BART:
Bayesian Additive Regression Trees

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

June, 2008

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 dimensionally adaptive random basis elements. Motivated by ensemble methods in general, and boosting algorithms in particular, BART is defined by a statistical model: a prior and a likelihood. This approach enables full posterior inference including point and interval estimates of the unknown regression function as well as the marginal effects of potential predictors. By keeping track of predictor inclusion frequencies, BART can also be used for model free variable selection. BART's many features are illustrated with a bake-off against competing methods on 42 different data sets, with a simulation experiment and on a drug discovery classification problem.

KEY WORDS: Bayesian backfitting; Boosting; CART; Classification; MCMC; Nonparametric regression; Probit model; Random basis; Regularization; Sum-of-trees model; Variable selection; Weak learner.

*Hugh Chipman is 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. A preliminary version of this paper was disseminated as a June 2006 technical report. This work was supported by NSF grant DMS-0605102, NSERC, the National Institute for Complex Data Structures and the Isaac Newton Institute for Mathematical Sciences.

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 $x = (x_1, \ldots, x_p)$ 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 $f(x) \approx h(x) \equiv \sum_{j=1}^{m} g_j(x)$ where each g_i denotes a regression tree. Thus we approximate (1) by a sum-of-trees model

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

A sum-of-trees model is fundamentally an additive model with multivariate components. Compared to generalized additive models based on sums of low dimensional smoothers, these multivariate components can more naturally incorporate interaction effects. And compared to a single tree model, the sum-of-trees can more easily incorporate additive effects.

Various methods which combine a set of tree models, so called ensemble methods, have attracted much attention. 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 (2007). 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 to model or approximate $f(x) = E(Y \mid x)$. The essential idea is to elaborate the sum-of-trees model (2) by imposing a prior that regularizes the fit by keeping the individual tree effects small. In effect, the g_j's become a dimensionally adaptive random basis of "weak learners", to borrow a phrase from the boosting literature. By weakening the g_j effects, BART ends up with a sum of trees, each of which explains a small and different portion of f. Note that BART is not equivalent to posterior averaging of single tree fits of the entire function 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.

Inferences obtained from BART are based on successive iterations of the backfitting algorithm which are effectively an MCMC sample from the induced posterior over the sum-of-trees model space. A single posterior mean estimate of $f(x) = E(Y \mid x)$ at any input value x is obtained by a simple average of these successive sum-of-trees model draws evaluated at x. Further, pointwise uncertainty intervals for $f(x)$ are easily obtained from the corresponding quantiles of the sample of draws. Point and interval estimates are similarly obtained for functionals of f, such as partial dependence functions which reveal the marginal effects of the x components. Finally, by keeping track of the relative frequency

with which x components appear in the sum-of-trees model iterations, BART can be used to identify which components are more important for explaining the variation of Y. Such variable selection information is model-free in the sense that it is not based on the usual assumption of an encompassing parametric model.

Finally, to facilitate the use of the BART methods described in this paper, we have provided 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`, and as the `BayesTree` library in R at `http://cran.r-project.org/`.

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, a Bayesian backfitting MCMC algorithm and methods for inference are described. In Section 4, we describe a probit extension of BART for classification of binary Y. In Section 5, examples, both simulated and real, are used to demonstrate the potential of BART. Section 6 describes extensions and a variety of recent developments and applications of BART based on an early version of this paper. Section 7 concludes with a discussion.

2 The BART Model

As described in the introduction, the BART model consists of two parts: a sum-of-trees model and a regularization prior on the parameters of that model. We describe each of these in detail in the following subsections.

2.1 A Sum-of-Trees Model

To elaborate the form of the sum-of-trees model (2), 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. These are typically based on the single components of $x = (x_1, \ldots, x_p)$ and are of the form $\{x_i \leq c\}$ vs $\{x_i > c\}$ for continuous x_i. 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. Under (3), the conditional mean of Y given x, $E(Y\,|x)$ equals the terminal node parameter μ_i assigned by $g(x; T, M)$.

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

$$Y = \left(\sum_{j=1}^{m} g(x; T_j, M_j) \right) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2), \tag{4}$$

where for each binary regression tree T_j and its associated terminal node parameters M_j, $g(x; T_j, M_j)$ is the function which assigns $\mu_{ij} \in M_j$ to x. Under (4), $E(Y \mid x)$ equals the sum of all the terminal node μ_{ij}'s assigned to x by the $g(x; T_j, M_j)$'s. When the number of trees $m > 1$, each μ_{ij} here is merely a part of $E(Y \mid x)$, unlike the single tree model (3). Furthermore, each such μ_{ij} will represent a main effect when $g(x; T_j, M_j)$ depends on only one component of x (i.e., a single variable), and will represent an interaction effect when $g(x; T_j, M_j)$ depends on more than one component of x (i.e., more than one variable). Thus, the sum-of-trees model can incorporate both main effects and interaction effects. And because (4) may be based on trees of varying sizes, the interaction effects may be 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, a sum of step functions 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. This representational flexibility is obtained by rapidly increasing the 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" in the sense that many different choices of $(T_1, M_1), \ldots, (T_m, M_m)$ can lead to an identical function $\sum_{j=1}^{m} g(x; T_j, M_j)$.

2.2 A Regularization Prior

We complete the BART model specification by imposing a prior over all the parameters of the sum-of-trees model, namely $(T_1, M_1), \ldots, (T_m, M_m)$ and σ. As discussed below, we advocate specifications of this prior that effectively regularize the fit by keeping the individual tree effects from being unduly influential. Without such a regularizing influence, large tree components would overwhelm the rich structure of (4), thereby limiting the advantages of the additive representation both in terms of function approximation and computation.

To facilitate the easy implementation of BART in practice, we recommend automatic default specifications below which appear to be remarkably effective, as demonstrated in the many examples of Section 5. Basically we proceed by first reducing the prior formulation problem to the specification of just a few interpretable hyperparameters which govern priors on T_j, M_j and σ. Our recommended defaults are then obtained by using the observed variation in y to gauge reasonable hyperparameter values when external subjective information

is unavailable. Alternatively one can use the computationally more demanding approach of hyperparameter selection via cross validation.

2.2.1 Prior Independence and Symmetry

Specification of our regularization prior 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

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

where $\mu_{ij} \in M_j$. Under such priors, the tree components (T_j, M_j) are independent of each other and of σ, and the terminal node parameters of every tree are independent.

The independence restrictions above simplify the prior choice problem to the specification of prior forms for just $p(T_j), p(\mu_{ij} \mid T_j)$ and $p(\sigma)$, and to simplify it further we consider identical forms for all $p(T_j)$ and for all $p(\mu_{ij} \mid T_j)$. As described in the ensuing subsections, we use the same prior forms for these as those proposed by CGM98 for Bayesian CART. In addition to their valuable computational benefits, these forms are controlled by just a few interpretable hyperparameters which can be calibrated using the data to yield effective default specifications for regularization of the sum-of-trees model. However, as will be seen, considerations for the choice of these hyperparameter values for BART are markedly different than those for Bayesian CART.

2.2.2 The T_j Prior

For $p(T_j)$, the form recommended by CGM98 is easy to specify and dovetails nicely with calculations for the backfitting MCMC algorithm described later in Section 3.1. It 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 5, 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, which puts most probability on tree sizes of 2 or 3, 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 $m = 1$.

2.2.3 The $\mu_{ij} \,|\, T_j$ Prior

For $p(\mu_{ij} \,|\, T_j)$, we use the conjugate normal distribution $N(\mu_\mu, \sigma_\mu^2)$ which offers tremendous computational benefits because μ_{ij} can be margined out. To guide the specification of the hyperparameters μ_μ and σ_μ, note that $E(Y \,|\, x)$ is the sum of m μ_{ij}'s under the sum-of-trees model, and because the μ_{ij}'s are apriori iid, the

induced prior on $E(Y \mid x)$ is $N(m\,\mu_\mu, m\,\sigma_\mu^2)$. Note also that it is highly probable that $E(Y \mid x)$ is between y_{min} and y_{max}, the observed minimum and maximum of Y in the data. The essence of our strategy is then to choose μ_μ and σ_μ so that $N(m\,\mu_\mu, m\,\sigma_\mu^2)$ assigns substantial probability to the interval (y_{min}, y_{max}). This can be conveniently done by choosing μ_μ and σ_μ so that $m\,\mu_\mu - k\,\sqrt{m}\,\sigma_\mu = y_{min}$ and $m\,\mu_\mu + k\,\sqrt{m}\,\sigma_\mu = y_{max}$ for some preselected value of k. For example, $k = 2$ would yield a 95% prior probability that $E(Y \mid x)$ is in the interval (y_{min}, y_{max}).

The strategy above uses an aspect of the observed data, namely y_{min} and y_{max}, to try to ensure that the implicit prior for $E(Y \mid x)$ is in the right "ballpark". That is to say, we want it to assign substantial probability to the entire region of plausible values of $E(Y \mid x)$ while avoiding overconcentration and overdispersion. We have found that, as long as this goal is met, BART is very robust to our exact specification. Such a data-informed prior approach is especially useful in our problem, where reliable subjective information about $E(Y \mid x)$ is likely to be unavailable.

For convenience, we implement our specification strategy by first shifting and rescaling Y so that the observed transformed y values range from $y_{min} = -0.5$ to $y_{max} = 0.5$, and then treating this transformed Y as our dependent variable. We then simply center the prior for μ_{ij} at zero $\mu_\mu = 0$ and choose σ_μ so that $k\sqrt{m}\sigma_\mu = 0.5$ for a suitable value of k, yielding

$$\mu_{ij} \sim N(0, \sigma_\mu^2) \text{ where } \sigma_\mu = 0.5/k\sqrt{m}. \tag{8}$$

This prior has the effect of shrinking the tree parameters μ_{ij} towards zero, limiting the effect of the individual 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 μ_{ij}'s. Prior shrinkage on μ_{ij}'s is the counterpart of the shrinkage parameter in Friedman's (2001) gradient boosting algorithm. The prior standard deviation σ_μ of μ_{ij} 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 linear 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 simplicity of our prior for μ_{ij} is an appealing feature of BART. In contrast, methods like neural nets that use linear combinations of predictors require standardization choices for each predictor.

2.2.4 The σ Prior

For $p(\sigma)$, we also use a conjugate prior, here the inverse chi-square distribution $\sigma^2 \sim \nu\,\lambda/\chi^2_\nu$. To guide the specification of the hyperparameters ν and λ, we again use a data-informed prior approach, in this case to assign substantial probability to the entire region of plausible values of σ while avoiding overconcentration and overdispersion. Essentially, we calibrate ν and λ for this purpose using a "rough data-based overestimate" $\hat{\sigma}$ of σ. Two natural choices for $\hat{\sigma}$ are 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. We then pick a value of ν between 3 and 10 to get an appropriate shape, and a value of λ so that the qth quantile of the prior on σ is located at $\hat{\sigma}$, that is $P(\sigma < \hat{\sigma}) = q$. We consider values of q such as 0.75, 0.90 or 0.99 to center the distribution below $\hat{\sigma}$.

Figure 1 illustrates priors corresponding to three (ν, q) settings when the rough overestimate is $\hat{\sigma} = 2$. We refer to these three settings, $(\nu, q) = (10, 0.75)$,

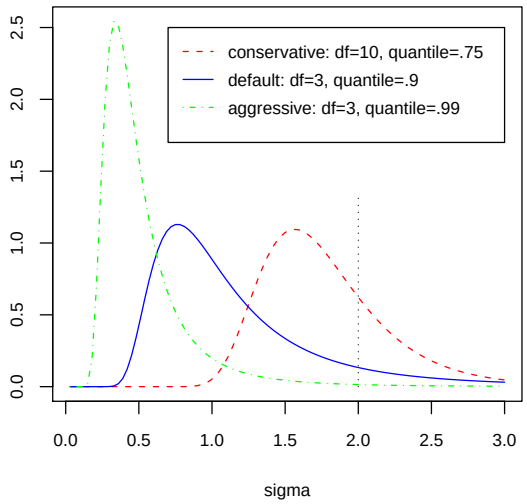

Figure 1: Three priors on σ when $\hat{\sigma} = 2$.

$(3, 0.90)$, $(3, 0.99)$, 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. Alternatively, the values of (ν, q) may be chosen by cross-validation.

2.2.5 The Choice of m

A major difference between BART and boosting methods is that for a fixed number of trees m, BART uses an iterative backfitting algorithm (described in Section 3.1) to cycle over and over through the m trees. If BART is to be used for estimating $f(x)$ or predicting Y, it might be reasonable to treat m as an unknown parameter, putting a prior on m and proceeding with a fully Bayes implementation of BART. Another reasonable strategy might be to select a "best" value for m by cross-validation. However, both of these strategies substantially increase computational requirements.

To avoid the computational costs of these strategies, we have found it fast and expedient for estimation and prediction to begin with a default of $m = 200$, and then perhaps to check if one or two other choices makes any difference. Our experience has been that as m is increased, starting with $m = 1$, the predictive performance of BART improves dramatically until at some point it levels off and then begins to very slowly degrade for large values of m. Thus, for prediction, it seems only important to avoid choosing m too small. As will be seen in Section 5, BART yielded excellent predictive performance on a wide variety of examples with the simple default $m = 200$.

However, as mentioned in Section 1, BART can also be used for variable selection by simply selecting those variables that appear most often in the fitted sum-of-trees models. Interestingly, this strategy does not seem to work so well when m is large, because the sum-of-trees model incorporates many irrelevant predictions in addition to the relevant ones, a consequence of the rich flexibility granted by the redundancy of having so many trees. However, as m is decreased and that redundancy is diminished, BART tends to heavily favor relevant predictors for its fit. In a sense, when m is small the predictors compete with each other to improve fit so that those that appear more often are likely to be more useful for prediction. As discussed in Section 3.2 and illustrated in Section 5, a useful implementation of this strategy is to use BART as an exploratory tool by observing what happens to component inclusion frequencies as m is decreased.

3 Extracting Information from the Posterior

3.1 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 the 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 $(T_{(j)}, M_{(j)}, \sigma)$:

$$(T_j, M_j)|T_{(j)}, M_{(j)}, \sigma, y, \tag{10}$$

$j = 1, \ldots, m$, 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. 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, the m draws of (T_j, M_j) given $(T_{(j)}, M_{(j)}, \sigma, y)$ in (10) are equivalent to m draws from

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

$j = 1, \ldots, m$.

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 each draw from (13) 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 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 μ_{ij}'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 simple 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. (2007) might well provide additional benefits.

3.2 Posterior Inference Statistics

The backfitting algorithm described in the previous section is ergodic, generating a sequence of draws of $(T_1, M_1), \ldots, (T_m, M_m), \sigma$ which is converging (in distribution) to the posterior $p((T_1, M_1), \ldots, (T_m, M_m), \sigma \,|\, y)$. The induced sequence of sum-of-trees functions

$$f^*(\cdot) = \sum_{j=1}^{m} g(\cdot; T_j^*, M_j^*), \tag{17}$$

for the sequence of draws $(T_1^*, M_1^*), \ldots, (T_m^*, M_m^*)$, is thus converging to $p(f \,|\, y)$, the posterior distribution on the "true" $f(\cdot)$. Thus, by running the algorithm long enough after a suitable burn-in period, the sequence of f^* draws, say $f_1^*, \ldots, f_K^*$, may be regarded as an approximate, dependent sample from $p(f \,|\, y)$. Bayesian inferential quantities of interest can then be approximated with this sample as

indicated below. Although the number of iterations needed for reliable inferences will of course depend on the particular application, our experience with the examples in Section 5 suggests that the number of iterations required is relatively modest.

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

$$\frac{1}{K}\sum_{k=1}^{K} f_k^*(x), \tag{18}$$

which approximates the posterior mean $E(f(x) \,|\, y)$. Another good choice would be the median of $f_1^*(x), \ldots, f_K^*(x)$ which approximates the posterior median of $f(x)$. Posterior uncertainty about $f(x)$ may be gauged by the variation of $f_1^*(x), \ldots, f_K^*(x)$. For example, a natural and convenient $(1-\alpha)\%$ posterior interval for $f(x)$ is obtained as the interval between the upper and lower $\alpha/2$ quantiles of $f_1^*(x), \ldots, f_K^*(x)$. As will be seen, these uncertainty intervals behave sensibly, for example by widening at x values far from the data.

It is also straightforward to use $f_1^*(x), \ldots, f_K^*(x)$ to estimate other functionals of f. For example, a functional of particular interest is the partial dependence function, (Friedman 2001), which summarizes the marginal effect of one (or more) predictors on the response. More precisely, letting $f(x) = f(x_s, x_c)$ where x has been partitioned into the predictors of interest, x_s and the complement $x_c = x \backslash x_s$, the partial dependence function is defined as

$$f(x_s) = \frac{1}{n}\sum_{i=1}^{n} f(x_s, x_{ic}), \tag{19}$$

where x_{ic} is the ith observation of x_c in the data. Note that (x_s, x_{ic}) will not generally be one of the observed data points. A draw from the induced BART posterior $p(f(x_s) \,|\, y)$ at any value of x_s is obtained by simply computing $f_k^*(x_s) = \frac{1}{n}\sum_i f_k^*(x_s, x_{ic})$. The average of $f_1^*(x_s), \ldots, h_K^*(x_s)$ then yields an estimate of $f(x_s)$, and the upper and lower $\alpha/2$ quantiles provide endpoints of $(1-\alpha)\%$ posterior intervals for $f(x_s)$.

As discussed in Section 2.2.5, BART can also be used as an exploratory tool for variable selection. This can be accomplished by observing what happens to the x component frequencies in a sequence of MCMC samples $f_1^*, \ldots, f_K^*$ as the number of trees m is set smaller and smaller. More precisely, for each simulated sum-of-trees model f_k^*, let z_{ik} be the proportion of all splitting rules that use the ith component of x. Then

$$v_i \equiv \frac{1}{K} \sum_{k=1}^{K} z_{ik} \tag{20}$$

is the average use per splitting rule for the ith component of x. As m is set smaller and smaller, the sum-of-trees models tend to more strongly favor inclusion of those x components which improve prediction of y and exclusion of those x components that are unrelated to y. In effect, smaller m seems to create a bottleneck that forces the x components to compete for entry into the sum-of-trees model. As is illustrated in Section 5, the x components with the larger v_i's will then be those that provide the most information for predicting y.

4 Extending BART for Classification

Our development of BART up to this point pertains to setups where the output of interest Y is a continuous variable. However, for binary Y (= 0 or 1), it is straightforward to extend BART to the probit model setup for classification

$$p(x) \equiv P[Y = 1 \,|\, x] = \Phi[G(x)] \tag{21}$$

where

$$G(x) \equiv \sum_{j=1}^{m} g(x; T_j, M_j) \tag{22}$$

and $\Phi[\cdot]$ is the standard normal cdf. For this extension of BART, we need to impose a regularization prior on $G(x)$ and to implement a Bayesian backfitting algorithm for posterior computation. Fortunately, these are obtained with only minor modifications of the methods in Sections 2 and 3.

As opposed to (4), the model (21) does not involve σ and so we need only put a prior on only $(T_1, M_1), \ldots, (T_m, M_m)$. Proceeding exactly as in Section 2.2.1, we consider a prior of the form

$$p((T_1, M_1), \ldots, (T_m, M_m)) = \prod_j \left[p(T_j) \prod_i p(\mu_{ij} \mid T_j) \right] \tag{23}$$

where each tree prior $p(T_j)$ is the choice recommended in Section 2.2.2. For the choice of $p(\mu_{ij} \mid T_j)$ here, we consider the case where the interval $(\Phi[-3.0], \Phi[3.0])$ contains most of the $p(x)$ values of interest, a case which will often be of practical relevance. Proceeding similarly to the motivation of (8) in Section 2.2.3, we would then recommend the choice

$$\mu_{ij} \sim N(0, \sigma_\mu^2) \text{ where } \sigma_\mu = 3.0/k\sqrt{m} \tag{24}$$

where k is such that $G(x)$ will with high probability be in the interval $(-3.0, 3.0)$. Just as for (8), this prior has the effect of shrinking the tree parameters μ_{ij} towards zero, limiting the effect of the individual tree components of $G(x)$. As k and/or the number of trees m is increased, this prior will become tighter and apply greater shrinkage to the μ_{ij}'s. 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.

By shrinking $G(x)$ towards 0, the prior (24) has the effect of shrinking $p(x) = \Phi[G(x)]$ towards 0.5. If it is of interest to shrink towards a value p_0 other than 0.5, one can simply replace $G(x)$ by $G_c = G(x) + c$ in (21) where $c = \Phi^{-1}[p_0]$. Note also that if an interval other than $(\Phi[-3.0], \Phi[3.0])$ is of interest for $p(x)$, suitable modification of (24) is straightforward.

Turning to posterior calculation, the essential features of the backfitting algorithm in Section 3.1 can be implemented by using the augmentation idea of Albert & Chib (1993). The key idea is to recast the model (21) by introducing

latent variables $Z_1, \ldots, Z_n$ iid $\sim N(G(x), 1)$ such that $Y_i = 1$ if $Z_i > 0$ and $Y_i = 0$ if $Z_i \leq 0$. Note that under this formulation, $Z_i \,|\, [y_i = 1] \sim \max\{N(g(x), 1), 0\}$ and $Z_i \,|\, [y_i = 0] \sim \min\{N(g(x), 1), 0\}$. Incorporating simulation of the latent Z_i values into the backfitting algorithm, the Gibbs sampler iterations here entail n successive draws of $Z_i \,|\, y_i$, $i = 1, \ldots, n$ followed by m successive draws of $(T_j, M_j)|T_{(j)}, M_{(j)}, z_1, ..., z_n$, $j = 1, \ldots, m$, as spelled out in Section 3.1. The induced sequence of sum-of-trees functions

$$p^*(\cdot) = \Phi\left[\sum_{j=1}^{m} g(\cdot; T_j^*, M_j^*)\right], \tag{25}$$

for the sequence of draws $(T_1^*, M_1^*), \ldots, (T_m^*, M_m^*)$, is thus converging to the posterior distribution on the "true" $p(\cdot)$. After a suitable burn-in period, the sequence of g^* draws, say $g_1^*, \ldots, g_K^*$, may be regarded as an approximate, dependent sample from this posterior which can be used to draw inference about $p(\cdot)$ in the same way that $f_1^*, \ldots, f_K^*$ was used in Section 3.2 to draw inference about $f(\cdot)$.

5 Applications

In this section we demonstrate the application of BART on several examples. We begin in Section 5.1 with a predictive cross-validation performance comparison of BART with competing methods on 42 different real data sets. We next, in Section 5.2, evaluate and illustrate BART's capabilities on simulated data used by Friedman (1991). Finally, in Section 5.3 we apply the BART probit model to a drug discovery classification problem. All of the BART calculations throughout this section can be reproduced with the `BayesTree` library at `http://cran.r-project.org/`.

5.1 Predictive Comparisons on 42 Data Sets

Our first illustration is a "bake-off", a predictive performance comparison of BART with competing methods on 42 different real data sets. These data sets are a subset of 52 sets considered by Kim, Loh, Shih & Chaudhuri (2007). Ten data sets were excluded either because Random Forests was unable to use over 32 categorical predictors, or because a single train/test split was used in the original paper. All data sets correspond to regression setups with between 3 and 28 numeric predictors and 0 to 6 categorical predictors. Categorical predictors were converted into 0/1 indicator variables corresponding to each level. Sample sizes vary from 96 to 6806 observations. In each of the 42 data sets, the response was minimally preprocessed, applying a log or square root transformation if this made the histogram of observed responses more bell-shaped. In about half the cases, a log transform was used to reduce a right tail. In one case (Fishery) a square root transform was most appropriate.

For each of the 42 data sets, we created 20 independent train/test splits by randomly selecting 5/6 of the data as a training set and the the remaining 1/6 as a test set. Thus, $42 \times 20 = 840$ test/train splits were created. Based on each training set, each method was then used to predict the corresponding test set and evaluated on the basis of its predictive RMSE.

We considered two versions of BART: BART-cv where the prior hyperparameters (ν, q, k, m) were treated as operational parameters to be tuned via cross-validation, and BART-default where we set (ν, q, k, m) to the defaults $(3, 0.90, 2, 200)$. For both BART-cv and BART-default, all specifications of the quantile q were made relative to the least squares linear regression estimate $\hat{\sigma}$, and the number of burn-in steps and MCMC iterations used were determined by inspection of a single long run. Typically 200 burn-in steps and 1000 iterations were used. For BART prediction at each x, we used the posterior mean estimates given by (18).

As competitors we considered linear regression with L1 regularization (the

Lasso) (Efron, Hastie, Johnstone & Tibshirani 2004) and three black-box models: gradient boosting (Friedman (2001), implemented as `gbm` in R by Ridgeway (2004)), random forests (Breiman 2001), and neural networks with one layer of hidden units, (implemented as `nnet` in R by Venables & Ripley (2002)). 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 BART-default (which requires no tuning), the operational parameters of every method were chosen via 5-fold cross-validation within each training set. The parameters considered and potential levels are given in Table 1. In particular, for BART-cv, we considered

- three settings (3,0.90) (default), (3,0.99)(aggressive) and (10,0.75)(conservative) as shown in Figure 1 for the σ prior hyperparameters (ν, q),
- three values $k = 2, 3, 5$ reflecting moderate to heavy shrinkage for the μ prior hyperparameter, and
- two values $m = 50, 200$ for the number of trees,

a total of 3*3*2 = 18 potential choices for (ν, q, k, m).

All the levels in Table 1 were chosen with a sufficiently wide range so that the selected value was not at an extreme of the candidate values in most problems. Neural networks are the only model whose operational parameters need additional explanation. In that case, the number of hidden units was chosen in terms of the implied number of weights, rather than the number of units. This design choice was made because of the widely varying number of predictors across problems, which directly impacts the number of weights. A number of hidden units were chosen so that there was a total of roughly u weights, with $u = 50, 100, 200, 500$ or 800. In all cases, the number of hidden units was further constrained to fall between 3 and 30. For example, with 20 predictors we used

Method	Parameter	Values considered
BART-cv	Sigma prior: (ν, q) combinations	(3,0.90), (3,0.99), (10,0.75)
	# trees m	50, 200
	μ Prior: k value for σ_μ	2, 3, 5
Lasso	shrinkage (in range 0-1)	0.1, 0.2, ..., 1.0
Gradient Boosting	# of trees	50, 100, 200
	Shrinkage (multiplier of each tree added)	0.01, 0.05, 0.10, 0.25
	Max depth permitted for each tree	1, 2, 3, 4
Neural Nets	# hidden units	see text
	Weight decay	.0001,.001, .01, .1, 1, 2, 3
Random Forests	# of trees	500
	% variables sampled to grow each node	10, 25, 50, 100

Table 1: Operational parameters for the various competing models.

3, 8 and 21 as candidate values for the number of hidden units.

To facilitate performance comparisons across data sets, we considered relative RMSE (RRMSE) which we defined as the RMSE divided by the minimum RMSE obtained by any method for each test/train split. Thus a method obtained an RRMSE of 1.0 when that method had the minimum RMSE on that split. As opposed to the RMSE, the RRMSE provides meaningful comparisons across data sets because of its invariance to location and scale transformations of the response variables. Boxplots of the 840 test/train split RRMSE values for each method are shown in Figure 2, and the median RRMSE values (the center of each box in Figure 2) are given in Table 2. (The Lasso was left off the boxplots because its many large RRMSE values visually overwhelmed the other comparisons).

It is clear from the distribution of RRMSE values in Figure 2 that BART-cv tended to more often obtain smaller RMSE than any of its competitors. Perhaps even more interesting is that the overall performance of BART-default was arguably second best. This is especially impressive since neural nets, random

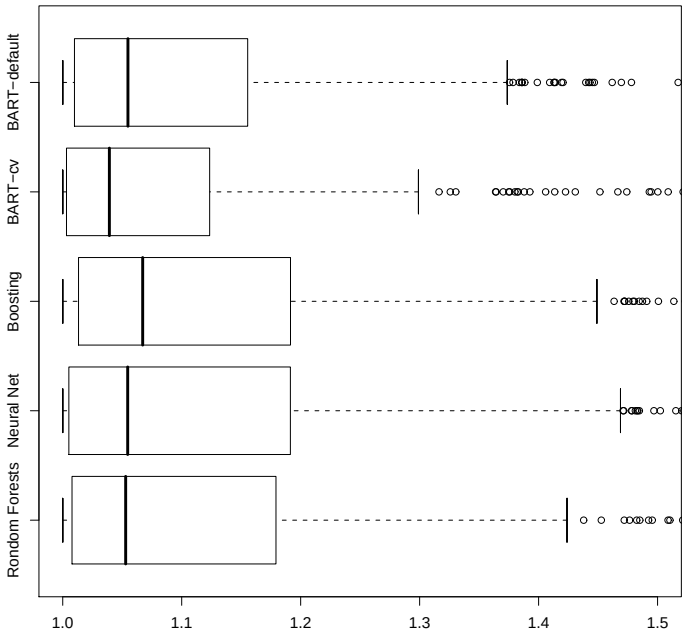

Figure 2: Boxplots of the RRMSE values for each method across the 840 test/train splits. Percentage RRMSE values larger than 1.5 for each method (and not plotted) were: Random forests 15.6%, Neural net 8.6%, Boosting 12.7%, BART-cv 9.0% and BART-default 11.2%. The Lasso (not plotted because of too many large RRMSE values), had 29.6% greater than 1.5.

Method	Median
Lasso	1.195
Boosting	1.067
BART-default	1.055
Neural Net	1.055
Random Forests	1.053
BART-cv	1.039

Table 2: Median RRMSE values for each method across the 840 test/train splits.

forests and gradient boosting all relied here on cross validation for control parameter tuning. By avoiding the need for parameter specification, BART-default is vastly easier and faster to use. For example, a single implementation of BART-cv here requires selection among the 18 possible hyperparameter values with 5 fold cv, followed by fitting the best model, for a total of 18*5 + 1 = 91 applications of BART. Ready for easy "off the shelf" use, computationally inexpensive and coherent from a Bayesian point of view (cross validation is not), BART-default is in a sense the real winner in this experiment. It may also be of interest to note that, as opposed to BART-cv, posterior interval estimates based on BART-default retain their Bayesian validity.

5.2 Friedman's Five Dimensional Test Function

We next proceed to illustrate various features of BART on simulated data where we can gauge its performance against the true underlying signal. For this purpose, we constructed data by simulating values of $x = (x_1, x_2, \ldots, x_p)$ where

$$x_1, x_2, \ldots, x_p \text{ iid } \sim \text{Uniform}(0, 1), \tag{26}$$

and y given x where

$$y = f(x) + \epsilon = 10 \sin(\pi x_1 x_2) + 20(x_3 - .5)^2 + 10x_4 + 5x_5 + \epsilon \tag{27}$$

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 more challenging to find $f(x)$ by standard parametric methods. Friedman (1991) used this setup with $p = 10$ to illustrate the potential of multivariate adaptive regression splines (MARS).

In Section 5.2.1, we illustrate various basic features of BART. We illustrate point and interval estimation of $f(x)$, model-free variable selection and estimation of partial dependence functions. We see that the BART MCMC burns-in

quickly and mixes well. We illustrate BART's very robust performance with respect to hyperparameter changes. In Section 5.2.2, we show that BART outperforms competing methods including gradient boosting and neural nets. In Section 5.2.3, we elaborate this application to show BART's effectiveness at detecting low dimensional structure within high dimensional data.

5.2.1 A Simple Application of BART

We begin by illustrating the basic features of BART on a single simulated data set of the Friedman function (26) and (27) with $p = 10$ $x's$ and $n = 100$ observations. For simplicity, we applied BART with the default setting $(\nu, q, k, m) = (3, 0.90, 2, 200)$ 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.

To begin with, for each value of x, we obtained posterior mean estimates $\hat{f}(x)$ of $f(x)$ by averaging the 5000 $f^*(x)$ values as in (18). Endpoints of 90% posterior intervals for each $f(x)$ were obtained as the 5% and 95% quantiles of the f^* values. Figure 3(a) plots $\hat{f}(x)$ against $f(x)$ for the $n = 100$ in-sample values of x from (26) which were used to generate the y values using (27). Vertical lines indicate the 90% posterior intervals for the $f(x)$'s. Figure 3(b) is the analogous plot at 100 randomly selected out-of-sample x values. We see that in-sample the $\hat{f}(x)$ values correlate very well with the true $f(x)$ values and the intervals tend to cover the true values. Out-of sample, there is a slight degradation of the correlation and wider intervals indicating 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 3(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%

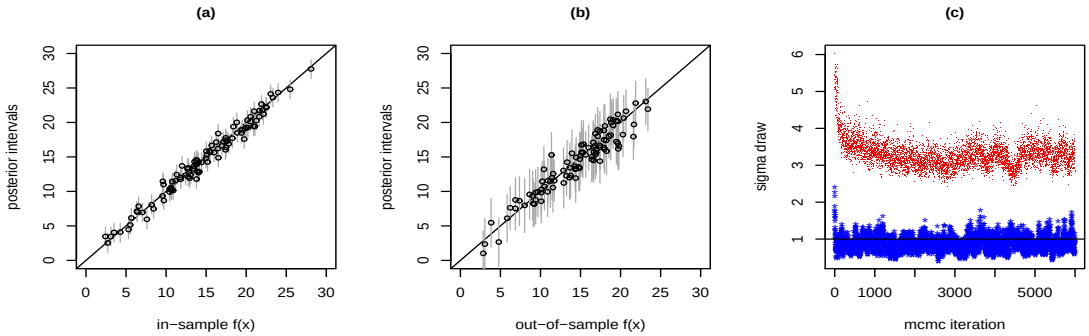

Figure 3: Inference about Friedman's $f(x)$ in $p = 10$ dimensions.

(in-sample) and 93% (out-of-sample). In real data settings where f is unknown, bootstrap and/or cross-validation methods might be helpful to get similar calibrations of frequentist coverage. In any case, 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 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 $m = 1$, a single tree model, is used. The sequence seems to take longer to reach equilibrium and remains substantially above the true value $\sigma = 1$. Evidently a single tree is inadequate to fit this data.

Moving beyond estimation and inference about the values of $f(x)$, BART estimates of the partial dependence functions $f(x_i)$ in (19) reveal the marginal effects of the individual x_i's on y. Figure 4 shows the plots of point and interval estimates of the partial dependence functions for $x_1, \ldots, x_{10}$ from the 5000 MCMC samples of f^*. The nonzero marginal effects of $x_1, \ldots, x_5$ and the zero marginal effects of $x_6, \ldots, x_{10}$ seem to be completely consistent with the form of

f which of course would be unknown in practice.

As described in Section 3.2, BART can also be used as a screening method for model-free variable selection information by keeping track of the average use per splitting rule v_i in (20) for each x_i over MCMC samples of f^* for various choices of m. Figure 5 plots these v_i's for $x_1, \ldots, x_{10}$ for $m = 10, 20, 50, 100, 200$. Notice that as the number of trees m is made smaller, the fitted sum-of-trees model tends to incorporate only those x variables, namely $x_1, \ldots, x_5$, that are needed to explain the variation of y. Thus, if one did not know the form of f here, selecting those variables which appear most often in the fitted sum-of-tree models for small m would have been effective.

Yet another very appealing feature of BART is that it appears to be very robust to small changes in the prior and to the choice of m, the number of trees. This robustness is illustrated in in Figures 6(a) and (b) which display the in- and out-of-sample RMSE 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 (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 6 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

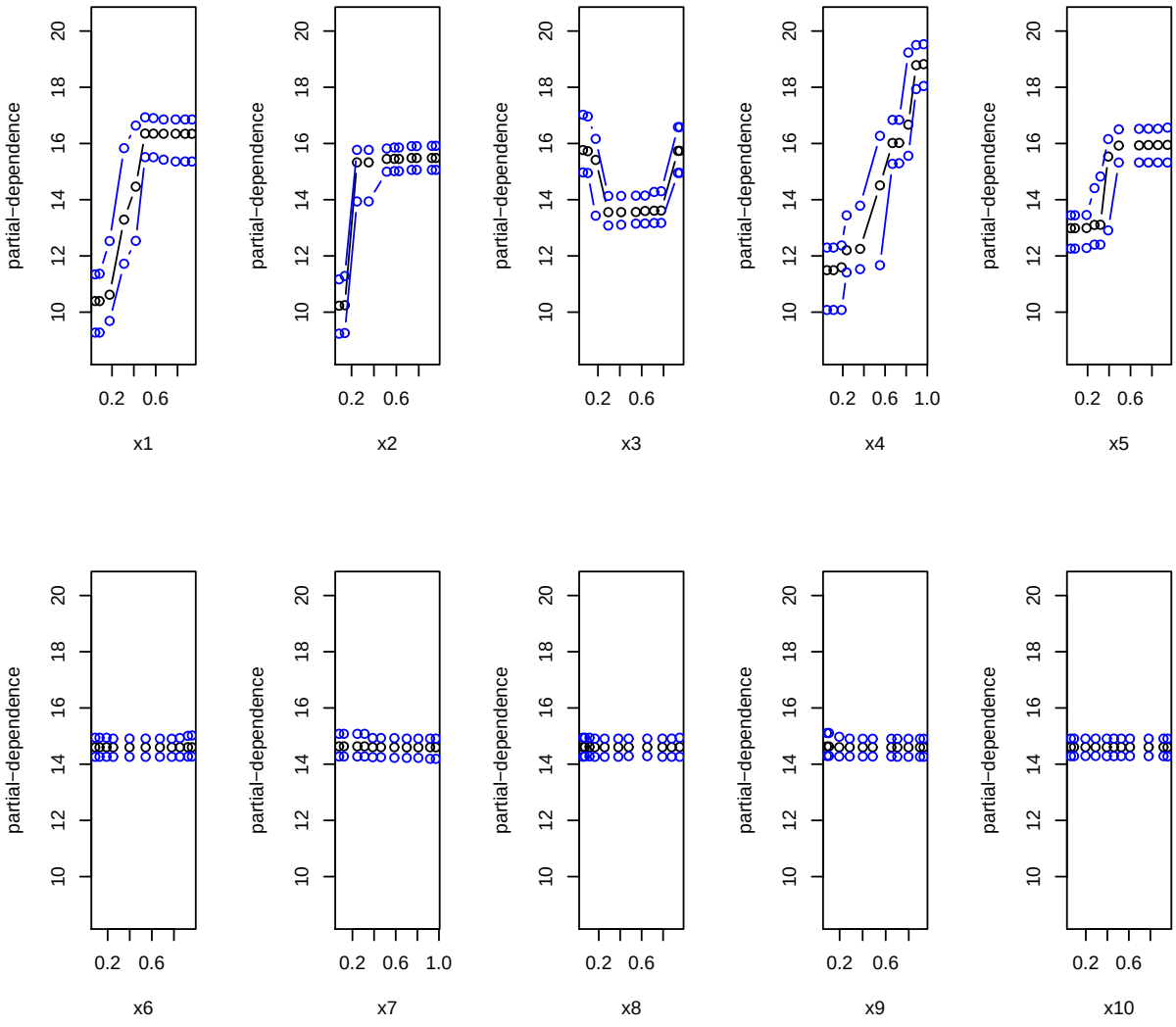

Figure 4: Partial dependence plots for the 10 predictors in the Friedman data.

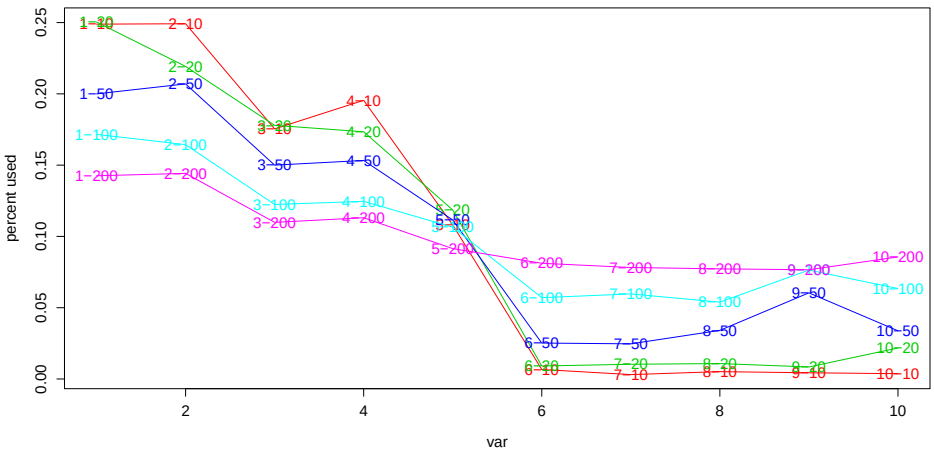

Figure 5: Average use per splitting rule for $x_1, \ldots, x_{10}$. Notation: 2-20 means x_2 with $m = 20$ trees, etc.

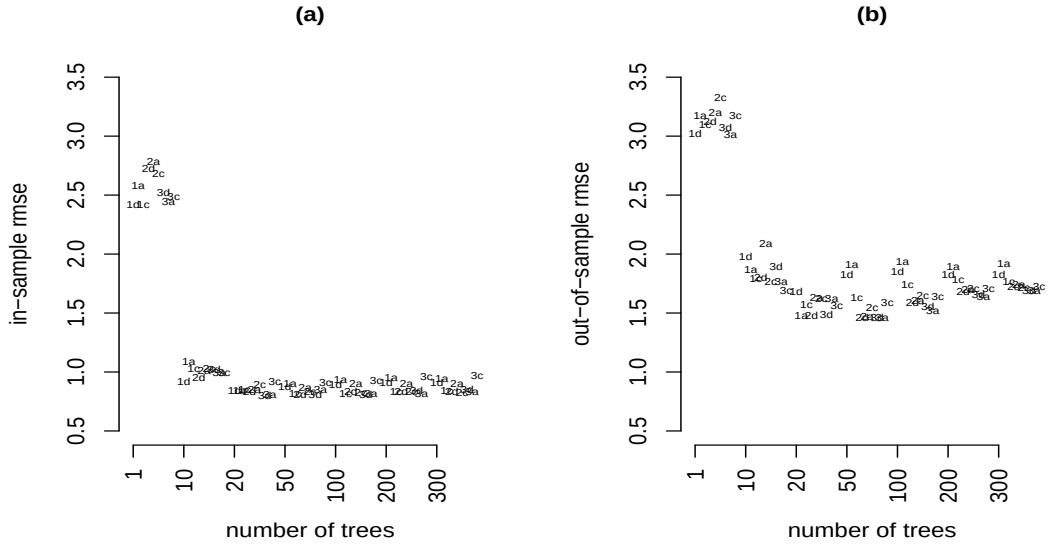

Figure 6: BART's robust RMSE performance as (ν, q, k, m) is varied: (a) in-sample RMSE comparisons and (b) out-of-sample RMSE comparisons.

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.

5.2.2 Out-of-Sample Comparisons with Competing Methods

To gauge how well BART performs on the Friedman setup, we compared the out-of-sample performance of BART-cv and BART-default with the four competing methods considered in Section 5.1, the Lasso, gradient boosting, neural nets and random forests. We also considered MARS (Friedman (1991), implemented as `polymars` in R by Kooperberg, Bose & Stone (1997)).

The methods were compared using 50 independent replications of the simulation used in Section 5.2.1, namely 100 independent values of (x, y) from (26) and (27) with $p = 10$. Each method was then trained on these 100 in-sample values to obtain an estimate $\hat{f}$ of f using (18). With the exception of BART-default, 5-fold cross-validation was used to select from the operational parameter values listed in Table 1 or for MARS, to select a value of its `gcv` parameter from 1,...,8. We then simulated an additional 1000 out-of-sample x values from (26). For each of our 50 training samples, the performance of each method was then evaluated by the RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{f}(x_i) - f(x_i))^2}$ over the $n = 1000$ out-of-sample values.

Average RMSEs over 50 replicates and standard errors of averages are given in Table 3. All the methods explained substantial variation, as the average RMSE for the constant model ($\hat{y} \equiv \bar{y}$) is 4.87. However, both BART-cv and BART-default substantially outperformed all the other methods by a significant amount. Surprisingly, BART-default performed here even better than BART-cv. Confirming what we saw in Section 5.1, 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).

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

5.2.3 Finding Low Dimensional Structure in High Dimensional Data

Of the p variables $x_1, \ldots, x_p$ from (26), f in (27) 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 a 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 3 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 7 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

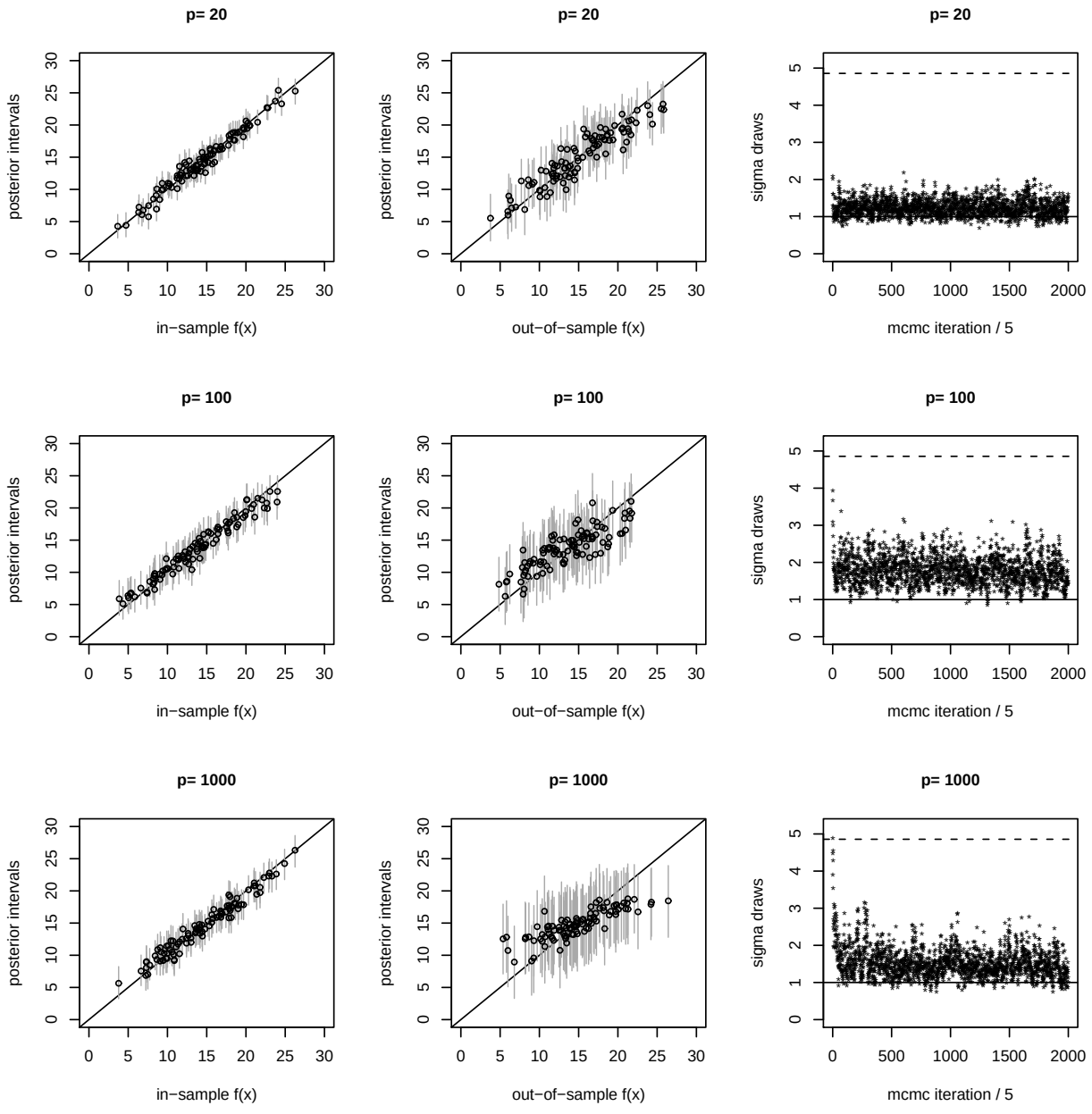

Figure 7: Inference about Friedman's function in p dimensions.

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, when the dimension p is so large compared to the sample size $n = 100$, it is remarkable that the BART inferences are at all reliable, at least in the middle of the data.

In the third column of Figure 7, 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. Lastly, note that the sequence of σ draws in Figure 7 are systematically higher than the σ draws in Figure 3(c). This may be due in part to the fact that the regression $\hat{\sigma}$ rather than the naive $\hat{\sigma}$ was used to anchor the prior in Figure 3. Indeed if the naive $\hat{\sigma}$ was instead used for Figure 3, 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 (26) 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.

5.3 Classification: A Drug Discovery Application

Our last example illustrates an application of the BART probit approach described in Section 4 to a drug discovery classification problem. In such problems, the goal is to predict the "activity" of a compound against a biological target, using predictor variables that characterize the molecular structure of the compound. By "activity", one typically means the ability to effect a desired outcome against some biological target, such as inhibiting or killing a certain virus.

The data we consider describe $p = 266$ molecular characteristics of $n = 29,374$ compounds, 542 of which were classified as active. These predictors represent topological aspects of molecular structure. This data set was collected by the National Cancer Institute, and is described in Feng, Lurati, Ouyang, Robinson, Wang, Yuan & Young (2003). For the in-sample and out-of-sample comparisons below, we randomly split the data into nonoverlapping train and test sets, each with 14687 compounds of which 271 were active.

Designating the activity of a compound by a binary variable ($Y = 1$ if active and $Y = 0$ otherwise), we applied BART probit to the train set to obtain posterior mean estimates of the conditional probabilities, $P[Y = 1 \,|\, x]$, for each x vector of the 266 molecular predictor values. For this purpose we used BART probit with the default $k = 2$ and $m = 50$ trees (ν and q have no meaning for the probit model, since the latent response is assumed to have variance 1). To gauge MCMC convergence, we performed four independent repetitions of 250,000 MCMC iterations and obtained essentially the same results each time.

To get a feel for the extent to which BART's posterior mean estimates of $P[Y = 1 \,|\, x]$ can be used to identify promising drugs, Figure 8 plots the 20 largest $P[Y = 1 \,|\, x]$ estimates for the train and the test sets. Also provided are the 90% posterior intervals which convey uncertainty and the identification whether the drug was in fact active ($y = 1$) or not ($y = 0$). As there are four inactives in each plot, the true positive rates in both the train and test sets for these 20 largest

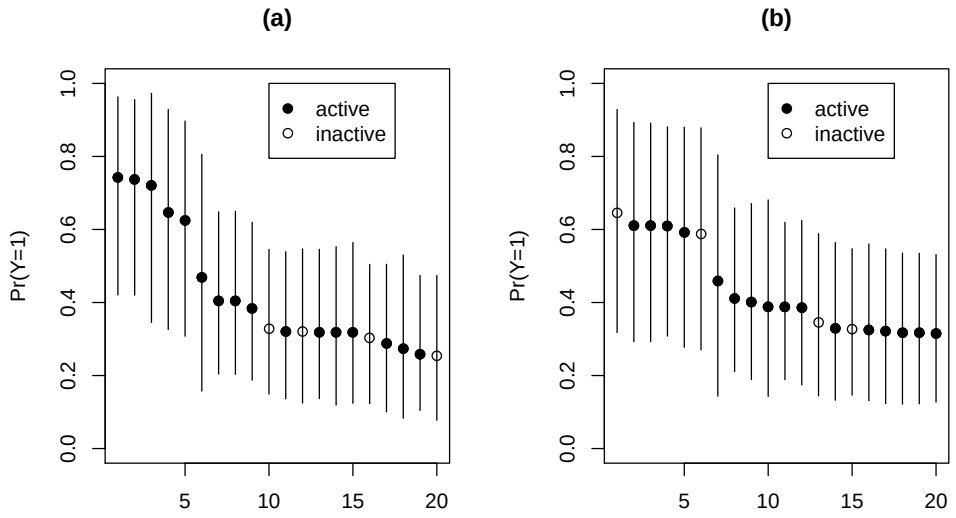

Figure 8: BART posterior intervals for the 20 compounds with highest predicted activity, using train (a) and test (b) sets.

estimates is $16/20 = 80\%$, an impressive gain over the $271/14687 = 1.85\%$ base rate. It may be of interest to also note that the test set intervals are slightly wider, with an average width of 0.50 compared to 0.47 for the training intervals.

Going beyond the performance of the top 20 estimates, we assess overall out-of-sample predictive performance with Receiver Operating Characteristic (ROC) curves based on the orderings of the obtained test set $P[Y = 1 \,|\, x]$ estimates. As a frame of reference for comparison, we also applied BART probit with $m = 1$ (which we denote by BCART because it is essentially a version of Bayesian CART), and a greedy nonBayesian recursive partitioning method implemented as the `rpart` package in R. For RPART, a large tree (around 80 terminal nodes) was grown without pruning, as the superior performance of large trees over pruned trees on this problem has been previously documented Wang (2005).

The ROC curves for BART, BCART and RPART, calculated with the `ROCR`

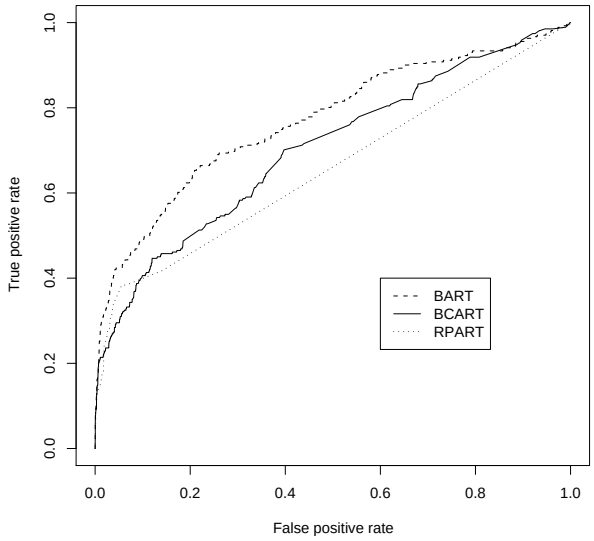

Figure 9: Test set ROC curves for BART, BCART, and RPART.

package in R and displayed in Figure 9, plot the tradeoffs between true positive and false positive rates across thresholds for each method. The AUC (area under the curve) values for the BART, BCART and RPART curves here are 0.77, 0.70 and 0.66, respectively. Noting that higher curves indicate better performance, and that a classifier's AUC value is the probability that it will rank a randomly chosen $y = 1$ example higher than a randomly chosen $y = 0$ example (Bamber 1975), we see that BART has provided more reliable predictions than BCART which in turn has provided more reliable predictions than RPART. We note in passing that all three methods are substantially better than random guessing which corresponds to a 45-degree line ROC curve with an AUC of 0.50.

Finally, we turn to the issue of variable selection and demonstrate that by decreasing the number of trees m, BART probit can be used, just as BART in Section 5.3.1, to identify those predictors which have the most influence on the response. For this purpose, we modify the data setup as follows: instead of hold-

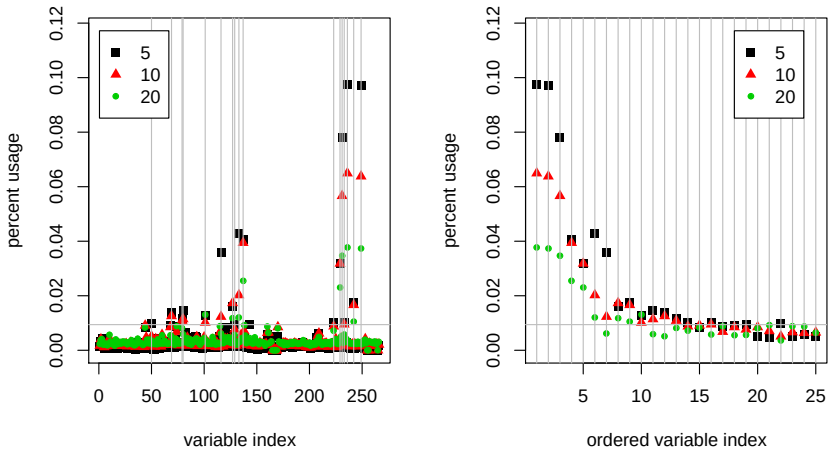

Figure 10: Variable importance measure, drug discovery example. Values are given for 5, 10 and 20 trees in the ensemble, for all 266 variables (a) and the 25 variables with the highest mean usage (b). Vertical lines in (a) indicate variables whose percent usage exceeds the 95th percentile. The 95th percentile is indicated by a horizontal line.

ing out a test set, all 542 active compounds and a subsample of 542 inactives were used to build a model. Four independent chains, each with 1,000,000 iterations, were used. The large number of iterations was used to ensure stability in the "percent usage" variable selection index (20). Ensembles with $5, 10$, and 20 trees were considered.

The results are displayed in Figure 10. The same three variables are selected as most important for all three ensembles. Considering that $1/266 \approx 0.004$, percent usages of 0.050 to 0.100 are quite a bit larger than one would expect if all variables were equally important. As expected, variable usage is most concentrated in the case of a small ensemble (i.e., $m = 5$ trees).

6 Extensions and Related Work

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. For instance, one might consider a model of the form

$$Y = h_1(x) + h_2(z) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2) \tag{28}$$

where $h_1(x)$ is a sum of trees as in (2) and $h_2(z)$ is a parametric form involving z, a second vector of predictors. One can also extend the sum-of-trees model to a multivariate framework such as

$$Y_i = h_i(x_i) + \epsilon_i, \quad (\epsilon_1, \epsilon_2, \ldots, \epsilon_p) \sim N(0, \Sigma), \tag{29}$$

where each h_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 3.1 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 (29) is easily fit by drawing each h_i^* given $\{h_j^*\}_{j \neq i}$ and Σ, and then drawing Σ given all the h_i^*.

An early version of our work on BART (Chipman, George & McCulloch 2007) was published in the proceedings of the conference Advances in Neural Information Processing Systems 2006. Based on this and other preliminary technical reports of ours, a variety of extensions and applications of BART have begun to appear. Zhang, Shih & Muller (2007) proposed SBART an extension of BART obtained by adding a spatial component along the lines of (28). Applied to the problem of merging data sets, they found that SBART improved over the conventional census based method. For the predictive modeling problem of TF-DNA

binding in genetics, Liu & Zhou (2007) and Zhou & Liu (2007) considered a variety of learning methods, including stepwise linear regression, MARS, neural networks, support vector machines, boosting and BART. Concluding that "the BART method performed best in all cases", they noted BART's "high predictive power, its explicit quantification of uncertainty and its interpretability". By keeping track of the per sample inclusion rates, they successfully used BART to identify some unusual predictors. Zhang & Haerdle (2007) independently discovered the probit extension of BART, which they call BACT, and applied it to credit risk data to predict the insolvency of firms. They found BACT to outperform the logit model, CART and support vector machines. Abu-Nimeh, Nappa, Wang & Nair (2008) also independently discovered the probit extension of BART , which they call CBART, and applied it for the automatic detection phishing emails. They found CBART to outperform logistic regression, random forests, support vector machines, CART, neural networks and the original BART. Abreveya & McCulloch (2006) applied BART to hockey game penalty data and found evidence of referee bias in officiating. Finally, Hill & McCulloch (2007) apply BART to a counterfactual causality analysis for assessing the effectiveness of a job training program based on observational data. They found that BART produced more efficient estimates than propensity score matching, propensity-weighted estimators and regression adjustment. Without exception, these papers provide further evidence for the remarkable potential of BART.

7 Discussion

The essential components of BART are the sum-of-trees model, the regularization prior and the backfitting MCMC algorithm. As opposed to the Bayesian approaches of CGM98 and Denison et al. (1998), where a single tree is used to explain all the variation in y, each of the trees in BART accounts for only part

of the overall fit. This is accomplished with a regularization prior that shrinks the tree effects towards a simpler fit. To facilitate the implementation of BART, the prior is formulated in terms of rapidly computable forms that are controlled by interpretable hyperparameters, and which allow for a highly effective default version for immediate "off-the-shelf" use. Posterior calculation is carried out by a tailored backfitting MCMC algorithm that appears to converge quickly, effectively obtaining a (dependent) sample from the posterior distribution over the space of sum-of-trees models. A variety of inferential quantities of interest can be obtained directly from this sample.

The application of BART to a wide variety of data sets and a simulation experiment (Section 5) served to demonstrate many of its appealing features. In terms of out-of sample predictive RMSE performance, BART compared favorably with boosting, the lasso, MARS neural nets and random forests. In particular, the computationally inexpensive and easy to use default version of BART performed extremely well. In the simulation experiments, BART obtained reliable posterior mean and interval estimates of the true regression function as well as the marginal predictor effects. BART's performance was seen to be remarkably robust to hyperparameter specification, and remained effective when the regression function was buried in ever higher dimensional spaces. BART was also seen to be a new effective tool for model-free variable selection. Finally, a straightforward probit extension of BART for classification of binary Y was seen to be an effective tool for discovering promising drugs on the basis of their molecular structure.

References

Abreveya, J. & McCulloch, R. (2006), Reversal of fortune: a statistical analysis of penalty calls in the national hockey league, Technical report, Purdue University.

Abu-Nimeh, S., Nappa, D., Wang, X. & Nair, S. (2008), Detecting phishing emails via Bayesian additive regression trees, Technical report, Southern Methodist University.

Albert, J. H. & Chib, S. (1993), 'Bayesian analysis of binary and polychotomous response data', *Journal of the American Statistical Association* **88**, 669–679.

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.

Chipman, H. A., George, E. I. & McCulloch, R. E. (2007), Bayesian ensemble learning, *in* 'Neural Information Processing Systems 19'.

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

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004), 'Least angle regression', *Annals of Statistics* **32**, 407–499.

Feng, J., Lurati, L., Ouyang, H., Robinson, T., Wang, Y., Yuan, S. & Young, S. (2003), 'Predictive toxicology: Benchmarking molecular descriptors and statistical methods', *Journal of Chemical Information and Computer Sciences* **43**(5), 1463–1470.

Freund, Y. & Schapire, R. E. (1997), 'A decision-theoretic generalization of online 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.

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

Hill, J. L. & McCulloch, R. E. (2007), 'Bayesian nonparametric modeling for causal inference', *Journal of the American Statistical Association*. In press.

Kim, H., Loh, W.-Y., Shih, Y.-S. & Chaudhuri, P. (2007), 'Visualizable and interpretable regression models with good prediction power', *IEEE Transactions: Special Issue on Data Mining and Web Mining*. In press.

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

Liu, J. S. & Zhou, Q. (2007), 'Predictive modeling approaches for studying protein-DNA binding', *ICCM*.

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.

Wang, Y. (2005), 'Statistical methods for high-throughput screening drug discovery data'.

Wu, Y., Tjelmeland, H. & West, M. (2007), 'Bayesian CART: Prior specification and posterior simulation', *Journal of Computational and Graphical Statistics*. In press.

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.

Zhang, J. L. & Haerdle, W. K. (2007), The Bayesian additive classification tree applied to credit risk modelling, Technical report, Humboldt-Universitat zu Berlin.

Zhang, S., Shih, Y.-C. T. & Muller, P. (2007), 'A spatially-adjusted Bayesian additive regression tree model to merge two datasets', *Bayesian Analysis* **2**, 611–634.

Zhou, Q. & Liu, J. S. (2007), Extracting sequence features to predict protein-DNA binding: A comparative study, Technical report, Harvard University.