their values for particular data sets. Prior information about the amount of residual variation, the level of interaction involved within trees, and the anticipated number of important variables can be used to choose prior parameters. We have also indicated how these numbers could, in turn, be ballparked from simple summaries of the data such as the sample variance of Y, or of the residuals from a linear regression model. Even if these parameters are viewed from a non-Bayesian perspective as tuning parameters to be selected by cross-validation, these recommendations can provide sensible starting values and ranges for the search.

To sample from the complex and high dimensional posterior on the space of sum-of-trees models, our backfitting MCMC algorithm iteratively samples the trees, the associated terminal node parameters, and residual variance σ^2, making use of several analytic simplifications of the posterior. We find that the algorithm converges quickly to a good solution, meaning that a point estimate competitive with other ensemble methods is almost immediately available. For full inference, additional iterations are necessary.

CGM98 and Chipman et al. (2002) (CGM02) developed Bayesian methods for tree based models. BART dominates this previous work in several key dimensions. To begin with, BART gives better out-of-sample predictions. For example, Bayesian treed regression models predicted well but not as well as neural nets on the Boston Housing data, see CGM02. In contrast, BART significantly outperformed all competitors on the same data, and in simulations. Factors that may contribute to BART's predictive success include: the sum-of-tree model shrinks towards additive models but adaptively fits interactions of various levels, an effective MCMC stochastic search, model averaged posterior estimates, and regularization of the fit via prior hyperparameter choice. BART's backfitting MCMC exhibits faster burn-in, vastly better mixing and is easy to use. The CGM98 and CGM02 MCMC implementations require a number of restarts of the chain and various associated ad hoc choices. In contrast, one long run of BART MCMC works very well as evidenced by the stability of repeated runs with different seeds and different settings. Thus, the BART posterior sample can be used more reliably for estimation by the posterior mean or for construction of posterior intervals. In addition, the results seem to be remarkably robust to the prior specification. In particular, the BART default setting allows for excellent performance with an automatic specification.

Although we have framed BART as a stand alone procedure, it can also be incorporated into larger statistical models, for example, by adding other components such as linear terms or linear random effects. One can also extend the sum-of-trees model to a multivariate framework such as

$$Y_i = f_i(x_i) + \epsilon_i, \quad (\epsilon_1, \epsilon_2, \ldots, \epsilon_p) \sim N(0, \Sigma), \tag{22}$$

where each f_i is a sum of trees and Σ is a p dimensional covariance matrix. If all the x_i are the same, we have a generalization of multivariate regression. If the x_i are different we have a generalization of Zellner's SUR model (Zellner 1962). The modularity of the BART MCMC algorithm in Section 3 easily allows for such incorporations and extensions. Implementation of linear terms or random effects in a BART model would only require a simple additional MCMC step to draw the associated parameters. The multivariate version of BART (22) is easily fit by drawing each f_i^* given $\{f_j^*\}_{j \neq i}$ and Σ, and then drawing Σ given all the f_i^*.

Finally, to facilitate its use, we have provided free open-source software implementing BART as a stand-alone package or with an interface to R, along with full documentation and examples. It is available at `http://gsbwww.uchicago.edu/fac/robert.mcculloch/research`. The R library is available at `http://cran.r-project.org/`.

References

Breiman, L. (1996), 'Bagging predictors', *Machine Learning* **26**, 123–140.

Breiman, L. (2001), 'Random forests', *Machine Learning* **45**, 5–32.

Chipman, H. A., George, E. I. & McCulloch, R. E. (1998), 'Bayesian CART model search (C/R: p948-960)', *Journal of the American Statistical Association* **93**, 935–948.

Chipman, H. A., George, E. I. & McCulloch, R. E. (2002), 'Bayesian treed models', *Machine Learning* **48**, 299–320.

Cook, R. D. (1977), 'Detection of influential observations in linear regression', *Technometrics* **19**(1), 15–18.

Denison, D. G. T., Mallick, B. K. & Smith, A. F. M. (1998), 'A Bayesian CART algorithm', *Biometrika* **85**, 363–377.

Freund, Y. & Schapire, R. E. (1997), 'A decision-theoretic generalization of on-line learning and an application to boosting', *Journal of Computer and System Sciences* **55**, 119–139.

Friedman, J. H. (1991), 'Multivariate adaptive regression splines (Disc: P67-141)', *The Annals of Statistics* **19**, 1–67.

Friedman, J. H. (2001), 'Greedy function approximation: A gradient boosting machine', *The Annals of Statistics* **29**, 1189–1232.

Green, P. J. (1995), 'Reversible jump mcmc computation and Bayesian model determination', *Biometrika* **82**, 711–732.

Harrison, D. & Rubinfeld, D. L. (1978), 'Hedonic prices and the demand for clean air', *Journal of Environmental Economics and Management* **5**, 81–102.

Hastie, T. & Tibshirani, R. (2000), 'Bayesian backfitting (with comments and a rejoinder by the authors', *Statistical Science* **15**(3), 196–223.

Hastie, T., Tibshirani, R. & Friedman, J. H. (2001), *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*, Springer-Verlag Inc.

Kooperberg, C., Bose, S. & Stone, C. J. (1997), 'Polychotomous regression', *Journal of the American Statistical Association* **92**, 117–127.

R Development Core Team (2004), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3.

Ridgeway, G. (2004), *The gbm package*, R Foundation for Statistical Computing, Vienna, Austria.

Venables, W. N. & Ripley, B. D. (2002), *Modern applied statistics with S*, Springer-Verlag Inc.

Wu, Y., Tjelmeland, H. & West, M. (2005), Bayesian cart: Prior specification and posterior simulation, Technical report, Duke University.

Zellner, A. (1962), 'An efficient method of estimating seemingly unrelated regressions and testing for aggregation bias', *Journal of the American Statistical Association* **57**, 348–368.