Communications for Statistical Applications and Methods
2017, Vol. 24, No. 6, 543–559

https://doi.org/10.29220/CSAM.2017.24.6.543
Print ISSN 2287-7843 / Online ISSN 2383-4757

A review of tree-based Bayesian methods

Antonio R. Linero[1,a]

[a]Department of Statistics, Florida State University, USA

Abstract

Tree-based regression and classification ensembles form a standard part of the data-science toolkit. Many commonly used methods take an algorithmic view, proposing greedy methods for constructing decision trees; examples include the classification and regression trees algorithm, boosted decision trees, and random forests. Recent history has seen a surge of interest in Bayesian techniques for constructing decision tree ensembles, with these methods frequently outperforming their algorithmic counterparts. The goal of this article is to survey the landscape surrounding Bayesian decision tree methods, and to discuss recent modeling and computational developments. We provide connections between Bayesian tree-based methods and existing machine learning techniques, and outline several recent theoretical developments establishing frequentist consistency and rates of convergence for the posterior distribution. The methodology we present is applicable for a wide variety of statistical tasks including regression, classification, modeling of count data, and many others. We illustrate the methodology on both simulated and real datasets.

Keywords: Bayesian additive regression trees, boosting, random forests, semiparametric Bayes

1. Introduction

Tree-based regression and classification ensembles form a standard part of the data-science toolkit. Consider the generic problem of estimating the distribution $f(y \quad x)$ of some response $Y \top \mathcal{Z}$ conditional on a predictor $X \top \mathcal{Y}$. Tree-based methods recursively partition $\mathcal{Y}$ to obtain a decision tree $\mathcal{X}$ such that the information in X about Y is contained entirely in which leaf node X falls in; this process is illustrated in Figure 1 when $X \top [0,1]^2$. Once the tree is constructed, we associate to each leaf node η a parameter θ_η and set $f(y \quad x) = f(y \quad \theta_{\eta(x)})$ where $\eta(x)$ denotes the leaf node associated to x. For example, we might obtain a semiparametric regression model by setting $Y \approx \text{Normal}(\mu_{\eta(X)}, \sigma^2_{\eta(X)})$.

Decision trees have attracted the interest of practitioners due to their strong empirical performance, and many algorithmic methods for constructing decision trees exist (Breiman *et al.*, 1984; Kass, 1980; Quinlan, 1993). An essential fact about decision trees is that a given dataset can typically be described well by many different tree structures. This causes greedily-constructed trees to be very unstable, with small perturbations of trees leading to vastly different tree topologies. This motivates methods based on *ensembles* of decision trees, such as bagging (Breiman, 1996) and random forests (Breiman, 2001), which increase estimation stability by averaging predictions over the bootstrap distribution of the decision tree (Efron and Gong, 1983).

Bayesian approaches fundamentally differ from these algorithmic approaches in that they posit a full probability model for the data, combining a prior distribution $\pi(\mathcal{X})$ for $\mathcal{X}$ with a likelihood $m(\mathcal{I} \quad \mathcal{X})$ for the data $\mathcal{I} = \}(X_i, Y_i) : 1 \geq i \geq N|$. This approach falls in the framework of Bayesian

[1] Department of Statistics, Florida State University, 600 W. College Avenue, Tallahassee, FL 32306 USA.
E-mail: arlinero@stat.fsu.edu

Published 30 November 2017 / journal homepage: http://csam.or.kr

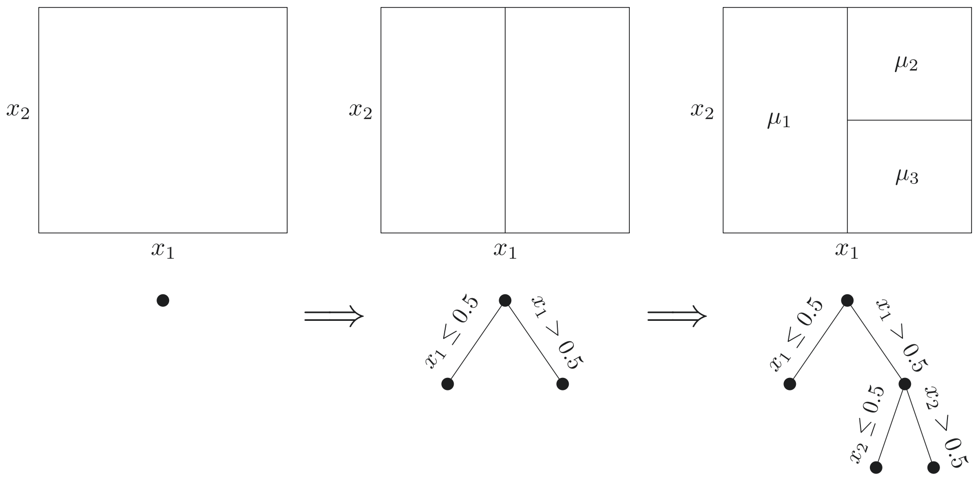

Figure 1: *Schematic depicting the construction of a decision tree (bottom) and the induced recursive partition associated with the construction of the tree (top). After the decision tree is constructed, parameters associated to leaf node η is given a mean parameter μ_ℓ.*

nonparametrics/semiparameterics (see MacEachern, 2016 for a review) This viewpoint allows for natural uncertainty quantification using the posterior distribution $\pi(\mathcal{X} \quad \mathcal{I})$. Furthermore, with no additional effort, the Bayesian approach naturally has many of the features of state-of-the-art tree ensembling methods. For example, the process of averaging predictions across samples from the posterior distribution is similar to the averaging which occurs in the random forests algorithm, naturally leading to stability in inferences.

Bayesian decision tree methods have experienced rapid development in recent years, and the purpose of this article is to offer a timely review. We first provide a basic review of the Bayesian classification and regression trees (BCART) and Bayesian additive regression trees (BART) methods, and provide an overview of the techniques used to fit them. We then make connections to other machine learning techniques, including random forests, boosting, and Gaussian process regression. Additionally, several recent works have provided theoretical justification for these methods by establishing optimal posterior convergence rates for variants of BCART (Rockova and van der Pas, 2017) and BART (Linero and Yang, 2017; Rockova and van der Pas, 2017); we review some of these developments.

2. Bayesian classification and regression trees

Our notation follows Chipman *et al.* (2013), who also provide a review of tree-based methods. Associated to a tree $\mathcal{X} = \}\mathcal{L}_\mathcal{X}, \mathcal{T}_\mathcal{X}, \mathcal{N}_\mathcal{X}|$ we have a collection of internal (branch) nodes $\mathcal{L}_\mathcal{X}$ with decision rules $\mathcal{T}_\mathcal{X}$ and leaf nodes $\mathcal{N}_\mathcal{X}$. We let $\eta(x) \top \mathcal{N}_\mathcal{X}$ denote the unique leaf node associated to x. Associated to each $\eta \top \mathcal{N}_\mathcal{X}$ is a parameter $\theta_\eta \top \Theta_\mathcal{X}$ where $\Theta_\mathcal{X} = \}\theta_\eta : \eta \top \mathcal{N}_\mathcal{X}|$.

The BCART was introduced by Chipman *et al.* (1998) and Denison *et al.* (1998), and provides the basis for most of the subsequent developments. To simulate from the BCART prior, one first draws a tree structure $\mathcal{X} \approx \pi(\mathcal{X})$ from a prior $\pi(\mathcal{X})$ (see Section 2.1) and next samples, for each leaf $\eta \top \mathcal{N}_\mathcal{X}$, a parameter vector θ_η. The response is then modeled as $(Y_i \quad X_i = x, \mathcal{X}, \Theta_\mathcal{X}) \approx f(y \quad \theta_{\eta(x)})$ for some family of densities $f(y \quad \theta)$.

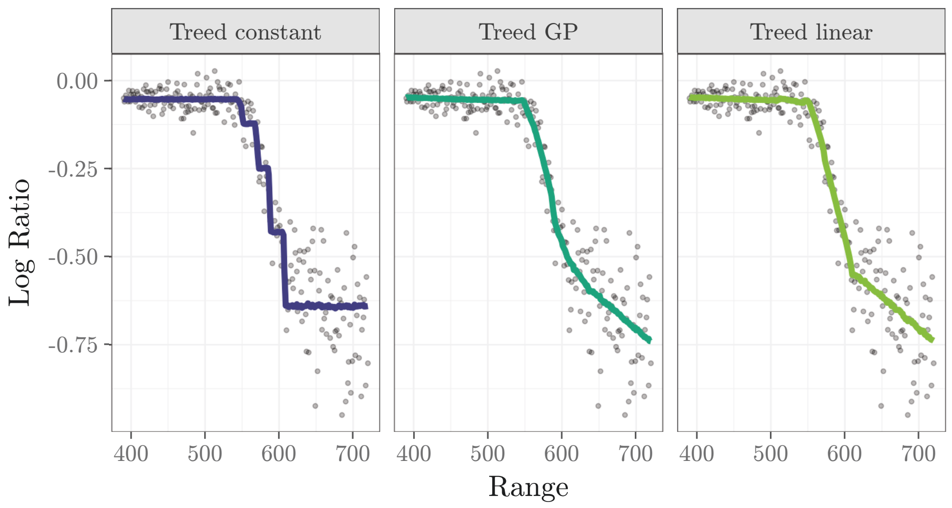

Figure 2: *Fits of the Bayesian classification and regression trees model in which the within-leaf response mean is (left) a constant μ_η, (middle) a Gaussian process $\mu_\eta(x)$, and (right) a linear model $\mu_\eta(x) = \alpha_\eta + x^{\mathcal{D}}\beta_\eta$.*

Example 1. [Semiparametric regression] Suppose Y_i is continuous. We specify the model

$$Y_i = \mu(X_i) + \epsilon_i, \quad \epsilon_i \approx \text{Normal}\Big)0, \sigma(X_i)^2\Big(, \tag{2.1}$$

where $\mu(x) = \mu_{\eta(x)}$ and $\sigma^2(x) = \sigma^2_{\eta(x)}$. Additional flexibility can be obtained by considering a locally linear model $Y_i = \alpha(X_i) + \beta(X_i)^{\mathcal{D}} X_i + \epsilon_i$; similarly, Gramacy and Lee (2008) allow for the within-leaf models to be Gaussian process regressions (Rasmussen and Williams, 2006). The fits of these models to the `lidar` data described in Section 4.1 are given in Figure 2.

Example 2. [Poisson regression for count data] Suppose Y_i is a count. As an alternative to a Poisson log-linear model, we set $Y_i \approx \text{Poisson}(\lambda_{\eta(X_i)})$. For an application of this type of model to overdispersed count data, see Section 4.2.

Example 3. [Survival analysis with non-proportional hazards] Suppose Y_i is a survival time, potentially subject to right-censoring. We set $Y_i \approx S_{\eta(X_i)}(t)$ where $S_\eta(t)$ is a nonparametrically modeled survival function given, e.g., a Pölya tree prior (Muliere and Walker, 1997).

The above approach extends naturally to other semiparametric models; for example, one obtains a natural extension to a generic exponential family $f(y \quad \theta) = \exp\}y\theta \quad B(\theta) + C(y)|$. The computational algorithms described in Section 3 are applicable whenever θ_η is given a prior for which the marginal likeihood $m(\mathcal{I} \quad \mathcal{X}) = \int_{\eta \top \mathcal{N}_X} \sum\int_{i:\eta(X_i)=\eta} f(Y_i \quad \theta)\pi_\theta(\theta_\eta)\, d\theta_\eta$ can be efficiently computed. Chipman *et al.* (1998) and Denison *et al.* (1998) consider, in particular, the classification problem in which $f(y \quad x)$ is categorical with $Y \approx \text{Categorical}(\theta_{\eta(X)})$ where $\theta_\eta \approx \text{Dirichlet}(\alpha_1, \ldots, \alpha_C)$ is a leaf-specific distribution over categories. These extensions can be viewed as semiparametric competitors to classical generalized linear models (McCullagh, 1984) or generalized additive models (Hastie and Tibshirani, 1987). An attractive feature of these models is that they naturally allow for the inclusion of complex interactions between predictors.

2.1. Priors on tree structures

We consider priors on $(\mathcal{X}, \Theta_\mathcal{X})$ of the form

$$\pi(\mathcal{X}, \Theta_\mathcal{X}) = \pi_\mathcal{X}(\mathcal{X}) \bigcirc\int_{\eta \int \mathcal{N}_\mathcal{X}} \pi_\theta(\theta_\eta). \tag{2.2}$$

That is, the θ_η's are conditionally independent given the tree structure $\mathcal{X}$.

We require a prior distribution $\pi_\mathcal{X}(\mathcal{X})$ for the tree structure. A common choice is to use a branching process prior, described initially by Chipman *et al.* (1998). For each node η in the tree, we split η with prior probability $q(\eta) = \gamma(1 + D(\eta))^{\ \beta}$, where $D(\eta)$ denotes the depth of η (where the root of the tree has depth $D(\eta) = 0$). The parameters (γ, β) are parameters which control the shape of the trees. The parameter $\gamma > 0$ controls the prior probability that the tree is empty (and hence is typically large), while $\beta > 0$ penalizes trees which are too deep.

Once the tree topology is generated, we associate to each branch $\eta \top \mathcal{L}_\mathcal{X}$ a *splitting rule* of the form $[x_{j(\eta)} \geq C_\eta]$. We set $j(\eta) \overset{\text{indep}}{\approx}$ Categorical(s) where $s = (s_1, \dots, s_P)$ is a probability vector and P is the dimension of the prediction X_i; most implementations set $s_j = P^{\ 1}$, although this choice is not ideal when the predictors do not have equal importance (Linero, 2016). Conditional on $j(\eta) = j$, we take $C_\eta \approx G_{\mathcal{X},\eta}$ where $G_{\mathcal{X},\eta}$ is a distribution G_j restricted to the set of values which do not lead to logically empty nodes when variable j is split on (if no such split exists, another predictor is selected according to s). In most publicly available software G_j is the empirical distribution of $(X_{ij} : 1 \geq i \geq N)$.

There are several alternatives in the literature to the prior described above, differing in how the topology of $\mathcal{X}$ is generated. Denison *et al.* (1998) suggest a hierarchical prior in which a prior is placed on the number of leaf nodes $L = \ \mathcal{N}_\mathcal{X} \ \approx 1 + \text{Poisson}(\lambda)$ and, conditional on L, a uniform distribution is placed on the tree topologies with L leaves. Wu *et al.* (2007) construct a "pinball" prior on the topology of $\mathcal{X}$, which gives finer control over the shape of the resulting trees. Finally, Roy and Teh (2009) introduced the *Mondrian process*, which gives an elegant joint prior on all components of $\mathcal{X}$ (more precisely, the Mondrian process is a stochastic process $\}\mathcal{X}_t : t > 0|$, with larger values of t corresponding to finer partitions). Lakshminarayanan *et al.* (2014) leverage the self-consistency properties of the Mondrian process to construct probabilistic alternatives random forests which allow online implementations.

2.2. Bayesian additive regression trees

The BART framework introduced by Chipman *et al.* (2010) replaces the single $\mathcal{X}$ with a sum of trees

$$\mu(x) = \prod_{t=1}^{T} g(x; \mathcal{X}_t, \Theta_t), \tag{2.3}$$

where $g(x; \mathcal{X}_t, \Theta_t) = \mu_{\eta(t,x)}$, $\eta(t, x) = \eta$ if x is associated to leaf node η in tree $\mathcal{X}_t$, and $\Theta_t = \}\mu_\eta : \eta \top \mathcal{N}_{\mathcal{X}_t}|$. The tree structures $\mathcal{X}_t$ are then given independent priors as described in Section 2.1. The leaf parameters μ_η are given iid Normal$(0, \sigma^2_\mu/T)$ priors, with the scaling factor T chosen so that Var$\}\mu(x)| = \sigma^2_\mu$ irrespective of T. The BART model was initially motivated by analogy with boosting algorithms (Freund *et al.*, 1999), which combine many so-called "weak learners" to construct predictions; the weak learners of choice in many boosting applications are shallow decision trees. By taking β large in the branching process prior, the BART model represents a Bayesian version of this idea with $\mu(x)$ represented as a sum of shallow decision trees.

The BART algorithm applies most naturally to the semiparametric regression problem; repeated experience has shown that it typically outperforms BCART in this setting, and is typically competitive with state-of-the-art methods. Additionally, it is straight-forward to extend BART to classification using the data augmentation strategy of Albert and Chib (1993). Much further work has gone into extending BART to various other problems, including survival analysis (Sparapani *et al.*, 2016) and as an alternative to loglinear models (Murray, 2017). The strong predictive performance of BART has also lead to it being widely adopted in estimating average causal effects in causal inference problems (Hill, 2011).

BCART is more flexible as a framework than BART in the sense that it generalizes more easily to non-regression settings. Whereas BCART allows for essentially any type of object in the leaf nodes (scalars, probability vectors, survival functions, etc.), one must be able to add together the parameters in the leaf nodes of BART, limiting us to leaf parameters in $\mathbb{R}$ (or possibly $\mathbb{R}^D$ for some D). As suggested in Section 5.2, the BART model is perhaps best thought of as a high-quality drop-in replacement for Gaussian process priors (see Rasmussen and Williams, 2006, for a textbook level treatment).

Because the estimates of BART correspond to sums of step functions, BART produces estimates which are not smooth. To address this issue, Linero and Yang (2017) proposed *smoothed* Bayesian additive regression trees (SBART) which use so-called "soft" decision trees (Irsoy *et al.*, 2012) in constructing the ensemble; rather than going left or right down the tree, the tree assigns differing weights to the left and right paths according to how close the observation is to the cutpoint. As we show in illustrations here, this additional smoothing usually produces better estimates when the underlying regression function is smooth. The cost of this additional flexibility is an increase in computational effort.

2.3. Variable selection and automatic relevance determination

Bayesian tree-based models ostensibly conduct model selection through which variables are used to construct splits. As noted by Linero (2016), however, Bayesian tree-based methods are not immediately applicable for variable selection due to the tendency for spurious predictors to be selected once the number of branches becomes sufficiently large. This issue is most obvious in the case of BART, which naturally produces ensembles with many branches, but is also expected to occur with BCART when sufficiently deep trees are required. One attractive option to conduct variable selection is to place a sparsity-inducing Dirichlet prior on the vector of splitting proportions

$$s \approx \text{Dirichlet}\left)\frac{\alpha}{P}, \dots, \frac{\alpha}{P}\right\{. \tag{2.4}$$

Linero (2016) shows that, given that the ensemble includes B branch nodes, the number of predictors Q included in the model has an approximate $1 + \text{Poisson}(\theta)$ distribution, where $\theta = \prod_{i=1}^{B} \alpha/(\alpha + i)$. In addition to providing variable selection, the Dirichlet prior also functions as an *automatic relevance determination* prior (Neal, 1995), allowing the model to adaptively learn the relative importance of each predictor.

3. Computational details

Constructing efficient algorithms for exploring the posterior $\pi(\mathcal{X} \quad \mathcal{I})$ is perhaps the biggest standing problem in the widespread adoption of Bayesian methods for decision trees. While Markov chain Monte Carlo (MCMC) is the frequently used, we show in Section 4 that the widely available implementations of BCART generally perform poorly compared to recent particle filtering algorithms

(Lakshminarayanan *et al.*, 2013; Taddy *et al.*, 2011). We review the MCMC methods used to fit BCART before discussing more recent methods based on sequential Monte Carlo. We then discuss approaches for fitting BART.

3.1. Markov chain Monte Carlo for Bayesian classification and regression trees

Consider first BCART under a prior respecting (2.2). The posterior $\pi_{\mathcal{X}}(\mathcal{X} \quad \mathcal{I})$ can be written

$$\pi_{\mathcal{X}}(\mathcal{X} \quad \mathcal{I}) = \frac{\pi_{\mathcal{X}}(\mathcal{X})}{\{\ (\mathcal{I})} \int_{\eta \top \mathcal{N}} \int_{i:\eta(X_i)=\eta} f(Y_i \quad \theta_\eta)\, d\pi_\theta(\theta_\eta) = \frac{\pi_{\mathcal{X}}(\mathcal{X}) m(\mathcal{I} \quad \mathcal{X})}{\{\ (\mathcal{I})}.$$

Efficient inference requires that the marginal likelihood $m(\mathcal{I} \quad \mathcal{X})$ can be computed efficiently, while the normalizing constant $\{\ (\mathcal{I})$ is not required for further computations.

Example 4. [Poisson regression] Suppose $(Y_i \quad X_i = x, \mathcal{X}, \Theta_{\mathcal{X}}) \approx \text{Poisson}(\lambda_{\eta(x)})$. Suppose that $(\lambda_\eta \quad \mathcal{X}) \approx \text{Gam}(a, b)$. Then

$$m(\mathcal{I} \quad \mathcal{X}) = \frac{b^{aL}}{\Gamma(a)^L \int_{i=1}^{N} Y_i!} \int_{\eta \mathcal{J} \mathcal{N}_{\mathcal{X}}} \frac{\Gamma(a + S_\eta)}{(b + n_\eta)^{a+S_\eta}},$$

where $L = \ \mathcal{N}$, $n_\eta = \ \}i : \eta(X_i) = \eta|$, and $S_\eta = \prod_{i:\eta(X_i)=\eta} Y_i$.

Example 5. [Gaussian homoskedastic regression] Suppose $Y_i = \mu(X_i) + \epsilon_i$ where $\mu(x) = \mu_{\eta(x)}$ and $\epsilon_i \overset{\text{iid}}{\approx} \text{Normal}(0, \sigma^2)$. Suppose that $(\mu_\eta \quad \mathcal{X}) \overset{\text{iid}}{\approx} \text{Normal}(m, \sigma_\mu^2)$. Then

$$m(\mathcal{I} \quad \mathcal{X}) = \int_{\eta \mathcal{J} \mathcal{N}_{\mathcal{X}}} \big)2\pi\sigma^2\big(\ ^{\frac{n_\eta}{2}} \overline{\frac{\sigma^2}{\sigma^2 + n_\eta \sigma_\mu^2}} \exp \Big\} \ \frac{\text{SSE}(\eta)}{2\sigma^2} \quad \frac{1}{2} \frac{n_\eta \big)\bar{Y}_\eta \quad m\big(^2}{n_\eta \sigma_\mu^2 + \sigma^2} \Big\{,$$

where, using the same notation as in the Poisson setting, $\bar{Y}_\eta = S_\eta / n_\eta$ and $\text{SSE}(\eta) = \prod_{i:\eta(X_i)=\eta} (Y_i \quad \bar{Y}_\eta)^2$.

With the marginal likelihood $m(\mathcal{I} \quad \mathcal{X})$ in hand, one then typically samples approximately from $\pi(\mathcal{X} \quad \mathcal{I})$ using the Metropolis-Hastings algorithm (Hastings, 1970), which uses a *proposal distribution* $q(\mathcal{X}^{(m)} \propto \ \mathcal{X}^{\infty})$ to construct a Markov chain $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots$ with invariant distribution $\pi(\mathcal{X} \quad \mathcal{I})$. If $\mathcal{X}^{(m)}$ is the state of $\mathcal{X}$ at iteration m of the Markov chain we (i) sample $\mathcal{X}^{\infty} \approx q(\mathcal{X} \propto \ \mathcal{X}^{\infty})$ and (ii) set $\mathcal{X}^{(m+1)} = \mathcal{X}^{\infty}$ with probability

$$a(\mathcal{X}^{(m)} \propto \ \mathcal{X}^{\infty}) = \min \Big\} \frac{\pi(\mathcal{X}^{\infty} \ \mathcal{I})\, Q(\mathcal{X}^{\infty} \propto \ \mathcal{X}^{(m)})}{\pi(\mathcal{X}^{(m)} \quad \mathcal{I})\, Q(\mathcal{X}^{(m)} \propto \ \mathcal{X}^{\infty})}, 1 \Big(,$$

and set $\mathcal{X}^{(m+1)} = \mathcal{X}^{(m)}$ with probability $1 \quad a(\mathcal{X}^{(m)} \propto \ \mathcal{X}^{\infty})$. The difficulty with lies in constructing useful proposals $q(\mathcal{X} \propto \ \mathcal{X}^{\infty})$. This is because there are typically a large number of trees with non-negligible posterior probability which are structurally dissimilar, separated by decision trees with low posterior probability. Hence, if $q(\mathcal{X} \propto \ \mathcal{X}^{\infty})$ places its mass on trees which are structurally similar to $\mathcal{X}$, one expects the sampler to get stuck in a local mode of $\pi(\mathcal{X} \quad \mathcal{I})$. Works addressing the construction of proposal distributions to use with Metropolis-Hastings include Chipman *et al.* (1998), Pratola (2016), Wu *et al.* (2007), and Lakshminarayanan *et al.* (2015).

The transition function $q(\mathcal{X} \propto \ \mathcal{X}^{\infty})$ is constructed as a mixture distribution over a collection of possible modifications to $\mathcal{X}$. The following modifications proposed by Chipman *et al.* (1998) are widely used in implementations; they all suffer from the same problem of being "local" modifications to $\mathcal{X}$. Let $\mathcal{R}_{\mathcal{X}}$ denote the set of branches whose children are both leaves (i.e., $\mathcal{R}_{\mathcal{X}}$ is the set of branches without grandchildren).

≡ Grow: Randomly select $\eta \top \mathcal{N}_{\mathcal{X}}$ and turn it into a branch node with two children, randomly sampling the predictor and cutpoint for the splitting rule from the prior.

≡ Prune: Select $\nu \top \mathcal{R}_{\mathcal{X}}$. Make ν a leaf by deleting its children.

≡ Change: Randomly select $\nu \top \mathcal{L}_{\mathcal{X}}$ and change the decision rule $[x_j \geq C]$ by sampling (j, C) according to the prior.

To complement these local moves, Wu *et al.* (2007) propose a non-local *radical restructure* move, which proposes $\mathcal{X}^{\infty}$so that the data partition of $\mathcal{X}$ is preserved; hence, one moves to a very different tree structure while simultaneously preserving the likelihood of the data. Pratola (2016) develops global Metropolis-Hastings proposals by constructing tree-rotation operators on the space of trees, and further develops a generalization of the "change" rule. These additional moves can provide massive improvements in mixing of the Markov chain, but unfortunately we are unaware of any publicly available software implementations.

3.2. Sequential Monte Carlo

Several authors have proposed sequential Monte Carlo (SMC) algorithms to approximate the posterior distribution, bypassing MCMC. By taking advantage of particle systems which evolve (almost) in parallel, they are also able to explore different modes of the posterior. Additionally, SMC methods may be easier to implement in practice than MCMC methods. The strategies described here are based on a generic *particle filtering* approach; see Cappé *et al.* (2007) for a review.

To apply SMC it is helpful to introduce auxiliary variables $\mathcal{X}^{(0)}, \ldots, \mathcal{X}^{(T)} = \mathcal{X}$ with joint distribution $\pi(\mathcal{X}^{(0)})\pi(\mathcal{X}^{(0)} \propto \ \mathcal{X}^{(1)}) \circ\circ\circ \pi(\mathcal{X}^{(T\ \ 1)} \propto \ \mathcal{X}^{(T)})$ so that we can view the model as a state-space model. Lakshminarayanan *et al.* (2013) do this by considering $\mathcal{X}^{(t)}$ to be the state of $\mathcal{X}$ in the branching process prior described in Section 2.1 truncated to t steps. At each stage of the particle filter the density $h(\mathcal{X}^{(t)}) \prime \ \pi(\mathcal{X}^{(t)})m(\mathcal{I} \quad \mathcal{X}^{(t)})$ is targeted, where $m(\mathcal{I} \quad \mathcal{X}^{(t)})$ represents the marginal likelihood of the data given that $\mathcal{X} = \mathcal{X}^{(t)}$. Given a proposal kernel $q(\mathcal{X}^{(t\ \ 1)} \propto \ \mathcal{X}^{(t)})$ (possibly depending on $\mathcal{I}$) this leads to the sequential approximation (with $w_m^{(0)} \leq 1$)

$$h\big)\mathcal{X}^{(t)}\big(\ll \frac{1}{\prod_{m=1}^{M} w_m^{(t)}} \prod_{m=1}^{M} w_m^{(t)} \delta_{\mathcal{X}_m^{(t)}}, \quad w_m^{(t)} \prime \ w_m^{(t\ \ 1)} \frac{\pi\big)\mathcal{X}^{(t\ \ 1)} \propto \ \mathcal{X}^{(t)}\big(m\big)\mathcal{I} \quad \mathcal{X}^{(t)}\big(}{q \ \mathcal{X}^{(t\ \ 1)} \propto \ \mathcal{X}^{(t)\mathsf{I}}\, m \ \mathcal{I} \quad \mathcal{X}^{(t\ \ 1)\mathsf{I}}}, \tag{3.1}$$

where $\delta_{\mathcal{X}}$ denotes a point-mass distribution at $\mathcal{X}$. To prevent weight degeneracy, one may resample $\mathcal{X}^{(t)}$ from the approximation of $h(\mathcal{X}^{(t)})$ and set $w_m^{(t)} \to \prod_m w_m^{(t)}/M$. There are several possibilities for choosing $q(\mathcal{X}^{(t\ \ 1)} \propto \ \mathcal{X}^{(t)})$, but Lakshminarayanan *et al.* (2013) recommend simply using $\pi(\mathcal{X}^{(t\ \ 1)} \propto \ \mathcal{X}^{(t)})$ and choosing M large.

Taddy *et al.* (2011) take a different approach, directly using a dynamic model $\mathcal{X}^{(0)}, \ldots, \mathcal{X}^{(N)}$ for observations $i = 1, \ldots, N$ in which an evolving predictive distribution $f(Y_i \quad X_i, \mathcal{I}^{(i\ \ 1)}, \mathcal{X}^{(i\ \ 1)})$ is specified, where $\mathcal{I}^{(n)} = \}(X_i, Y_i)|_{i=1}^{n}$. The trees $\mathcal{X}^{(t)}$ are assumed to evolve according to a Markov process (growing, pruning, or staying the same). This algorithm leads to weights of the form $w_m^{(i)} = w_m^{(i\ \ 1)} f(Y_i \quad X_i, \mathcal{I}^{(i\ \ 1)}, \mathcal{X}^{(i\ \ 1)})$ in (3.1), and we note that Taddy *et al.* (2011) recommend resampling steps after every update.

3.3. Bayesian additive regression trees and Bayesian backfitting

BART is fit by combining the Metropolis-Hastings schemes in Section 3.1 with a so-called Bayesian backfitting algorithm (Hastie and Tibshirani, 2000). The use of shallow trees by BART has a large computational payoff, as the "local moves" described in Section 3.1 are much more effective at exploring the posterior of shallow trees. Consider the regression setting with $Y_i = \prod_{t=1}^{T} g(X_i; \mathcal{X}_t, \Theta_t) + \epsilon_i$. The Bayesian backfitting algorithm proceeds by Gibbs sampling, updating $(\mathcal{X}_t, \Theta_t \quad \mathcal{I}, \mathcal{X}_{\ t}, \Theta_{\ t})$ where $\mathcal{X}_{\ t} = (\mathcal{X}_j : j \neq t)$ and $\Theta_{\ t} = (\Theta_j : j \neq t)$. The term "backfitting" comes from the observation that one can write

$$R_{it} = g\,(X_i; \mathcal{X}_t, \Theta_t) + \epsilon_i, \quad \text{where } R_{it} = Y_i \quad \prod_{j \neq t} g\,\Big)X_i; \mathcal{X}_j, \Theta_j\Big(.$$

After conditioning on $(\mathcal{X}_{\ t}, \Theta_{\ t}, \mathcal{I})$, this reduces to the BCART problem for regression, with the pseudo-response R_{it} playing the role of Y_i; hence, the MCMC strategies in Section 3.1 can be used with R_{it} in place of Y_i.

4. Illustrations

4.1. Nonparametric regression and classification

We now compare and contrast the various methods presented here on both simulated and real datasets. First we consider the `lidar`, which is available in the package `SemiPar` in `R` (Wand, 2014) and consists of measurements from a light detection and ranging experiment; the goal is to use the distance light travels before being reflected to predict the ratio of received light from two laser sources. This data is amenable to the semiparametric regression model (2.1). Because the predictor is one-dimensional, it allows for easy visual comparisons of methods. We consider the BCART model given in (2.1), the BART model (2.3), as well as the dynamic regression trees model of Taddy *et al.* (2011). We also consider the smoothed variant SBART of BART described in Section 2.2, the random forests algorithm of Breiman (2001) as implemented in the `grf` package, and the tree boosting algorithm in the package `xgboost`. Fits to the `lidar` dataset are given in Figure 3.

From Figure 3, we see that BCART and dynamic trees induce the least amount of smoothing, and essentially produce step functions. For comparison, the random forests algorithm produces a smoother estimate. BART smooths the data more than BCART, and produces a fit which is very similar to gradient boosting and random forests. The smoothest fit is given by SBART, which takes advantage of an assumed continuity of the regression curve.

Our experience is that the additional smoothing obtained by random forests, BART, gradient boosting, and SBART almost always leads to improved performance. Moreover, it is not clear if the lack of smoothness in BCART and dynamic trees is due to fundamental properties of the prior or is merely a symptom of inaccuracies in the Monte Carlo approximations. We have found dynamic trees to generally perform better than BCART, and speculate that this is due to the use of particle filtering rather than MCMC to fit the model.

To see more clearly the potential benefits of smoothing, we consider

$$Y_i = \mu(X_i) + \epsilon_i, \quad \mu(x) = 10\sin(\pi x_1 x_2) + 20(x_3 \quad 0.5)^2 + 10x_4 + 5x_5, \tag{4.1}$$

where $X_i \approx \text{Uniform}([0,1]^{10})$ and $\epsilon_i \overset{\text{iid}}{\approx} \text{Normal}(0, \sigma^2)$. This function was introduced by Friedman (1991) to illustrate the use of multivariate adaptive regression splines. We consider $\sigma^2 = 1$ and a sample size of $N = 250$, and evaluate methods by integrated root mean squared error }RMSE =

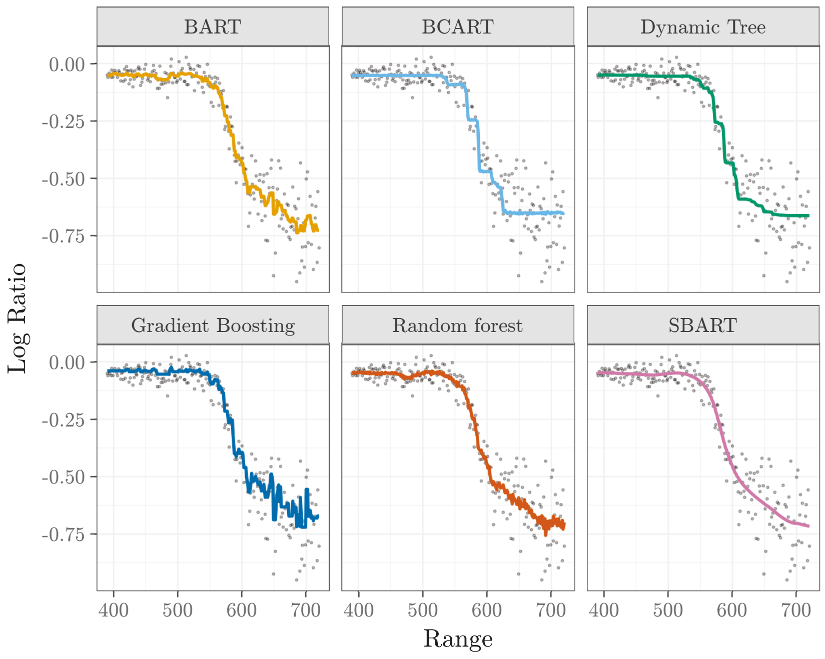

Figure 3: *Fits of various methods to the* `lidar` *dataset. BART = Bayesian additive regression trees; BCART = Bayesian classification and regression trees; SBART = smoothed Bayesian additive regression trees.*

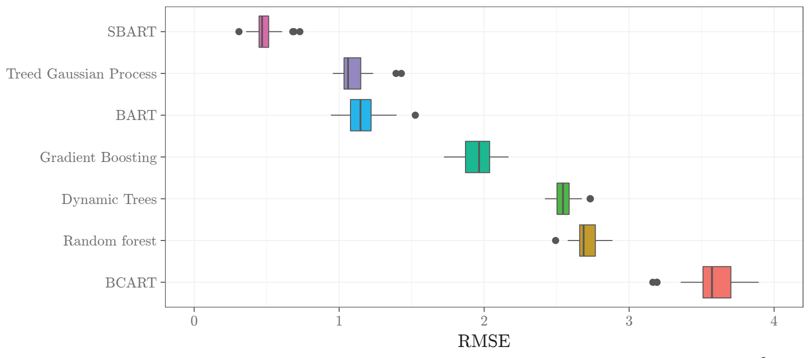

Figure 4: *Integrated RMSE under* (4.1) *across 30 independent replications, with* $n = 250, \sigma^2 = 1$*. SBART = smoothed Bayesian additive regression trees; BART = Bayesian additive regression trees; BCART = Bayesian classification and regression trees; RMSE = root mean squared error.*

$\sum\mu(x) \quad \hat{\mu}(x))^2 \, dx|^{1/2}$ where $\hat{\mu}(x)$ is an estimate of $\mu(x)$ produced by each method. Results on 30 replications are given in Figure 4. In addition to the methods used on the `lidar` data, we also consider the treed Gaussian process described in Example 2.

On (4.1), Figure 4 demonstrates that smoother methods typically do better. The best performing method is SBART, followed by treed Gaussian processes and BART. BCART performs the worst, however the fact that it is outperformed substantially by dynamic trees suggests that this may be due to poor mixing of the MCMC scheme.

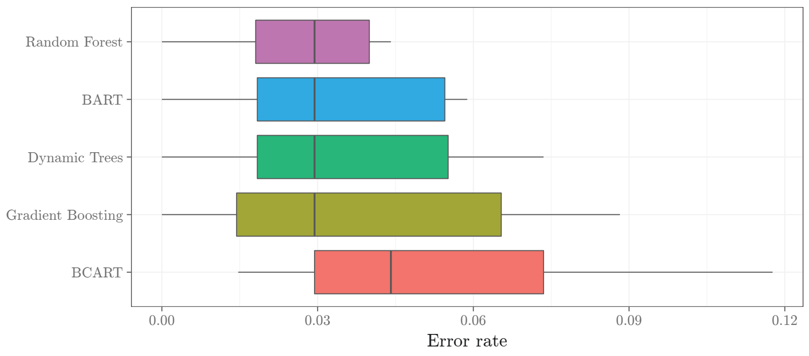

Figure 5: *Error rates for the competing methods on the Wisconsin breast cancer dataset for each fold in the cross-validation experiment. BART = Bayesian additive regression trees; BCART = Bayesian classification and regression trees.*

Finally, we consider the performance of these methods on the Wisconsin breast-cancer detection dataset. Given $P = 10$ features, the goal is to classify a tumor as benign or malignant. This dataset is available at the UCI machine learning repository (Lichman, 2013). We performed 10-fold cross validation on this dataset, and evaluated methods according to the 0–1 loss $L(Y, Y) = I(Y \neq Y)$. We did not consider SBART or treed Gaussian processes on this dataset due to the lack of software available to fit these methods on classification data. Results on the breast cancer dataset are given in Figure 5, and the trends are essentially the same as in the semiparametric regression setting. The primary difference is that random forests outperforms the Bayesian methods, with the out-of-fold average loss being 0.026, compared to 0.032 for BART and 0.037 for dynamic trees.

4.2. Overdispersed count data with the `Yelp` dataset

We consider the `Yelp` dataset, which consists of the reviews for restaurants in the city of Phoenix. This data is available publicly at https://www.yelp.com/dataset/challenge. We examine the number of reviews Y_i which are recorded for restaurant i. Intuitively, one might expect reviews for restaurant i to arrive roughly according to a Poisson process, leading to $Y_i \approx \text{Poisson}(\lambda_i)$.

As restaurants are generally quite heterogeneous, one expects the λ_i's to vary considerably. One mode of variability is in the spatial location of the restaurant, with restaurant's situated in more populated areas expected to accrue more reviews. In addition, unmeasured restaurant-specific variables might influence the restaurant-specific rate λ_i. To get a sense of this, Figure 6 displays a histogram of the Y_i's, and we note that the data is highly over-dispersed.

We set $\lambda_i = \delta_i \theta_{\eta(X_i)}$ where X_i consists of the latitude and longitude of restaurant i. The model $\delta_i \approx \text{Gam}(a, b)$ leads to a negative-binomial model within each leaf, with likelihood

$$\int_{i:\eta(X_i)=\eta} \frac{\Gamma(a+Y_i)}{Y_i!\Gamma(a)} \nu_\eta^{Y_i}(1 \quad \nu_\eta)^a = \frac{\int \Gamma(a+Y_i)}{\Gamma(a)^{n_\eta} \int Y_i!} \nu_\eta^{S_\eta}(1 \quad \nu_\eta)^{an_\eta},$$

where $\nu_\eta = \theta_\eta/(\theta_\eta + b)$ is the success probability of the negative binomial distribution with a trials. The likelihood of ν is conjugate to the Beta(α, β) distribution; using this prior, we obtain the marginal likelihood,

$$m(\mathcal{T} \quad \mathcal{X}) = \int_{\eta \mathcal{N}_X} \frac{\int_{i:\eta(X_i)=\eta} \Gamma(a+Y_i)}{\Gamma(a)^{n_\eta} \int_{i:\eta(X_i)=\eta} Y_i!} \bigcirc \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \bigcirc \frac{\Gamma(\alpha+S_\eta)\Gamma(\beta+an_\eta)}{\Gamma(\alpha+\beta+S_\eta+an_\eta)},$$

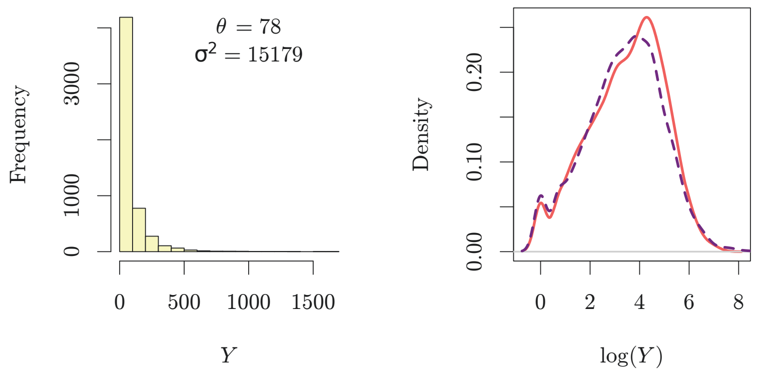

Figure 6: *Left: histogram of the raw* `Yelp` *data, where Y denotes the number of reviews a given restaurant has; the estimated mean μ and variance σ^2 are also given. Right: kernel density estimator of the distribution of* $\log(Y)$ *(solid) overlaid with the distribution of* $\log(Y)$ *under the negative binomial model with* $\alpha = 0.65$ *(dashed).*

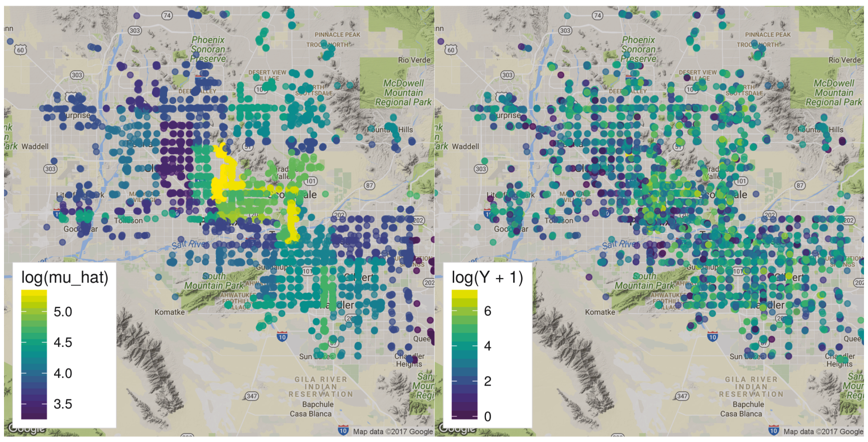

Figure 7: *Plots of the* `Yelp` *data. On the left, points are colored by the log of their mean response value according to the model, while on the right the points are colored by the log of their realized value. Figure created using the* `ggmap` *package (Kahle and Wickham, 2013).*

where $S_\eta = \prod_{i:\eta(X_i)=\eta} Y_i$ and $n_\eta = \}i : \eta(X_i) = \eta|$. The right panel of Figure 6 displays the fit of this model, when each observation is assigned its own leaf, to the data when $\alpha = 0.65$ and (β, a) are chosen by matching the moment-matching equations

$$\mu = \frac{a\alpha}{\beta \quad 1}, \qquad \sigma^2 = \frac{\mu^2 + \mu}{\beta \quad 2},$$

where σ^2 is the empirical variance and μ is the empirical mean of the data (leading to $a = 168$ and $\beta = 2.4$). We see that the negative binomial model is capable of accurately modeling the marginal distribution of Y. Accordingly, we set $(a, \alpha, \beta) = (168, 0.65, 2.4)$ when applying BCART.

Figure 7 displays the fit of BCART to the `Yelp` data. The BCART model effectively de-noises the raw data and reveals areas which are more likely to have a high number of reviews.

5. Connections to machine learning algorithms

Tree-based Bayesian methods possess a number of interesting connections to their algorithmic cousins. We describe points of contact between Bayesian methods and conventional machine learning methods. In addition to the obvious connections between BCART and random forests and connections between BART and boosting, we provide less well-known connections between BART and kernel-based methods, focusing in particular on Gaussian process regression.

5.1. Connections to random forests

BCART is similar operationally to random forests. For regression tasks, both methods make predictions of the form

$$\mu(x) = \mathbb{E}\,\}g\,(x; \mathcal{X}, \Theta_{\mathcal{X}})|\ , \tag{5.1}$$

where $\mathbb{E}(\cdot)$ is the expectation operator associated to $(\mathcal{X}, \Theta_{\mathcal{X}}) \approx \mathbb{P}$ for some probability distribution $\mathbb{P}$. Further, both approximate the expectation in (5.1) through Monte Carlo, using $\mu(x) = M^{\ 1} \prod_{m=1}^{M} g(x; \mathcal{X}^{(m)}, \Theta^{(m)}_{\mathcal{X}^{(m)}})$ to produce predictions, where $(\mathcal{X}^{(m)}, \Theta^{(m)}_{\mathcal{X}^{(m)}})$ are (ideally) sampled from $\mathbb{P}$. The random forests algorithm samples exactly $(\mathcal{X}^{(m)}, \Theta^{(m)}_{\mathcal{X}^{(m)}}) \overset{\text{iid}}{\approx} \mathbb{P}$, while BCART instead samples from a Markov chain with invariant distribution $\mathbb{P}$. Consequently, BCART and random forests have similar operating properties; for example, random forests performs better as $T \propto \in$ for the same reason that longer Markov chains are preferred with BCART (adding more samples reduces the Monte Carlo error).

BCART and random forests differ in the distribution $\mathbb{P}$ that they aim to sample from. A large advantage random forests have over BCART is that one can sample *exactly* from $\mathbb{P}$; random forests grow trees in a greedy fashion, with randomness coming from (i) bootstrap sampling the data used to construct the tree and (ii) random sub-sampling of features when constructing splits. BCART, by contrast, takes $\mathbb{P}$ to be the posterior distribution $\pi(\mathcal{X}, \Theta_{\mathcal{X}} \quad \mathcal{I}\ )$, which cannot be sampled from exactly.

5.2. Connections to kernel-based methods

BART has an interesting connection to kernel-based estimators. Note that the BART model can equivalently be written

$$g(x) = T^{\ \frac{1}{2}} \prod_{t=1}^{T} g(x; \mathcal{X}_t, \Theta_t), \qquad \mu_\eta \overset{\text{indep}}{\approx} \text{Normal}\,\big)0, \sigma^2_\mu\big(\,.$$

The random variables $\}g(x; \mathcal{X}_t, \Theta_t) : t \sim 1|$ are iid Normal$(0, \sigma^2_\mu)$ random variables, implying $g(x) \approx$ Normal$(0, \sigma^2_\mu)$. More generally, the multivariate central limit theorem implies that, for any collection of design points $(x_1, \ldots, x_M)^{\mathcal{D}}$, the vector $z = (g(x_1), \ldots, g(x_m))^{\mathcal{D}}$ has a limiting multivariate Gaussian distribution

$$z \propto \ \text{Normal}(0, \Sigma), \quad \Sigma_{ij} = \sigma^2_\mu \Pr(x_i \approx x_j), \quad (\text{as } T \propto \ \in\ ),$$

where $[x_i \approx x_j]$ is the event that x_i and x_j are associated to the same terminal node in $\mathcal{X}_1$ (the equality $\text{Cov}(g(x_i), g(x_j)) = \sigma^2_\mu \Pr(x_i \approx x_j)$ is easily checked). This suggests the following result.

Theorem 1. (Heuristic) *Consider the BART specification* (2.3) *with prior as described in Section 2.2 with a fixed prior $\pi_{\mathcal{X}}$ (which is not assumed to be a branching process prior). Then, as*

$T \propto \in, \}\mu(x) : x \top \mathcal{Y}|$ converges to a Gaussian process with mean function $m(x) = 0$ and covariance function $\Sigma(x, x^{\propto}) = \sigma_\mu^2 \Pr(x \approx x^{\propto})$.

"Theorem" 1 is heuristic because a formally correct statement requires technical conditions on the prior $\pi_{\mathcal{X}}$. Theorem 1 is intuitively implied by the convergence of the finite dimensional joint distributions noted above, but giving a formal statement and proof is an exercise in empirical process theory (Van der Vaart and Wellner, 1996) that we omit.

This connection to Gaussian processes is interesting from several standpoints. First, it provides a link between BART and kernel-based methods such as Gaussian process regression and support vector machines. Second, it formalizes several known connections between decision tree ensembling methods and kernel-based methods. For example, Scornet (2016) provides connections between random forests and kernel-methods, which were known also to Breiman (2000). Remarkably, kernels corresponding the BART, random forests, and Mondrian forests are very similar and often coincide; in particular, the kernels are similar to the exponential kernel described in the following proposition.

Proposition 1. *Consider the BART model* (2.3) *constructed so that (i) for any $\eta \top \mathcal{N}_t$, $D(\eta) \approx$* Poisson(λ)*; (ii) for any $\eta \top \mathcal{L}_t$, $j(\eta) \approx$* Categorical(s)*; (iii) given $j(\eta)$, $C_\eta \approx$* Uniform(0, 1) *(*allowing *for logically empty nodes); (iv) given $\mathcal{X}_1, \ldots, \mathcal{X}_T$, $\mu_\eta \overset{iid}{\approx}$* Normal($0, \sigma_\mu^2$)*. Then the associated kernel function $\Sigma(x, x^{\propto}) = \sigma_\mu^2 \Pr(x \approx x^{\propto})$ is given by*

$$\Sigma(x, x^{\propto}) = \sigma_\mu^2 \exp\Big) \; \lambda \prod_{j=1}^{P} s_j \Big(x_j \quad x_j^{\propto} \Big(\tag{5.2}$$

A very similar result is obtained by Balog *et al.* (2016) in the context of their Mondrian forests algorithm, giving some evidence that all of these methods are fundamentally making use of the same information about the dissimilarity of the points $(x, x^{\propto})$ in Euclidean space. Balog *et al.* (2016) use this connection to approximate inference under the "ideal" kernel (5.2). We note that samples from the BART prior could also be used for such a purpose.

A natural question one might ask is "given this connection, why not use Gaussian process regression directly?" First, Gaussian process regression does not scale favorably with the sample size; without making any approximations, training a Gaussian process regression model requires $\Theta(n^3)$ computations. Furthermore, our experience is that the empirical performance of a minimally-tuned implementation of BART is frequently *better* than Gaussian process regression using the equivalent kernel $\Sigma(x, x^{\propto})$. That is, BART provides better performance using less computation. We conjecture that the reason for BART outperforming Gaussian process regression is that limiting the number of trees in the ensemble allows one to learn a data-adaptive notion of distance between points.

A natural question one might ask is "given this connection, why not use Gaussian process regression directly?" First, Gaussian process regression does not scale favorably with the sample size; without making any approximations, training a Gaussian process regression model requires $\Theta(n^3)$ computations. Furthermore, our experience is that the empirical performance of a minimally-tuned implementation of BART is frequently *better* than Gaussian process regression using the equivalent kernel $\Sigma(x, x^{\propto})$. That is, BART provides better performance using less computation. We conjecture that the reason for BART outperforming Gaussian process regression is that limiting the number of trees in the ensemble allows one to learn a data-adaptive notion of distance between points.

5.3. Connections to boosting

The original motivation for the BART algorithm given by Chipman *et al.* (2010) was to provide a Bayesian analog to boosted decision trees (Freund *et al.*, 1999). Boosted decision tree methods take essentially the same form as BART, setting $\hat{\mu}(x) = \prod_{t=1}^{T} g(x; \mathcal{T}_t, \Theta_t)$ and, like BART, a preference is given for the individual trees to be *weak learners*. Unlike random forests, ensembles of decision trees obtained from boosting are typically such that (i) the individual decision trees are shallow and (ii) no single decision tree is sufficient to provide an adequate fit to the data on its own.

BART differs from boosting in a number of ways. First, boosting algorithms are greedy, and do not update trees once they are included in the model. By contrast, BART makes use of a Bayesian backfitting which continuously updates all trees in the model. Second, boosting algorithms run the risk of overfitting the data as more trees are added, while BART is shielded from overfitting as more trees are added due to its link with Gaussian processes noted in Section 5.2. Third, the Bayesian paradigm also allows for automatic tuning of hyperparameters through the use of priors, while boosting is typically tuned by cross-validation.

6. Frequentist properties of trees and tree ensembles

Beyond consistency, one may be interested in the convergence rate of the posterior. A natural target to aim for is the minimax rate for a particular estimation problem. For the semiparametric regression problem $Y_i = \mu_0(X_i) + \epsilon_i, \epsilon_i \overset{\text{iid}}{\approx} \text{Normal}(0, \sigma_0^2)$, the optimal rate of convergence is $\epsilon_n = n^{\ d/(2d+Q_0)}$ whenever $f_0(x)$ is d-times differentiable and depends on $Q_0 \leftarrow P$ predictors (more precisely, the assumption is that $f_0(x)$ is a d-Hölder function; see, e.g., Yang and Tokdar, 2015) where, for simplicity, P and Q_0 are regarded as fixed. This problem was considered by Linero and Yang (2017), whose results imply that the BART posterior concentrates at the rate ϵ_n (up-to logarithmic factors), $d \top (0, 1]$. Independently, Rockova and van der Pas (2017), using a specifically chosen prior $\pi_{\mathcal{T}}$, established contraction of the posterior BART and BCART priors at the rate ϵ_n (up-to logarithmic factors) under essentially the same assumptions.

The limitation $d \top (0, 1]$ described above is informative. When $d = 1$, $\mu_0(x)$ is a Lipschitz-continuous function; $d < 1$ corresponds to fractionally smooth $\mu_0(x)$; and $d > 1$ corresponds to functions which are smoother than Lipschitz-continuous. This suggests that BART and BCART can adapt to rough functions ($d \geq 1$) but cannot take advantage of high-order smoothness in $\mu_0(x)$. Linero and Yang (2017) show that this barrier can be overcome by smoothing the decision trees in BART, leading to the SBART model.

In view of Rockova and van der Pas (2017), there is no difference in convergence rate between BART and BCART for the problem of estimating a sparse function $\mu_0(x)$; hence, the results described above do not fully account for the improved performance of BART relative to BCART. There are several practical reasons for expecting BART to outperform BCART. In particular, BART may outperform BCART for the following reasons: (i) BART uses shallow trees, so the mixing of the associated Markov chain is much better; (ii) realizations of BART are smoother, so that BART is better able to borrow information from neighboring points; and (iii) functions $\mu_0(x)$ which arise in practice are more structured than merely being sparse, and BART takes advantage of this structure. We focus on (iii) here. Yang and Tokdar (2015) establish additional minimax rates of convergence when $\mu_0(x)$ admits a sparse additive structure $\mu_0(x) = \prod_{v=1}^{V} \mu_{0v}(x)$. In particular, if $\mu_{0v}(x)$ is α_v-smooth and Q_{0v} sparse, the minimax rate of convergence is simply obtained by summing the squared component-specific convergence rates, i.e., the rate is $\epsilon_n^{\star} = \}\prod_{v=1}^{V} n^{\ 2\alpha_v/(2\alpha_v+Q_{0v})}|^{1/2}$. Comparing $\epsilon_n^{\star}$ to ϵ_n, note that $\epsilon_n^{\star}$ can be substantially smaller than ϵ_n as long as the number of *interactions* between the predictors is small.

if the true model is additive, for example, the $\epsilon_n^\star$ is limited only by the number V of additive components. Linero and Yang (2017) and Rockova and van der Pas (2017) independently show that the BART posterior concentrates at the rate $\epsilon_n^\star$ up-to logarithmic factors (assuming $\alpha_v \top (0, 1]$ for BART and $\alpha_v > 0$ for SBART). This fact hinges crucially on the fact that BART decomposes additively; hence, the BCART posterior is not thought to concentrate at the rate $\epsilon_n^\star$.

7. Discussion

In this paper we have reviewed Bayesian tree-based models, focusing on recent developments in computation and modeling. We also outlined connections to more traditional machine learning algorithms and reviewed some exciting results on posterior consistency which fill a previously long-standing gap between theory and practice. Much work remains to be done; in particular, further improvements in computation would make these methods more feasible in structured regression problems where boosting algorithms currently dominate (Nielsen, 2016).

Also important to the dissemination of these methods is the development of software, and currently there are many options. The fits in this paper were obtained using the `CRAN` packages `tgp` for treed Gaussian processes, `dbarts` for BART, and `dynaTree` for dynamic trees, and the non-`CRAN` package `SoftBart` available at www.github.com/theodds/SoftBART. Other packages of note include `bartMachine` and `BART` for fitting BART models.

Acknowledgements

This research was partially supported by NSF grant DMS-1712870 and DOD grant SOT-FSU-FATS-16-06.

References

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

Balog M, Lakshminarayanan B, Ghahramani Z, Roy DM, and Teh YW (2016). The Mondrian kernel. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*, New York, 32–41.

Breiman L (1996). Bagging predictors, *Machine Learning*, **24**, 123–140.

Breiman L (2000). *Some infinity theory for predictor ensembles* (Technical Report No. 577). Department of Statistics, University of California, Berkeley.

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

Breiman L, Friedman J, Stone CJ, and Olshen RA (1984). *Classification and Regression Trees*, Wadsworth International Group, Belmont.

Cappé O, Godsill SJ, and Moulines E (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, **95**, 899–924.

Chipman H, George EI, Gramacy RB, and McCulloch R (2013). Bayesian treed response surface models, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **3**, 298–305.

Chipman HA, George EI, and McCulloch RE (1998). Bayesian CART model search, *Journal of the American Statistical Association*, **93**, 935–948.

Chipman HA, George EI, and McCulloch RE (2010). BART: Bayesian additive regression trees, *The Annals of Applied Statistics*, **4**, 266–298.

Denison DG, Mallick BK, and Smith AF (1998). A Bayesian CART algorithm, *Biometrika*, **85**, 363–

377.

Efron B and Gong G (1983). A leisurely look at the bootstrap, the jackknife, and cross-validation, *The American Statistician*, **37**, 36–48.

Freund Y, Schapire R, and Abe N (1999). A short introduction to boosting, *Journal of Japanese Society for Artificial Intelligence*, **14**, 771–780.

Friedman JH (1991). Multivariate adaptive regression splines, *The Annals of Statistics*, **19**, 1–67.

Gramacy RB and Lee HKH (2008). Bayesian treed Gaussian process models with an application to computer modeling, *Journal of the American Statistical Association*, **103**, 1119–1130.

Hastie T and Tibshirani R (1987). Generalized additive models: some applications, *Journal of the American Statistical Association*, **82**, 371–386.

Hastie T and Tibshirani R (2000). Bayesian backfitting (with comments and a rejoinder by the authors), *Statistical Science*, **15**, 196–223.

Hastings WK (1970). Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, **57**, 97–109.

Hill JL (2011). Bayesian nonparametric modeling for causal inference, *Journal of Computational and Graphical Statistics*, **20**, 217–240.

Irsoy O, Yildiz OT, and Alpaydin E (2012). Soft decision trees. In *Proceedings of the 21st International Conference on Pattern Recognition*, Tsukuba, Japan, 1879–1822.

Kahle D and Wickham H (2013). ggmap: spatial visualization with ggplot2, *The R Journal*, **5**, 144–161.

Kass GV (1980). An exploratory technique for investigating large quantities of categorical data, *Applied Statistics*, **29**, 119–127.

Lakshminarayanan B, Roy DM, and Teh YW (2013). Top-down particle filtering for Bayesian decision trees. In *Proceedings of the International Conference on Machine Learning*, Atlanta, GA, 280–288.

Lakshminarayanan B, Roy DM, and Teh YW (2014). Mondrian forests: efficient online random forests. In *Advances in Neural Information Processing Systems*, Montreal, Canada, 3140–3148.

Lakshminarayanan B, Roy DM, Teh YW, and Unit G (2015). Particle Gibbs for Bayesian additive regression trees. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, San Diego, CA, 553–561.

Lichman M (2013). UCI machine learning repository, Retrieved November 10, 2017, from: http://archive.ics.uci.edu/ml

Linero AR (2016). Bayesian regression trees for high dimensional prediction and variable selection, *Journal of the American Statistical Association*, https://doi.org/10.1080/01621459.2016.1264957

Linero AR and Yang Y (2017). Bayesian regression tree ensembles that adapt to smoothness and sparsity. arXiv preprint arXiv:1707.09461.

Ma L (2017). Recursive partitioning and multi-scale modeling on conditional densities, *Electronic Journal of Statistics*, **11**, 1297–1325.

MacEachern SN (2016). Nonparametric Bayesian methods: a gentle introduction and overview, *Communications for Statistical Applications and Methods*, **23**, 445–466.

McCullagh P (1984). Generalized linear models, *European Journal of Operational Research*, **16**, 285–292.

Muliere P and Walker S (1997). A Bayesian non-parametric approach to survival analysis using Pólya trees, *Scandinavian Journal of Statistics*, **24**, 331–340.

Murray JS (2017). Log-linear Bayesian additive regression trees for categorical and count responses. arXiv preprint arXiv:1701.01503.

Neal RM (1995). Bayesian learning for neural networks (Doctoral dissertation), University of Toronto, Toronto.

Nielsen D (2016). Tree boosting with XGBoost: why does XGBoost win "every" machine learning competition? (Master's thesis), Norwegian University of Science and Technology, Trondheim.

Pratola MT (2016). Efficient Metropolis-Hastings proposal mechanisms for Bayesian regression tree models, *Bayesian Analysis*, **11**, 885–911.

Quinlan JR (1993). *C4. 5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo.

Rasmussen CE and Williams CKI (2006). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, MIT Press, Cambridge.

Rockova V and van der Pas S (2017). Posterior concentration for Bayesian regression trees and their ensembles. arXiv preprint arXiv:1078.08734.

Roy DM and Teh YW (2009). The Mondrian process, *Advances in Neural Information Processing Systems*, **21**, 1377–1381.

Scornet E (2016). Random forests and kernel methods, *IEEE Transactions on Information Theory*, **62**, 1485–1500.

Sparapani RA, Logan BR, McCulloch RE, and Laud PW (2016). Nonparametric survival analysis using Bayesian additive regression trees (BART), *Statistics in Medicine*, **35**, 2741–2753.

Taddy MA, Gramacy RB, and Polson NG (2011). Dynamic trees for learning and design, *Journal of the American Statistical Association*, **106**, 109–123.

Van der Vaart AW and Wellner JA (1996). *Weak Convergence and Empirical Processes: with Applications to Statistics*, Springer, New Work.

Wand M (2014). SemiPar: Semiparametric Regression. R package (Version 1.0-4.1).

Wu Y, Tjelmeland H, and West M (2007). Bayesian CART: prior specification and posterior simulation, *Journal of Computational and Graphical Statistics*, **16**, 44–66.

Yang Y and Tokdar ST (2015). Minimax-optimal nonparametric regression in high dimensions, *The Annals of Statistics*, **43**, 652–674

Received October 7, 2017; Revised November 17, 2017; Accepted November 17, 2017