BART:
Bayesian Additive Regression Trees

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

July 2005

Abstract

We develop a Bayesian "sum-of-trees" model where each tree is constrained by a prior to be a weak leaner. Fitting and inference are accomplished via an iterative backfitting MCMC algorithm. This model is motivated by ensemble methods in general, and boosting algorithms in particular. Like boosting, each weak learner (i.e., each weak tree) contributes a small amount to the overall model, and the training of a weak learner is conditional on the estimates for the other weak learners. The differences from boosting algorithms are just as striking as the similarities: BART is defined by a statistical model: a prior and a likelihood, while boosting is defined by an algorithm. MCMC is used both to fit the model and to quantify inferential uncertainty through the variation of the posterior draws.

The BART modelling strategy can also be viewed in the context of Bayesian nonparametrics. The key idea is to use a model which is rich enough to respond to a variety of signal types, but constrained by the prior from overreacting to weak signals. The ensemble approach provides for a rich base model form which can expand as needed via the MCMC mechanism. The priors are formulated so as to be interpretable, relatively easy to specify, and provide results that are stable across a wide range of prior hyperparameter values. The MCMC algorithm, which exhibits fast burn-in and good mixing, can be readily used for model averaging and for uncertainty assessment.

KEY WORDS: Bayesian backfitting; Boosting; CART; MCMC; 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 $f(x)$, the mean of Y given x, by a sum of m regression trees

$$f(x) = g_1(x) + g_2(x) + \ldots + g_m(x) \tag{2}$$

where each g_i denotes the fit from a regression tree model. We assume familiarity with the initial CART formulation of Breiman, Friedman, Olshen & Stone (1984), and the Bayesian CART formulations of Chipman, George & McCulloch (1998) (hereafter CGM98) and Denison, Mallick & Smith (1998), although the essentials of CGM98 are reviewed in Sections 2, 3 and 4.

A sum-of-trees model (1) and (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 model for a single tree as in CGM98. Such model averaging uses posterior probabilities as weights for averaging the predictions from individual trees.

In this paper, we propose a new Bayesian approach called BART for fitting a sum-of-trees model. 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. This is accomplished by imposing a strong prior on each of the trees to keep their effect small, effectively making them into “weak learners”. To fit each of these weak trees, BART uses a new version of a backfitting MCMC algorithm (see Hastie & Tibshirani (2000)) that iteratively constructs and fits successive residuals. Although similar to the gradient boosting approach of Friedman (2001), BART is fundamentally different in both how it weakens the individual trees by instead using a prior, and how it performs the iterative fitting by instead using MCMC. Conceptually, we are proposing a Bayesian nonparametric approach that fits a parameter rich model using substantial prior information.

Another very important feature of BART is the induced posterior over the space of sum-of-trees models, and the fact that this BART posterior can be sampled from by using the BART MCMC algorithm. Note that the BART posterior is different from the single tree model posterior proposed by CGM98 which conveys uncertainty over single tree models but not over sum-of-tree models. The

BART posterior can be readily used to further enhance inference. For example, averaging of the sum-of-tree samples from the BART posterior yields an improved model averaged estimate of f. Moreover, pointwise uncertainty bounds are available for the prediction $f(x) = E(Y|x)$ at any input value x. We will convincingly demonstrate that these uncertainty intervals behave sensibly, for example by widening for predictions at test points far from the training set.

The essential components of BART are the sum-of-trees model, the prior distribution specifications and the implementation of a BART 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, we employ priors that "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 5), we demonstrate 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 also indicate 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 BART 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. Unlike the Bayesian CART algorithm of CGM98, the BART algorithm mixes well enough that a single long chain can be used for inference.

The sum-of-trees model can be incorporated into a larger statistical model, for example, by adding other components such as linear terms or random effects. One can also extend the sum-of-trees model to a multivariate framework such as

$$Y_i = f_i(x_i) + \epsilon_i, \quad (\epsilon_1, \epsilon_2, \ldots, \epsilon_p) \sim N(0, \Sigma), \tag{3}$$

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). As will be seen in Section 4, the modularity of the BART MCMC algorithm easily allows for such incorporations and extensions.

Open-source software implementing BART as a stand-alone package or with an interface to R, along with full documentation and examples, is available from `http://gsbwww.uchicago.edu/fac/robert.mcculloch/research`.

The remainder of the paper is organized as follows. In Section 2, the model is outlined. Section 3 presents prior distributions for the parameter, and describes semi-automatic methods for hyperparameter selection. The backfitting MCMC algorithm for estimation is described in Section 4. In Section 5, simulated and real examples are used to demonstrate BART performance. The paper concludes with a discussion in Section 6.

2 The Basic 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 tree, consisting of a set of interior nodes with decision rules and a set of terminal nodes. Any x is "sent down the tree" according to the decision rules until it arrives at a terminal node to which it is assigned. We limit the decision rules to binary splits. The i^{th} terminal node is associated with a real parameter μ_i. Hence, any x is associated with one of the μ_i. Letting $M = (\mu_1, \mu_2, \ldots, \mu_b)$ where b is the number of terminal nodes of T, a single tree model may denoted by the pair (T, M). Let $g(x, T, M)$ denote the μ_i associated with x by T.

Using this notation, our sum-of-trees model (1) and (2) can be expressed as

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

When $m = 1$, $g(x, T_1, M_1)$ is the conditional mean of Y given x. However, when $m > 1$, the terminal node parameters are merely components of the conditional mean of Y given x. Furthermore, these 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 direct 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 splits on the individual components of x.

As will be seen, we obtain excellent predictive performance by setting the number of trees in the sum to be quite large (e.g., $m = 200$). With a large number of trees, the sum-of-trees model clearly gains increased representation flexibility.

There are many parameters of which only σ is identified. For example, if each tree had three terminal nodes, 200 trees would give us 600 μ's! Furthermore, each tree component has intermediate node decision rules which allow for a multitude of ways to define the partition. Somewhat surprisingly, this redundancy turns out to work to our advantage. As with any Bayesian analysis, lack of identification is not a problem given a sensible prior specification. Under our prior setup, the multitude of unidentified parameters adds tremendous mixing potential to the backfitting MCMC algorithm, enabling it to rapidly find effective representations.

3 Prior Distributions

As mentioned in the introduction, the key to our approach is a prior specification that encourages each simple tree component contribution to be small. For fixed m, a Bayesian treatment of the sum-of-trees model requires that we put priors on all the tree components $(T_1, M_1), (T_2, M_2), \ldots, (T_m, M_m)$ and on σ. This seemingly formidable task can be vastly simplified by incorporating a number of independence assumptions. To begin with, we assume the tree components are independent of σ and write

$$\begin{aligned} &p((T_1, M_1), (T_2, M_2), \ldots, (T_m, M_m), \sigma) \\ &= p(T_1, T_2, \ldots, T_m)\, p(M_1, M_2, \ldots, M_m \,|\, T_1, T_2, \ldots, T_m)\, p(\sigma). \end{aligned} \tag{5}$$

Since the dimension of each M_j depends on the corresponding T_j this conditional structure is essential. We simplify further by imposing independence whenever possible:

$$p(T_1, T_2, \ldots, T_m) = \prod p(T_j), \tag{6}$$

$$p(M_1, M_2, \ldots, M_m \,|\, T_1, T_2, \ldots, T_m) = \prod p(M_j \,|\, T_j), \tag{7}$$

$$p(M_j \,|\, T_j) = \prod p(\mu_{i,j} \,|\, T_j), \tag{8}$$

where $\mu_{i,j}$ is the ith component of M_j.

These strong independence assumptions reduce the prior choice problem to the specification of just three prior forms (6) - (8). In addition to their important computational advantages, these prior forms are controlled by just a few interpretable hyperparameters. This allows the user to express any available prior information, and to at least avoid unreasonable allocations of prior probability. In particular, an overly diffuse prior formulation would be disastrous, leading to an undue preference for smaller models. We are able to avoid this with appropriately informative hyperparameter specifications. In particular, our prior below on the $\mu_{i,j}$'s depends on m, the number of trees in a simple way which seems to be crucial to the effectiveness of our approach.

3.1 The Tree Prior

For $p(T_j)$, we follow CGM98 which is easy to specify and dovetails nicely with calculations for the Metropolis-Hastings MCMC algorithm. The prior is specified in three components: (i) the probability that a node at depth d is nonterminal, given by

$$\alpha(1+d)^{-\beta}, \qquad \alpha \in (0,1), \beta \in [0,\infty), \tag{9}$$

(ii) a distribution on the splitting variable in each nonterminal node, and (iii) a distribution on the splitting rule in each nonterminal 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, in a sum-of-trees model with m large, it is especially useful to keep the individual tree components simple. In all our examples, we do this by using $\alpha = .95$ and $\beta = 2$ in (9). 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.

Tree size can have considerable impact on the functional form and predictive accuracy of a sum of trees model. In boosting, tree size (or perhaps maximum depth) might be chosen by cross-validation. A prior distribution on tree size is more flexible: If the data demands it, trees with many terminal nodes can be grown. In one of our simulated examples, we set $m = 1$ and observe considerable posterior probability on trees of size 17, even with the above prior.

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

In our model (4), $E(Y|x)$ is the sum of m $\mu_{i,j}$'s. Given our independence assumptions, 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. The essence of our strategy is then to choose the prior mean and standard deviation 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 $E(Y|x) \in (-.5, .5)$ is very high. We then center the prior at zero and choose the standard deviation so that the mean of Y falls in the interval $(-.5, .5)$ with "high" probability. Now if the standard deviation of a single $\mu_{i,j}$ is σ_μ, then the standard deviation of the sum of m independent $\mu_{i,j}$'s is $\sqrt{m}\,\sigma_\mu$. We then 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

$$\mu_{i,j} \sim N(0, \sigma_\mu^2) \text{ where } \sigma_\mu = .5/k\sqrt{m}. \tag{10}$$

Note that for larger k, and as the number of trees m increases, this prior distribution will apply greater shrinkage to the $\mu_{i,j}$'s in each tree. 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 learners 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 can be chosen by cross-validation.

To guide the initial shifting and rescaling of Y, we usually "cheat" and use the observed y values, shifting and rescaling so that the observed y 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.

We view the extreme simplicity of our prior for $\mu_{i,j}$ as a key to the success of our approach. Because our models are based on trees, we don't have to think about the space of the explanatory variables. We just have to think in terms of Y. In contrast, methods like neural nets that use linear combinations of predictors need to pick a reasonable standardization for each predictor.

3.3 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

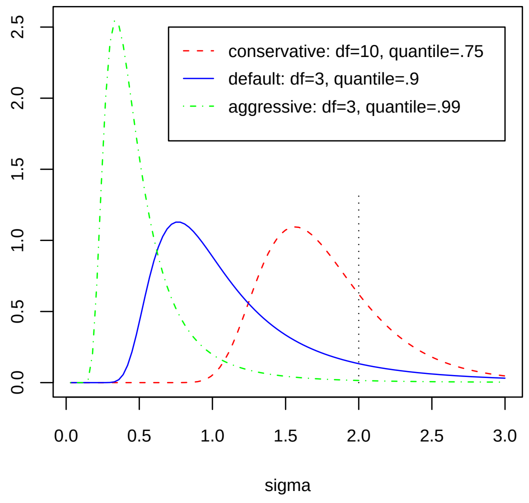

Figure 1: Three priors on σ when $\hat{\sigma} = 2$.

values. If strong prior belief that σ is very small is specified, the model can be prone 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, and 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 fit than a single mean of Y for all X values. The linear model specification represents the belief that BART can fit better than a linear model.

3.4 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. Often in applications we are "lazy" and just use a large number of trees (e.g., 200). In this regard we have found that predictive performance suffers more when too few trees are selected than when too many are selected. Although one could consider a fully Bayes approach that puts a prior on m, we find it easier and faster to simply consider and compare our results for various choices of m.

4 BART MCMC

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

$$p((T_1, M_1), (T_2, M_2), \dots, (T_m, M_m), \sigma |\, y) \tag{11}$$

on all the parameters that determine a sum-of-trees model (4). Although the sheer size of this parameter space precludes exhaustive calculation, the following 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 $(T_{(j)}, M_{(j)}, \sigma)$:

$$(T_1, M_1)|T_{(1)}, M_{(1)}, \sigma, y$$

$$(T_2, M_2)|T_{(2)}, M_{(2)}, \sigma, y \tag{12}$$

$$\vdots$$

$$(T_m, M_m)|T_{(m)}, M_{(m)}, \sigma, y,$$

followed by a draw of σ from the full conditional:

$$\sigma | T_1, T_2, \ldots T_m, M_1, M_2, \ldots, M_m, y. \tag{13}$$

Hastie & Tibshirani (2000) considered a similar application of the Gibbs sampler for posterior sampling for additive and generalized additive models, but 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 iterative backfitting MCMC.

The draw of σ in (13) 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 (12). 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

$$R_j \equiv y - \sum_{k \neq j} g(x, T_k, M_k), \tag{14}$$

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 (12) is equivalent to a draw from

$$(T_j, M_j)|R_j, \sigma. \tag{15}$$

Now (15) 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,

$$p(T_j | R_j, \sigma) \propto p(T_j) \int p(R_j | M_j, T_j, \sigma) p(M_j | T_j, \sigma) dM_j \tag{16}$$

can be obtained in closed form up to a norming constant. This allows us to carry out the draw from (15), or equivalently (12), in two successive steps as

$$T_j | R_j, \sigma \tag{17}$$

$$M_j | T_j, R_j, \sigma. \tag{18}$$

The draw of T_j in (17), 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 (16), we avoid the complexities associated with reversible jumps between continuous spaces of varying dimensions (Green 1995).

Finally, the draw of M_j in (18) 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 sculpting a complex figure by adding and subtracting small dabs of clay.

Compared to the single tree model MCMC approach of CGM98, the BART 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 get stuck in a local neighborhood of that tree. In sharp contrast, we have found that restarts of the BART 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, recently proposed modifications of Wu, Tjelmeland & West (2005) might provide additional benefits.

In some ways BART 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

$$f^* = g(x, T_1, M_1) + g(x, T_2, M_2) + \ldots + g(x, T_m, M_m) \tag{19}$$

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 $f(x)$.

Finally, the BART MCMC algorithm easily allows for the extensions of the sum-of-trees model discussed at the end of Section 1. 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 (3) 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^*.

5 Examples

In this section we illustrate the potential of BART on three 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). Finally, we report some results from Abreveya & McCulloch (2004) who use BART extensively in the analysis of data on penalty calls in the National Hockey League.

5.1 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

$$x_1, x_2, \ldots, x_p \text{ iid } \sim Uniform(0, 1), \tag{20}$$

and y given x where

$$y = f(x) + \epsilon = 10 \sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5 + \epsilon \tag{21}$$

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 5.1.1, we show that on such data, BART outperforms competing methods including gradient boosting. In Section 5.1.2, we illustrate various features of BART in more detail. There, we illustrate how BART is robust with respect to hyperparameter settings, how it can be used for in-sample and out-of-sample inference and how it remains remarkably effective when p is very large. We also see that the BART MCMC burns in fast, and mixes well.

5.1.1 Out-of-Sample Predictive Comparisons

In this section, we compare BART with competing methods using the Friedman simulation scenario above with $p = 10$. As plausible competitors to BART in this setting, we consider 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 Team 2004). These competitors were chosen because, like BART, they are black box predictors. Trees, Bayesian CART, 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)(conserva-

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.

tive) 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.

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.

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 (20) and (21). 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 (20). The predictive performance of each method was then evaluated by the root mean squared error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{f}(x_i) - f(x_i))^2} \tag{22}$$

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 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).

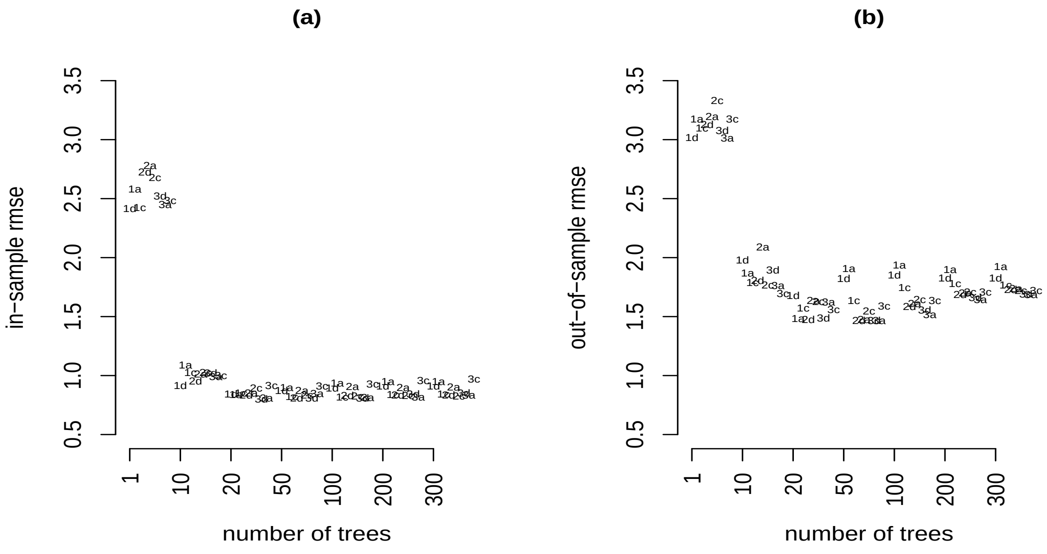

Figure 2: BART's robust RMSE performance as (ν, q, k, m) is varied

5.1.2 Further Illustration of BART's Features

We now use the simulation to further illustrate various features of BART.

We first demonstrate the robustness of BART's performance as hyperparameters are varied. In addition to varying hyperparameters ν, q and k, we varied the number of trees, m. One realization of data from Section 5.1.1 is used. We used 5000 MCMC draws to estimate $f(x)$ after skipping 1000 burn-in iterations. More draws are used than in Section 5.1.1 since we desire posterior intervals rather than just a simple point prediction.

Figures 2 (a) and (b) display the in-sample and out-of-sample RMSE (22) obtained by BART as (ν, q, k, m) are varied. The estimates $\hat{f}(x)$ used here are averages over the 5000 BART MCMC draws. In each plot, m is on the horizontal axis and RMSE is on the vertical axis. 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 small number of trees (m 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 2 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.

We next illustrate inference on the data above given by BART with the default setting $(\nu, q, k) = (3, 0.90, 2)$ and $m = 100$. Figure 3(a) plots $f(x)$ against posterior mean $\hat{f}(x)$ for the 100 in-sample values of x. Vertical lines join the 5% and 95% quantiles of the f draws. Figure 3(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 $f(x)$ 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.

In Figures 3(a) and (b), 89% and 96% of the intervals cover the true value, respectively. Over 200 replicates of the data in this figure, the mean coverage rates were 87% (in-sample) and 93% (out-of-sample). Strong prior input is a key to our approach so there is no logical necessity for exact frequency coverage. For example, at extreme x vectors our uncertainty is large and thus the prior may exert more shrinkage. In such situations we do not expect to be as successful at covering the true value as for an x in the interior. Nonetheless, if the frequentist coverage rate was way off, we would conclude that our prior was "poorly

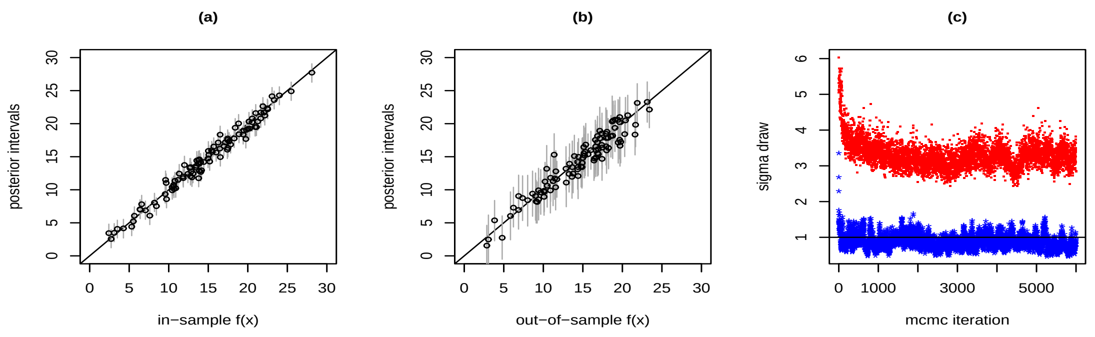

Figure 3: Inference about Friedman's function in $p = 10$ dimensions.

calibrated". The coverage rates in our simulation and the intervals depicted in Figure 3 seem very reasonable.

The lower sequence in Figure 3(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 3(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.

Lastly, we show that BART can remain effective in the Friedman simulation setup with substantially larger p. For this purpose, we repeated the analysis displayed in Figure 3 with $p = 20$, 100 and 1000. Note that we are trying to draw inference about a five dimensional function $f(x)$ embedded in a p dimensional space 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 $p \geq n$. Given that we are using the naive $\hat{\sigma}$, it might be reasonable to use the more aggressive prior setting for (ν, q).

Figure 4 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. However, 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 4, it is interesting to note what happens to the 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. Finally, we note that the sequence of σ draws in Figure 3(c) is systematically lower than the draws Figure 4. This is mainly due to the fact that the regression $\hat{\sigma}$ was used in Figure 3 to anchor the prior. Indeed if the naive $\hat{\sigma}$ was instead used the σ draws would similarly rise.

Finally, to gauge how BART would perform on pure noise, we simulated data from (20) 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.

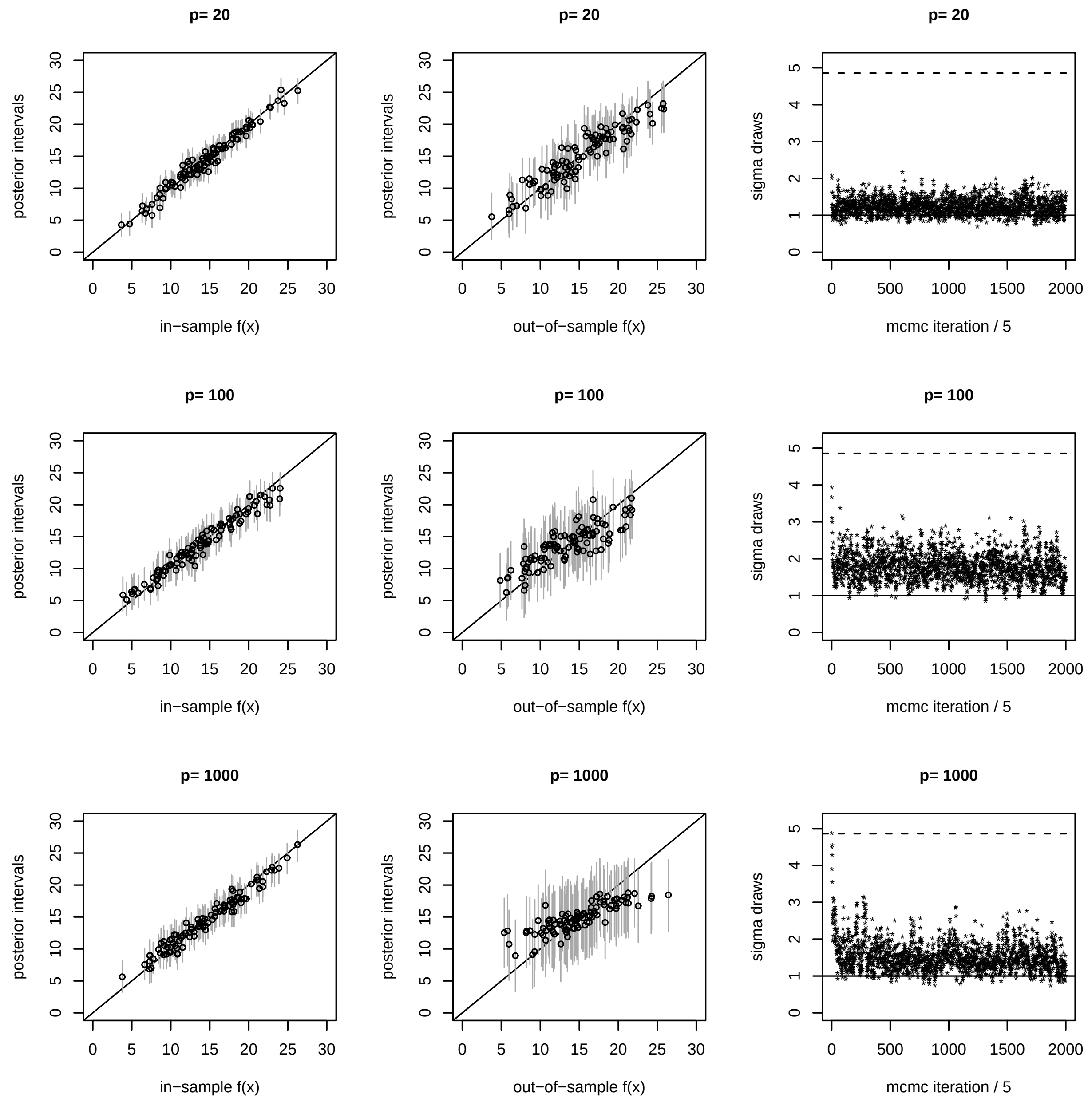

Figure 4: Inference about Friedman's function in p dimensions.

5.2 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.

5.2.1 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 5.1.1. 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 5.1.1, 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.

5.2.2 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

$$f_s(x_s) = \frac{1}{n} \sum_{i=1}^{n} f(x_s, x_{ic}), \tag{23}$$

where x_{ic} is the ith observation of x_c in the data. Note that (x_s, x_{ic}) need 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 (23) 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 (23) using every tenth MCMC BART draw of f^* above. With these

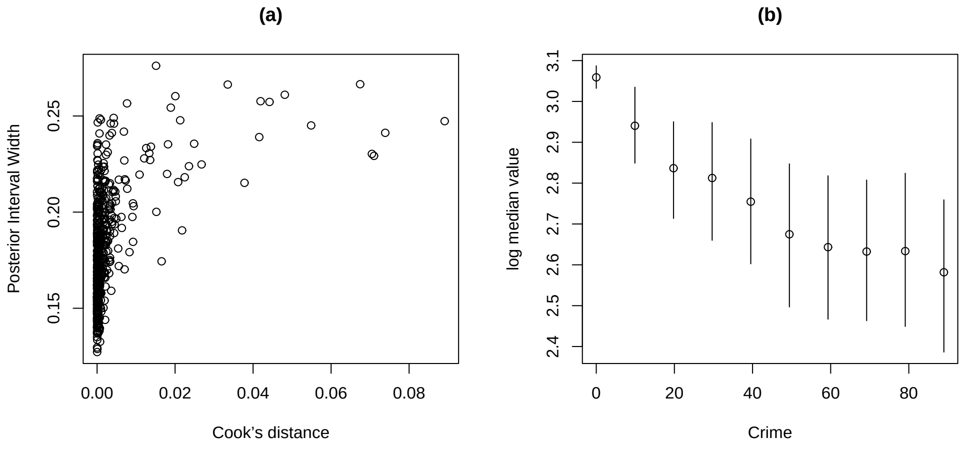

Figure 5: Plots from a single run of BART on the full Boston dataset. (a) Comparison of uncertainty bound widths with Cook's distance measure. (b) Partial dependence plot for the effect of `crime` on the response (log median property value).

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.

5.3 NHL Penalty Data

In this section we present selected results from Abreveya & McCulloch (2004) (AM) which uses BART extensively. AM collected information on every penalty in the National Hockey League from the 95-96 season to the 01-02 season. There are 57,883 observations with each observation corresponding to a penalty. The goal of this study is not strictly predictive. AM propound a theory of referee behavior and look for patterns which are consistent with this theory.

For all penalties except the first in a game, they define the response y to be 1 if the current penalty is *not* on the same team as the last penalty and 0 otherwise. AM refer to the event $y = 1$ as a “reverse call”. Nineteen explanatory variables were used. The key variables are (i) r: a binary variable where $r = 1$ indicates the last two penalties were on the same team, and $r = 0$ otherwise, (ii) g: the lead, in goals of the last team to be penalized, (iii) t: the time since the last penalty in the game, and (iv) n: the number of penalties called in the game so far. AM's basic theory is that hockey is extremely difficult to officiate and referees end up “letting a lot of stuff go”. Referees that make “too many” calls are severely criticized. Referees adopt a number of sensible strategies to avoid blame in a hostile environment. For example, we shall see that if $r = 1$ (the last two calls were on the same team) then it is more likely that $y = 1$ (the next penalty will be on the other team). Suppose the last penalized team is behind in the game. Would that make it more or less likely that the next penalty will be on the other team?

In order to assess the ability of different modelling strategies to capture the relationships in the data a simple “two-set” predictive exercise was carried out. 10,000 observations were randomly selected to be a hold-out data-set. Various modelling strategies were trained on the remaining 47,883 observations and then evaluated by their ability to predict the hold-out data. Predictive success was measured by the commonly used deviance measure: $-2\sum_{i=1}^{10,000}\log(\hat{p}_i)$, where $\hat{p}_i$

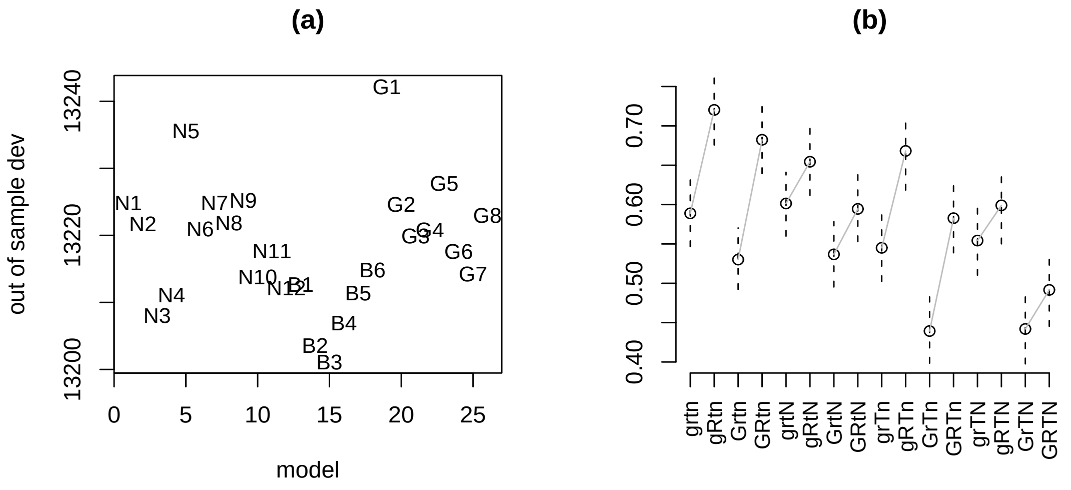

Figure 6: Hockey. (a) indicates out-of-sample deviance for neural nets (N), BART (B), and boosting (G). (b) gives predictions at 16 high/low combinations of four important predictors.

is the probability of the observed y given by a certain strategy. Note that AM are much more informal about prediction than in Sections 5.1 and 5.2 of this paper. They simply tried different modelling approaches and various tunings for each approach, fitting on one data-set and predicting on the other. This simple approach was motivated by the goal of the study which was more exploratory and interpretive than predictive.

Three models were considered for this data: neural nets, BART, and boosting. The deviance losses on hold-out data for neural nets, BART, and boosting are displayed in panel (a) of Figure 6. Various settings (described below) are considered for each model. BART with $\nu = 3$ and $\lambda = .5^2$ has the lowest out-of-sample predictive loss.

Details of the three model classes are given below:

Neural nets: One hidden layer with 10 units was used. After standardizing the explanatory variables, $N1 \ldots N6$ correspond to the use of the decay values 3,2,1,.8,.6, and .4. Some effort was needed to find a reasonable range for the decay. The next six neural net fits use the same settings as the first six and are included to illustrate the variation in neural net fits due to random starting values and the difficult nature of the likelihood for this model.

BART: While the BART approach discussed in this paper is easily extended to a binary response (e.g. via a latent continuous response), AM fit BART as in the previous examples. The model is $y = p(x) + \epsilon$, where we now think of $p(x)$ as the probability of $y = 1$ given x. Since the fitted probabilities are never extreme (you are never *sure* which team will get the next penalty) this is a reasonable approximation. All of the BART fits use 200 trees. The various BART fits correspond to different choices of the σ prior. In a departure from previous examples, AM specified λ in $\sigma^2 \sim \frac{\nu\lambda}{\chi^2_\nu}$. The first three BART fits use degrees of freedom $\nu = 10$, 5 and 3 and $\lambda = .5^2$. The second three BART fits used the same choices of ν but set $\lambda = \hat{\sigma}^2$ where $\hat{\sigma}$ is the estimate from a linear model.

Boosting: The first four boosting models (R1 through R4) have 200 trees, interaction depth of 3, and shrinkage factors of 0.05, 0.08, 0.10, and 0.15, respectively. The next four (R5 through R8) have 300 trees with the same settings repeated. AM also tried fitting `gbm` with 1000 trees but obtained worse results than reported in Figure 6.

Random forests were fit here, but not in AM. Results using the same settings as in Section 5.2 were comparable to those obtained with a single tree and very much worse the other three models considered.

AM refit BART using this prior and the entire data set. The BART representation of the function p is not directly interpretable. In order to interpret the BART fit, AM obtained the posterior distribution of $p(x)$ at a variety of x configurations each of which corresponds to a game scenario. All variables other than r, g, t, and n were fixed at a reasonable base level. A "low" and "high" level was picked for each of these four variables. A lower case symbol is used to denote the low level and an upper case symbol is used to denote the high level. For g, g denotes $g = -1$ and G denotes $g = 1$ (in goals). For r, r denotes $r = 0$ and R denotes $r = 1$. For t, t denotes $t = 2$ and T denotes $t = 10$ (in minutes). For n, n denotes $r = 3$ and N denotes $n = 12$ (number of penalties in the game so far). So, for example, $GrtN$ denotes an x corresponding to a game scenario where the last penalized team is ahead by a goal (G), the last two penalties have been on a different team (r), it has only been two minutes since the last penalty call (t), and several calls have already been made (N).

Choosing two levels for each of our four variables gives sixteen game scenarios. The posterior distribution of the probability of a reverse call under each of these scenarios is displayed in panel (b) of Figure 6. Each scenario is labelled on the horizontal axis. On the vertical axis, the mean of the draws of $p(x)$ is plotted along with a 90% interval. So, using the first scenario, the 90% posterior interval for the probability of a reverse call given the $x = grtn$ is about $.59 \pm .04$.

There is remarkable variation in the estimated probabilities and the intervals indicate that there is enough information in the data to take the differences seriously. When is a reverse call most likely? At $gRtn$ the probability is estimated to be about .72. When the referee just (t) gave a team two penalties in a row (R) and they are behind (g) and it is early in the game (n) he will really catch heat if he gives the same team another penalty. In contrast, at $GrTn$, the probability is estimated to be .45. Again the story makes sense. If the last penalized team is ahead, the last two penalties are on different teams, and the most recent penalty happened long ago, it is not too hard to give the next penalty to the same team.

In Figure 6 (b) the posteriors are presented in an order which makes it easy to see that changing from r to R always make a reverse call more likely. We also see that size of the effect depends on the levels of the other variables so that there are interactions. AM present plots which enable the reader to easily see nonlinearities in the main effects and various interactions. For example, the effect of changing from r to R is bigger at n than N, and bigger at G than g given n. BART is extremely effective in the analysis because it is able to uncover the patterns in the data including nonlinear effects and high order interactions and the posterior variation indicates which effects should be taken most seriously.

6 Discussion

CGM98 and Chipman et al. (2002) (CGM02) developed Bayesian methods for tree based models. BART dominates this previous work in several key dimensions.

BART gives better out-of-sample predictions. For example, CGM02 found that although Bayesian treed regression models predicted well, they did not do as well as neural nets on the Boston Housing data. 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 averaging, and regularization of the fit via prior hyperparameter choice. In future work we will investigate the contributions of these factors.

BART 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.

The BART model may be used as a component in a larger Bayesian hierarchical model. In such scenarios (and other situations), the notion of a hold-out sample may be difficult or impossible to define. For example we can use BART to fit an autoregressive model or when sample size n is small. In particular, if p denotes the number of explanatory variables, we can use BART in situations where $p \gg n$. In such complex modelling situations, bootstrapping a boosted model may be infeasible, and even if it can be done, it is unlikely to give a full and realistic assessment of all uncertainty.

References

Abreveya, J. & McCulloch, R. (2004), Reversal of fortune: a statistical analysis of penalty calls in the National Hockey League, Technical report, Purdue University.

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

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

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984), *Classification and Regression Trees*, CRC Press.

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.