Predicting human-driving behavior to help driverless vehicles drive: random intercept Bayesian Additive Regression Trees

Yaoyuan V. Tan*
Department of Biostatistics, University of Michigan
and
Carol A.C. Flannagan
University of Michigan, Transportation Research Institute
and
Michael R. Elliott
Department of Biostatistics, University of Michigan

Abstract

The development of driverless vehicles has spurred the need to predict human driving behavior to facilitate interaction between driverless and human driven vehicles. Predicting human driving movements can be challenging, and poor prediction models can lead to accidents between the driverless and human driven vehicles. We used the vehicle speed obtained from a naturalistic driving dataset to predict whether a human driven vehicle would stop before executing a left turn. In a preliminary analysis, we found that Bayesian additive regression trees (BART) produced less variable and higher AUC values compared to a variety of other state-of-the-art binary predictor methods. However, BART assumes independent observations, but our dataset consists of multiple observations clustered by driver. Although methods extending BART to clustered or longitudinal data are available, they are highly complex and lack readily available

*(e-mail: vincetan@umich.edu)
software for easy implementation. We extend BART to handle correlated observations by adding a random intercept and used a simulation study to determine it’s prediction performance, bias, and 95% coverage. We then successfully implemented our random intercept BART model to our clustered dataset and found substantial improvements in prediction performance compared to BART, linear logistic regression, and random intercept linear logistic regression.

Keywords: Bayesian additive regression trees, Classification and regression trees, Driverless vehicles, Hierarchical models, Longitudinal prediction, Transportation statistics.

1 Introduction

In transportation statistics, a new area of research brought about by improvements in artificial intelligence and engineering is the creation of the autonomous (self-driving) vehicle. These vehicles have been tested on city streets in certain locations since 2009. As of this writing, a number of companies have deployed or announced plans for deployment (Google, 2015; Mchugh, M., 2015; Davies, A., 2015). A major hurdle for self-driving vehicles driving on public roads is that these vehicles will have to interact with human-driven vehicles for the foreseeable future. Human drivers do not always communicate their plans to other drivers well. For example, when making a turn, the turn signal is the only explicit means of communicating plans, and even they are used with less than perfect reliability. Hence, the ability to deploy driverless vehicles on a large scale will critically depend on the development of a good prediction model for human driving behavior.

Building a prediction model that addresses all or most of the human driving behavior
is a massive and complex task. To keep this paper concise, we focus on the development of a preliminary model based on a single driving behavior: whether a human driver would stop at an intersection before executing a left turn. We are particularly interested in left turn stops because in countries with right-side driving, for example, US and China, left turn crashes can result in severe passenger-side impacts. Since left turn maneuvers already present a challenge for human drivers, we expect this maneuver to present difficulty for the driverless vehicle.

To develop the prediction model for left turn stops at intersections, we used a naturalistic driving study, the Integrated Vehicle Based Safety System (IVBSS) study (Sayer et al., 2011). Naturalistic driving studies (including the IVBSS) involve the collection of driving data from vehicles as they are piloted on actual roads. These driving data are collected by a data acquisition system (DAS) installed on a study subject’s vehicle or a research vehicle. Typical data collected include vehicle speed, brake application, and miles traveled.

Prediction models in statistics typically rely on regression models that require estimation of covariate main effects and interactions, and, when predictors are continuous or on a fine ordinal scale, assessment of non-linearities. In the settings where understanding associations or, under appropriate assumptions, causal mechanism between predictors and outcomes are of interest, approximations for non-linearities and averaging over interactions might be used to develop summaries to ease interpretation. In prediction, since obtaining the most accurate forecast is the goal, estimating highly complex non-linearities, including the interactions, is at a premium, as long as these non-linearities are true signals and not noise.

Perhaps the most common method for modeling non-linearity is to use a polynomial
transformation for a covariate, usually centered at the mean to reduce correlation. More sophisticated approaches use penalized splines or additive models that only require assumptions of smoothness (existence of derivatives) to obtain consistent estimates of a non-linear trend (Hastie and Tibshirani 1990; Ruppert et al. 2003). Modeling of non-linear interactions between two predictors using thin-plate splines (Franke 1982) can quickly become difficult, suffering from the “curse of dimensionality”, as the data required to estimate high-dimensional surfaces become enormous. In the binary outcomes setting, methods such as classification and regression trees (CART; Breiman et al. 1984) as well as more sophisticated machine learning techniques such as artificial neural networks (ANN; Smith et al. 1993) and support vector machines (SVM; Gammermann, 2000) are commonly used. Although CART is able to model complex interactions naturally, it faces difficulty when modeling non-linear interactions. In contrast, ANN and SVM excel at modeling non-linearities but may face difficulties when modeling complex interactions.

Because our goal is prediction and its associated interactions, we prefer regression methods that are able to account for highly complex non-linearities, including interactions. Based on preliminary analyses, we chose the Bayesian additive regression trees (BART; Chipman et al. 2010) to predict whether a human-driven vehicle would stop before executing a left turn at an intersection. Because BART was designed for independent subjects, we needed to extend BART to take into account the clustering in our dataset. We are aware of two papers that extended BART to handle longitudinal or clustered observations: Zhang et al. (2007) used a spatial random intercept BART to merge two datasets, and Low-Kam et al. (2015) did so in a dose-finding toxicity study. Zhang et al. developed an imputation model for a statistical matched problem (Rassler, 2002) that used BART with a conditional
auto-regressive distribution for the random intercept. Since the within-subject correlation of our dataset was induced by repeated measurements and not spatial effects, the distribution Zhang et al. placed on the random intercept may not be appropriate. Low-Kam et al. investigated the associations between the physico-chemical properties of nanoparticles and their toxicity profiles over multiple doses. The complex nature of their goal prompted them to first specify an autoregressive covariance matrix with truncated support on $[0, 1]$ to handle the correlated measurements, and then they specified a conditionally conjugate P-spline prior for the terminal nodes of the regression trees. The complexity of their method makes implementation to our dataset difficult since our outcomes are binary. Neither papers provided convenient software for implementing their methods.

Motivated by the lack of an appropriate and straightforward method to implement BART to handle the clustered nature of our dataset, we propose an extension of BART to account for clustered or longitudinal observations. Our proposed method follows Zhang et al. (2007) by adding a random intercept to BART. However, we differ in the distribution we assumed for our random intercept. We consider the more common normal distribution on the random intercept with various prior distributions on the within-subject correlation parameter. Since our method is based on adding a random intercept to BART, we call this the random intercept BART (riBART). Compared to Zhang et al., our method employs distributions that appear commonly in longitudinal and repeated measurements literature. Compared to Low-Kam et al. (2015), our method is more straightforward and can be seen as a hierarchical analysis. Our method also has the potential to be extended to handle multiple linear random effects. Finally, we provide a strategy to easily implement riBART by making use of the existing BART implementation packages in R (R Core Team, 2015).
The rest of our paper is organized as follows. In Section 2, we provide a review of BART. In Section 3, we present riBART and our strategy to implement riBART. In Section 4, we conduct a simulation study to compare the performance of BART and riBART when applied to correlated datasets. In Section 5, we implement riBART on our dataset and compare its prediction performance with BART, linear logistic regression, and random intercept linear logistic regression. Finally, we conclude with a discussion and possible future work in Section 6.

2 Bayesian Additive Regression Trees

2.1 Continuous outcomes

Denote a continuous outcome $Y_k$ with associated $p$ covariates $X_k = \{X_{k1}, \ldots, X_{kp}\}$ for $k = 1, \ldots, n$ subjects. BART models the outcome as

$$Y_k = \sum_{j=1}^{m} g(X_k, T_j, M_j) + \epsilon_k \quad \epsilon_k \overset{i.i.d.}{\sim} N(0, \sigma^2)$$  \hspace{1cm} (1)

where $T_j$ is the $j^{th}$ binary tree structure and $M_j = \{\mu_{1j}, \ldots, \mu_{bj}\}$ is the set of $b_j$ terminal node parameters associated with tree structure $T_j$. Typically, the number of trees $m$ is fixed and no prior distribution is placed on $m$. (Chipman et al. 2010) (henceforth, CGM) suggested fixing $m$ at 200 as this performs well in many situations. Alternatively, cross-validation could be used to determine $m$.

The binary tree $T_j$ is made up of both internal nodes and terminal nodes. At each internal node, there is a decision rule that splits estimation of the mean of $Y_k$ depending on the covariates $X_k$. For example in Figure 1, the first internal node at the top of the
tree drops the mean to the left if the corresponding covariate $X_{k2} < 100$ or to the right if $X_{k2} \geq 100$. At a terminal node (a node with no decision rules to split an outcome), the sample mean of the outcomes allocated to the terminal node can be calculated to obtain the parameter $\mu_{ij}$ at the terminal node. Thus, $g(X_k, T_j, M_j)$ can be viewed as the $j^{th}$ function that assigns the mean $\mu_{ij}$ to the $k^{th}$ outcome, $Y_k$.

The joint prior distribution for $[(T_1, M_1), \ldots, (T_m, M_m), \sigma]$ is $P[(T_1, M_1), \ldots, (T_m, M_m), \sigma]$. Note that by the independence of $\epsilon_k$ and $(T_j, M_j)$ as well as the independence between all $m$ tree structures and terminal node parameters, the joint prior distribution $P[(T_1, M_1), \ldots, (T_m, M_m), \sigma]$ can be decomposed as

$$P[(T_1, M_1), \ldots, (T_m, M_m), \sigma] = \prod_{j=1}^{m} P(T_j, M_j) P(\sigma)$$

$$= \prod_{j=1}^{m} P(M_j|T_j) P(T_j) P(\sigma)$$

$$= \prod_{j=1}^{m} \left\{ \prod_{i=1}^{b_j} P(\mu_{ij}|T_j) \right\} P(T_j) P(\sigma).$$

where $i = 1, \ldots, b_j$ indexes the terminal node parameters in tree $j$. This implies that we need to impose priors on $T_j$, $\mu_{ij}$, and $\sigma$ in order to obtain the posterior distributions of $T_j$, $\mu_{ij}$, and $\sigma$. CGM suggested the following prior distributions on $\mu_{ij}$ and $\sigma$:

$$\mu_{ij}|T_j \sim N(\mu_{\mu}, \sigma_{\mu}^2),$$

$$\sigma^2 \sim IG(\nu, \nu \lambda),$$

where $IG(\alpha, \beta)$ is the inverse gamma distribution with shape parameter $\alpha$ and rate parameter $\beta$. The prior distribution of $P(T_j)$ can be specified using three aspects: (i) the probability that a node at depth $d = 0, 1, 2, \ldots$ is an internal node given by $\alpha (1 + d)^{-\beta}$ where $\alpha \in (0, 1)$ and $\beta \in [0, \infty)$ so that $\alpha$ controls how likely a terminal node in the tree would split with
smaller $\alpha$ implying lesser likelihood a terminal node would split and $\beta$ controls the number of terminal nodes where a higher $\beta$ decreases the number of terminal nodes; (ii) the distribution used to choose which covariate to be selected for the decision rule in an internal node; and (iii) the distribution for the value of the selected covariate for the decision rule in an internal node. CGM suggests a discrete uniform distribution for the available covariates and values in both (ii) and (iii) respectively, although other more flexible distributions could be used [Kapelner and Bleich 2016].

In CGM, $\alpha = 0.95$ and $\beta = 2$. For $\mu_\mu$ and $\sigma_\mu$, they are set such that $\mathcal{N}(m\mu_\mu, m\sigma_\mu^2)$ assigns high probability to the interval $(\min(Y_k), \max(Y_k))$. This can be achieved by defining $v$ such that $\min(Y_k) = m\mu_\mu - v\sqrt{m\sigma_\mu}$ and $\max(Y_k) = m\mu_\mu + v\sqrt{m\sigma_\mu}$. For convenience, CGM suggested transforming the observed $Y_k$ as $\tilde{Y}_k = \frac{Y_k - \frac{\min(Y_k) + \max(Y_k)}{2}}{\frac{\max(Y_k) - \min(Y_k)}{2}}$, and then treating $\tilde{Y}_k$ as the dependent observation. This has the effect of allowing the hyperparameter of $\mu_\mu$ to be set as zero and $\sigma_\mu$ to be determined as $\sigma_\mu = \frac{0.5}{v\sqrt{m}}$ where $v$ is to be chosen. For $v = 2$, $\mathcal{N}(m\mu_\mu, m\sigma_\mu^2)$ assigns a prior probability of 0.95 to the interval $(\min(Y), \max(Y))$ and is the value CGM suggests. Finally for $\nu$ and $\lambda$, CGM suggested $\nu = 3$ and $\lambda$ is the value such that $P(\sigma^2 < s^2; \nu, \lambda) = 0.9$ where $s^2$ is the estimated variance of the residuals from the multiple linear regression with $Y_k$ as the outcomes and $X_k$ as the covariates.

This setup induces the posterior distribution $P[(T_1, M_1), \ldots, (T_m, M_m), \sigma|Y_k]$ which can be simplified to two major posterior draws using Gibbs sampling. First, draw $m$ successive

$$P[(T_j, M_j)|T(j), M(j), Y_k, \sigma]$$

for $j = 1, \ldots, m$, where $T(j)$ and $M(j)$ consist of all the tree structures and terminal nodes except for the $j$th tree structure and terminal node; then, draw $P[\sigma|(T_1, M_1), \ldots, (T_m, M_m), Y_k]$. 8
To obtain a draw from (2), note that this distribution depends on \((T_j, M_j, Y_k, \sigma)\) through

\[
R_{kj} = Y_k - \sum_{w \neq j} g(X_k, T_w, M_w),
\]  

the residuals of the \(m-1\) regression sum of trees fit excluding the \(j^{th}\) tree. Thus (2) is equivalent to the posterior draw from a single regression tree \(R_{kj} = g(X_k, T_j, M_j) + \epsilon_k\) or

\[
P[(T_j, M_j)|R_{kj}, \sigma].
\]

We can obtain a draw from (4) by first drawing from \(P(T_j|R_{kj}, \sigma)\) using a Metropolis-Hastings (MH) algorithm (Chipman et al., 1998, 2010; Kapelner and Bleich, 2016) and then drawing from \(P(M_j|T_j, R_{kj}, \sigma) \sim N(\sigma^2 \sum_{i} r_{ij} + \sigma^2 \mu_{ij}, \frac{\sigma^2 \sigma^2}{\sigma^2 + \sigma^2})\), where \(r_{ij}\) is the subset of elements in \(R_{kj}\) allocated to the terminal node with parameter \(\mu_{ij}\) and \(n_i\) is the number of \(r_{ij}\)s in \(R_{kj}\) allocated to \(\mu_{ij}\). Note that \(\mu = 0\) after transformation.

Complete details for the derivation of \(P(M_j|T_j, R_{kj}, \sigma)\) and \(P[\sigma|(T_1, M_1), \ldots, (T_m, M_m), Y_k]\) are provided in the supplementary materials available online. Explicit MH algorithm details for equation (4) can be found in Kapelner and Bleich (2016).

### 2.2 Binary outcomes

Extending BART to binary outcomes involve a modification of (1). First, let

\[
G(X_k) = \sum_{j=1}^{m} g(X_k, T_j, M_j).
\]

Using the probit formulation, the binary outcomes \(Y_k\) can be linked to (5) using \(P(Y_k = 1|X) = \Phi[G(X_k)]\) where \(\Phi[.]\) is the cumulative density function of a standard normal distribution. This formulation implicitly assumes that \(\sigma = 1\). Assuming once again that all \(m\) tree
structures and terminal node parameters are independent, this implies that we only need priors for \( T_j \) and \( \mu_{ij} \). CGM assumes that priors for \( T_j \) and \( \mu_{ij} \) and the hyperparameters for \( \alpha \) and \( \beta \) are the same as BART for continuous outcomes. However, for the hyperparameters of \( \mu_\mu \) and \( \sigma_\mu \), CGM suggested that \( \mu_\mu \) and \( \sigma_\mu \) should be chosen such that \( G(X_k) \) is assigned to the interval \((-3, 3)\) with high probability. This can be achieved by setting \( \mu_\mu = 0 \) and choosing an appropriate \( v \) in the formula \( \sigma_\mu = \frac{3}{v\sqrt{m}} \). Similar to the continuous outcome case, CGM suggested \( v = 2 \).

To draw from the posterior distribution \( P[(T_1, M_1), \ldots, (T_m, M_m)|Y_k] \), CGM proposed the use of data augmentation [Albert and Chib 1993; Tanner and Wong 1987]. This method proceeds by first generating a latent variable \( Z_k \) according to

\[
(Z_k|Y_k = 1) = N(0, \infty)(G(X_k), 1)
\]

\[
(Z_k|Y_k = 0) = N(-\infty, 0)(G(X_k), 1),
\]

where \( N(a, b)(\mu, \sigma^2) \) is the truncated normal distribution with mean \( \mu \) and variance \( \sigma^2 \) truncated to the range \((a, b)\). Once \( Z_k \) is drawn, \( P[(T_1, M_1), \ldots, (T_m, M_m)|Z_k] \) is drawn next as in (2)-(4) with the latent variables \( Z_k \) replacing \( Y_k \) in (2) and \( \sigma \) fixed at 1. Note that at each iteration, \( G(X_k) \) will be updated with the new \( (T_1, M_1), \ldots, (T_m, M_m) \) draws from \( P[(T_1, M_1), \ldots, (T_m, M_m)|Z_k] \) so that an updated draw of the latent variable \( Z_k \) can be obtained.

### 3 Random Intercept BART

We now extend BART to account for repeated measurements. To do so, we introduce to (1) a random intercept \( a_k, k = 1, \ldots, K \). Here, \( k \) still indexes the subjects but \( i = 1, \ldots, n_k \)
indexes the observations within a subject. With the addition of $a_k$, (1) becomes

$$Y_{ik} = \sum_{j=1}^{m} g(X_{ik}, T_j, M_j) + a_k + \epsilon_{ik} \sim i.i.d. N(0, \sigma^2), a_k \sim i.i.d. N(0, \tau^2), a_k \bot \epsilon_{ik}. \tag{6}$$

We decompose the joint prior distribution as (assuming $\sigma^2$ and $\tau^2$ are a priori independent)

$$P[(T_1, M_1), \ldots, (T_m, M_m), \sigma, \tau] = \left( \prod_{j=1}^{m} \{ \prod_{l=1}^{b_j} P(\mu_{lj}|T_j) \} P(T_j) \right) P(\sigma) P(\tau)$$

Next, we place the same prior distributions as the independent BART model for $T_j, \mu_{lj}$ (this is $\mu_{ij}$ for the independent BART model), and $\sigma^2$. There are various prior distributions we could place on $\tau^2$ and we discuss this in the next paragraph. We use the same hyperparameter values for $\alpha, \beta, \mu_{ij}, \sigma^2_{ij}$, and $\nu$ that CGM suggested for the independent BART model. For the setting of $\lambda$, we employed a slightly more sophisticated approach. We first estimated the outcomes $Y_{ik}$ using the multivariate adaptive regression splines (MARS) method [Friedman, 1991]. We then estimated an initial random intercept, $\hat{a}_k^{(0)}$, by taking the mean of the MARS residuals for each $k$. Finally, we obtained an initial estimate of $\sigma^2$ using

$$s^{(0)2} = \frac{\sum_{k=1}^{K} \sum_{n_k}^{n_k} (Y_{ik} - \hat{Y}_{ik}^{(0)} - \hat{a}_k^{(0)})^2}{N - N(1 - \sqrt{\frac{\text{RSS}}{\text{GCV} \times N}})}, \text{ where } N = \sum_{k=1}^{K} n_k, \text{ RSS and } GCV \text{ are the residual sum of squares and generalized cross-validation value from MARS, and } N(1 - \sqrt{\frac{\text{RSS}}{\text{GCV} \times N}}) \text{ is the effective number of parameters in MARS. Then } \lambda \text{ can be set as the value such that } P(\sigma^2 < s^{(0)2}; \nu, \lambda) = 0.9. \text{ We call this model the random intercept BART (riBART).}$$

We discuss three alternative specifications for the prior distribution of $\tau^2$, the within-subject correlation. The first specification is $P(\tau^2) \propto 1$, a flat improper prior which provides no further information to the posterior distribution. For random intercept models, Gelman [2006] noted that assuming $P(\tau^2) \propto 1$ may have inappropriate effects on inferences especially when $K$ is small or when $\tau$ is close to 0. This led us to our second alternative specification, a folded non-central $t$ (FNCT) prior on $\tau$, which can be achieved by reformulating the random
intercept as
\[ a_k = \xi \eta_k \quad \xi \sim N(0, B^2), \quad \eta_k \overset{i.i.d.}{\sim} N(0, \theta^2) \] (7)
and assuming that \( B^2, \theta^2, \sigma^2 \) and \( (T_j, M_j) \)‘s are independent. We assume that \( \theta^2 \sim IG\left(\frac{1}{2}, \frac{1}{2}\right) \) and take \( B \to \infty \) when evaluating the posterior distribution of \( \xi \). This effectively turns the FNCT prior on \( \tau^2 \) to a half-Cauchy prior (Gelman, 2006). The posterior draws of \( a_k \) and \( \tau \) can be obtained by setting \( a_k = \xi \eta_k \) and \( \tau = |\xi|\theta \). The flat improper prior and the half-Cauchy prior may lead to improper posterior distributions. Hence, we proposed a third and final alternative, the standard conjugate prior given by, \( \tau^2 \sim IG(1, 1) \).

To draw from the posterior distribution of riBART, we use ideas from Zhang et al. (2007) and Dorie et al. (2016). First, conditioning on \( a_k \), we remove it from the outcomes \( Y_{ik} \) to obtain \( \tilde{Y}_{ik} = Y_{ik} - a_k \). We then replace \( Y_{ik} \) with \( \tilde{Y}_{ik} \) in (3) to obtain \( \tilde{R}_{ikj} \) and use \( \tilde{R}_{ikj} \) to obtain the posterior draw of \( T_j | \tilde{R}_{ikj}, \sigma \). This posterior draw \( T_j | \tilde{R}_{ikj}, \sigma \) can be treated as the usual posterior draw of \( T_j \) in (4) with \( R_{kj} \) replaced by \( \tilde{R}_{ikj} \). We next draw \( M_j | T_j, \tilde{R}_{ikj}, \sigma \) in similar fashion. Once all the \( m \) \( T_j \) and \( M_j \)’s are drawn, we then draw the posterior of \( \sigma^2 | Y_{ikj}, (T_1, M_1), \ldots, (T_m, M_m), a_k \) and \( \tau^2 | Y_{ikj}, (T_1, M_1), \ldots, (T_m, M_m), a_k \) from the inverse-gamma distribution. Finally, we draw the posterior of \( a_k | Y_{ikj}, (T_1, M_1), \ldots, (T_m, M_m), \sigma, \tau \) from the normal distribution. We derive the explicit parameters of the posterior distributions of \( \sigma, a_k, \tau, \theta^2 \), and \( \eta_k \) in the supplementary materials available online.

Extending riBART to binary outcomes proceeds in similar fashion as binary outcomes for BART. We add \( a_k \) to (5) to obtain
\[ G_a(X_{ik}) = \sum_{j=1}^{m} g(X_{ik}, T_j, M_j) + a_k. \] (8)
We once again assume \( a_k \sim N(0, \tau^2) \). To link the sum of trees to the binary outcomes \( Y_{ik} \), we again use the probit link and write \( P(Y_{ik} = 1 | X_{ik}) = \Phi[G_a(X_{ik})] \). We suggest
prior distributions similar to the continuous outcomes riBART for $T_j$, $M_j$, and $\tau^2$. The same hyperparameters in BART for binary outcome can be used for $\alpha$, $\beta$, $\mu$, and $\sigma$. To obtain the posterior draws of $T_j$, $M_j$, $a_k$, and $\tau^2$, we employ the data augmentation method suggested by Albert and Chib (1996). First, we draw a latent variable $Z_{ik}$ according to

$$(Z_{ik}|Y_{ik} = 1) = N(0, \infty)(G_a(X_{ik}), 1)$$

$$(Z_{ik}|Y_{ik} = 0) = N(-\infty, 0)(G_a(X_{ik}), 1).$$

Next, we remove $a_k$ from $Z_{ik}$ to obtain $\tilde{Z}_{ik} = Z_{ik} - a_k$. Now, we replace $\tilde{Y}_{ik}$ with $\tilde{Z}_{ik}$ in riBART for continuous outcomes to obtain the posterior draws of $(T_1, M_1), \ldots, (T_m, M_m)$. We then use these posterior draws to draw the posterior distribution of $\tau^2$ and $a_k$. The posterior distribution for $\tau^2$ and $a_k$ is the same as riBART for continuous outcomes with $\sigma = 1$. An updated draw of $Z_{ik}$ can then be obtained using the most recent posterior draws of $(T_1, M_1), \ldots, (T_m, M_m)$, and $a_k$.

### 3.1 Implementation

While we have provided the full details of the Gibbs sampler for BART and riBART, in practice, implementing the posterior draw of the tree structure and terminal node parameters is quite daunting. Although the `dbarts()` function in R (Chipman et al., 2015) could be used to implement our proposed riBART model, it can only be applied to continuous outcomes as of this writing. Moreover, there is no easy way to manipulate the $\lambda$ parameter for the posterior draw of $\sigma$. Hence, we describe below an alternative strategy to implement this difficult part of the algorithm using standard BART package (e.g. `BayesTree` in R).

1. Begin with an initial estimate of $\sigma$ ($\sigma = 1$ for all iterations in binary outcomes) and
\( a_k \) (typically, \( a_k = 0 \)). For binary outcomes, use an existing BART package with default settings to get an initial estimate of \( G_a(X_{ik}), \hat{G}_a^{(0)}(X_{ik}) \). Then, use the data augmentation method to obtain an initial draw of \( Z_{ik}, \hat{Z}_{ik}^{(0)} \). Estimate \( \hat{a}_k^{(0)} \), by taking the mean of \( \hat{Z}_{ik}^{(0)} - \hat{G}_a^{(0)}(X_{ik}) \) for each \( k \). If the half-Cauchy prior is assumed for \( \tau^2 \), set \( \xi^{(0)} = 1 \) and \( \eta_k^{(0)} = 0 \) for all \( k \).

2. Subtract \( a_k \) from \( Y_{ik} \) (or \( Z_{ik} \) for binary outcomes) to obtain \( \tilde{Y}_{ik} \) (or \( \tilde{Z}_{ik} \)).

3. Provide the outcomes \( \tilde{Y}_{ik} \) (or \( \tilde{Z}_{ik} \)) with the covariates \( X_{ik} \) to any computer package or program that is able to implement continuous outcomes BART. To fix \( \sigma \), set the degrees of freedom for the prior distribution of \( \sigma \) to 100,000 and use the initial estimate of \( \sigma \) from Step 1 as the initial \( \sigma \) estimate in the BART program. Subsequent draws will use the most recent update of \( \sigma \). For binary outcomes, set \( \sigma = 1 \) for all draws and \( k = 6 \) to account for the difference in hyperparameter \( \sigma^2 \) between continuous outcomes and binary outcomes. Draw 100 posterior draws for the \( m T_j \) and \( M_j \)'s and compute

\[
\hat{y}_{ik} = \sum_{j=1}^{m} g(X_{ik}, T_j, M_j) | \tilde{Y}_{ik}, \sigma \quad \text{using the 100th posterior draw.}
\]

4. Draw \( \sigma^2 \) from \( IG\left(\frac{N + \nu}{2}, \frac{1}{2} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{(y_{ik} - \hat{y}_{ik} - a_k)^2 + \nu \lambda}{\tau^2 + \sigma^2}\right) \) where \( N = \sum_{k=1}^{K} n_k \).

5a Assuming \( P(\tau) \propto 1 \), draw \( \tau^2 \) from \( IG\left(\frac{K}{2} - 1, \frac{\sum_{k=1}^{K} a_k^2}{2}\right) \). Go to Step 6a.

5b Assuming Half-Cauchy prior for \( \tau^2 \), draw \( \theta^2 \) from \( IG\left(\frac{e + K}{2}, \frac{\sum_{k=1}^{K} \eta_k^2 + e f}{2}\right) \). Go to Step 6b.

5c Assuming \( \tau^2 \sim IG(1,1) \), draw \( \tau^2 \) from \( IG\left(\frac{K + 2}{2}, \frac{2 + \sum_{k=1}^{K} a_k^2}{2}\right) \). Go to Step 6a.

6a Draw \( a_k \) from \( N\left(\frac{\tau^2}{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{(y_{ik} - \hat{y}_{ik})}{n_k \tau^2 + \sigma^2}}, \frac{\sigma^2}{n_k \tau^2 + \sigma^2}\right) \). Continue to Step 7.

6b Draw \( \eta_k \) from \( N\left(\frac{\theta^2 \xi}{\sum_{k=1}^{K} \eta_k + \sigma^2}, \frac{\sigma^2 \theta^2}{\sum_{k=1}^{K} \eta_k + \sigma^2}\right) \). Go to Step 6c.

6c Draw \( \xi \) from \( N\left(\frac{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \eta_k (y_{ik} - \hat{y}_{ik})}{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \eta_k}, \frac{\sigma^2}{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \eta_k}\right) \) and set \( a_k = \xi \eta_k \). Continue to Step 7.
7 Repeat Steps 2-6 following the different paths in Steps 5 and 6 depending on the prior assumption made on $\tau^2$.

A key component of our strategy is the drawing of 100 posterior draws in Step 3 instead of just 1 posterior draw. This is because existing BART packages initialize all $m T_j$s with a single terminal node. Thus, if we extract the $m T_j$ and $M_j$s from only 1 posterior draw, we will not have allowed the trees to ‘grow’ enough to make the extraction of $\sum_{j=1}^m g(X_{ik}, T_j, M_j)\tilde{Z}_{ik}, \sigma$ useful.

To check whether our strategy for drawing the tree structure and terminal node parameters is valid, we compared the posterior draws of $\sum_{j=1}^m g(X_k, T_j, M_j)$ and $\sigma$ from our strategy with BART implemented using $\text{bart}$ from $\text{BayesTree}$. We generated a continuous outcome dataset using the Friedman regression function (Friedman, 1991)

$$Y_k = 10 \sin(\pi X_{k1}X_{k2}) + 20(X_{k3} - 0.5)^2 + 10X_{k4} + 5X_{k5} + \epsilon_k$$

(9)

where $X_{kq} \sim \text{Uniform}(0, 1)$, $q = 1, \ldots, 5$, $\epsilon_k \sim N(0, 1)$, and $n = 2,000$. We also generated a binary outcome dataset with $n = 2,000$ and the true $G(X_k)$ as

$$G(X_k) = 1.35[\sin(\pi X_{k1}X_{k2}) + 2(X_{k3} - 0.5)^2 - X_{k4} - 0.5X_{k5}]$$

(10)

where $G(X_k)$ is a modification from the true signal in (9) such that the resulting range would mostly lie between -2 to 2. Next, we ran 5 separate Monte Carlo Markov Chains (MCMCs) using our strategy and 5 using $\text{bart}$ from $\text{BayesTree}$. We then compared whether these 10 separate MCMCs of $G(X_k)$ and $\sigma$ converged to the same distribution using their posterior trace plots and GR statistics.

The leftmost plot and middle plot in Figure 2 show the posterior draw of $\sum_{j=1}^m g(X_k, T_j, M_j)$ for $Y_1$ for both continuous and binary outcomes respectively. The blue trace plot belongs to
our simulated BART while the red trace plot belongs to \textit{bart} from \textit{BayesTree}. It is clear that both trace plots converged to the same result with our simulated BART producing a slightly narrower range of posterior draws. The trace plots for other subjects produced similar results and are available upon request. The Gelman-Rubin statistics for both the continuous and binary outcome agree with our trace plots. Figure 3 gives the histogram of the upper 95% Confidence Interval (CI) for the 10 continuous outcome MCMCs (left) as well as the binary outcome (right). Both histograms reported a high frequency of 1 for the the upper 95% CI suggesting good convergence for the posterior draw of $\sum_{j=1}^{m} g(X_k, T_j, M_j)$. This suggests that $\sum_{j=1}^{m} g(X_k, T_j, M_j)$ from our simulated BART draw agrees with $\sum_{j=1}^{m} g(X_k, T_j, M_j)$ from BART.

The rightmost plot Figure 2 shows the posterior draw of $\sigma$ for the 10 MCMCs on the continuous outcome dataset thinned at every 5th observation (Recall $\sigma$ is fixed at 1 for binary outcome BART). We found that our strategy (blue trace plot) did not converge to the same posterior distribution of $\sigma$ as \textit{bart} (red trace plot; GR statistics of 2.85). However, the posterior interval of $\sigma$ from our strategy easily included the true value of $\sigma$. Comparatively, the posterior interval of $\sigma$ produced by the \textit{bart} package did not cover the true value of $\sigma$ at all. This underestimation of $\sigma$ by BART was reported and discussed recently by Pratola (2016). Since our strategy was able to produce posterior draws of $\sigma$ with approximately correct coverage of the true $\sigma$ value, we were not concerned that the posterior distribution of $\sigma$ from our strategy did not agree with that produced by \textit{bart}.  

16
4 Simulation Study

Having shown the validity of our implementation strategy, we now determine the prediction performance, bias, and 95% coverage of riBART compared to BART on a longitudinal dataset with correlated outcomes. To do so, we employed a simulation study. The models we compared were: (I) BART, (II) riBART with $P(\tau^2) \propto 1$ (flat), (III) riBART with half-Cauchy prior on $\tau^2$ (half-Cauchy), and (IV) riBART with $\tau^2 \sim IG(1,1)$ (proper). We quantified prediction performance by using the mean squared error (continuous) and AUC (binary) produced by each model. For the bias and 95% coverage, we investigated the predicted values (continuous), $\hat{Y}_{ik}$, predicted probabilities (binary), $\hat{\pi}_{ik}$, $\sigma$ (continuous), and the within-subject correlation, $\tau$. Here, $\hat{Y}_{ik}$ is the posterior mean of $\sum_{j=1}^{m} g(X_{ik}, T_j, M_j) + a_k$ while $\hat{\pi}_{ik} = \Phi[\hat{G}_a(X_{ik})]$ where $\hat{G}_a(X_{ik})$ is the posterior mean of $G_a(X_{ik})$.

We generated our correlated outcomes dataset as follows. We first generated the predictors as $X_{ikq} \sim \text{Uniform}(0, 1)$, $q = 1, \ldots, 5$. Next, for continuous outcomes, we generated $Y_{ik} = 10\sin(\pi X_{ik1}X_{ik2}) + 20(X_{ik3} - 0.5)^2 + 10X_{ik4} + 5X_{ik5} + a_k + \epsilon_{ik}$ \hspace{1cm} (11)

where $\epsilon_{ik} \overset{i.i.d.}{\sim} N(0, \sigma^2)$ and $a_k \overset{i.i.d.}{\sim} N(0, \tau^2)$. For binary outcomes, we first generated $G_a(X_{ik}) = 1.35[\sin(\pi X_{ik1}X_{ik2}) + 2(X_{ik3} - 0.5)^2 - X_{ik4} - 0.5X_{ik5}] + a_k$ \hspace{1cm} (12)

where $a_k \overset{i.i.d.}{\sim} N(0, \tau^2)$. Then, we generated the binary outcomes $Y_{ik}$ by drawing $Z_{ik} \sim N(G_a(X_{ik}), 1)$ and setting $Y_{ik} = 1$ if $Z_{ik} > 0$, otherwise $Y_{ik} = 0$. We implemented riBART using the strategy outlined in Subsection 3.1. We implemented the BART algorithm required in Step 3 of our strategy using the `bart` function from the `BayesTree R` package.

For the study design, we considered $K = 50$ clusters with $n_k = 5$ observations per cluster (small) and $K = 100$ clusters with $n_k = 20$ observations per cluster (large). We
also considered within subject correlation of $\tau = 0.5$ (small) and $\tau = 1$ (large). This produces eight different simulation scenarios summarized in Table 1. For each simulation, we conducted 1,000 burn ins followed by 5,000 posterior draws. MSE, AUC, bias, and coverage were estimated from 200 simulations for each scenario. All our simulations were done in $R$ 3.1.1 ([R Core Team], 2015).

Figure 4 shows the boxplots of the MSEs and Figure 5 shows the boxplots of the AUCs produced by the simulation scenarios under BART, riBART with half-Cauchy prior on $\tau^2$, riBART with $\tau^2 \sim IG(1, 1)$, and riBART with $P(\tau^2) \propto 1$. Other than the setting where the number of subjects and within subject correlation are small, the MSEs of continuous outcomes riBART under the three $\tau^2$ prior distributions were all smaller compared to BART. In addition, there does not seem to be much difference between the predicted values (continuous) for riBART under the three different $\tau^2$ prior distributions. Similar results were observed for the binary outcomes.

Table 1 shows the mean of the bias and 95% coverage for $\hat{Y}_{ik}$, $\hat{\pi}_{ik}$, $\sigma$, and $\tau$. The bias of $\hat{Y}_{ik}$, $\sigma$, and $\tau$ were largely similar between the three different riBART methods, with the exception that the half-Cauchy prior produced a large bias in $\tau$ when the number of subjects and within subject correlation are small. The 95% coverage of $\tau$ for all 3 riBART methods achieved close to nominal coverage except again, when the number of subjects and within subject correlation are small. For the 95% coverage of $\hat{Y}_{ik}$ and $\sigma$, there were some over coverage for these parameters. BART under the correlated continuous outcomes scenario works well in the estimation of $\hat{Y}_{ik}$ (in terms of bias and 95% coverage) but performed poorly in the estimation of $\sigma$.

Turning to the binary outcomes, we once again note that the bias for $\hat{\pi}_{ik}$ and $\tau$ were
similar for all 3 riBART methods except when the number of subjects and within subject correlation are small. In this scenario, the bias of $\tau$ under $\tau^2 \sim IG(1,1)$ was about 10 times larger compared to $P(\tau^2) \propto 1$ and half-Cauchy prior. For the 95% coverage of binary outcomes, the coverage of $\tau$ reached close to nominal levels for the 3 riBART methods under most scenarios. However, for $\hat{\pi}_{ik}$, there appears to be some under coverage when the number of subjects is small. Although BART under the correlated binary outcomes scenario produces a smaller bias for $\hat{\pi}_{ik}$ in most simulations, the 95% coverage for $\hat{\pi}_{ik}$ is poor ranging from 66-89%.

In summary, there does not seem to be much difference between the use of the 3 different prior distributions on $\tau^2$ for riBART in terms of the prediction performance, bias and 95% coverage. Accurate estimate of $\tau$ also required sufficiently large values of $\tau$ and sufficiently large numbers of observations for each subject. For small $n_k$ and weak within-subject correlation, the researcher may want to run both BART and riBART to determine whether there is substantial improvement in performance produced by riBART over BART.

5 Predicting Driver Stop before Left Turn Execution

5.1 Integrated Vehicle-Based Safety Systems (IVBSS) Study

The dataset we used to develop our prediction model was obtained from the Integrated Vehicle Based Safety System (IVBSS) study conducted by Sayer et al. (2011). This study collected naturalistic driving data from 108 licensed drivers in Michigan between April 2009 and April 2010. In the study, sixteen late-model Honda Accords were fitted with cameras,
recording devices, and several integrated collision warning systems. Each driver used a vehicle for a total of 40 days – 12 days baseline period with IVBSS switched off followed by 28 days with IVBSS activated. Since our objective was to develop a prediction model for human driving behavior, we used the 12 days baseline unsupervised driving data. In total the 108 drivers made 3,795 turns, of which 1,823 were left turns. Each driver took on average of 35 turns, with a range of 8 to 139 turns per driver. This suggests that riBART could potentially improve upon the prediction performance of our model compared to BART.

5.2 Analysis

To begin prediction, we extracted both the speed of the vehicle (in m/s) and the distance traveled (in m) at 10 millisecond intervals starting from 100 meters away from the center of an intersection. To obtain a practical prediction model, we converted the time series of vehicle speeds to a distance series and provided a new distance-varying definition for our binary outcomes of whether a vehicle would stop before executing a left turn in the future. Our outcome was estimating whether a vehicle would stop before executing a left turn, estimated repeatedly at 1 meter intervals before the intersection. For example, if the vehicle’s current location is -45 meters, the outcome is whether the vehicle will stop between -44 and -1 meter. If a vehicle stops and restarts, the outcome is reset: a vehicle that stops at -40 meters and then proceeds through the intersection will have an outcome of 1 (stopping) from -94 to -40 meters, and 0 (not stopping) from -39 to -1 meters.

At any given meter, we could use the full profile of a vehicle’s past speeds as the predictors but these speeds may contain irrelevant information. Thus, we determined a moving window of recent speeds to be used in our prediction model at every meter. Using
a 10-fold cross validation with AUC as the judging criteria and BART as the model, we selected an optimal window length of 6 meters. To further reduce the number of variables to consider in our model, we then used Principal Components Analysis (PCA) to summarize the vehicle speeds in each 6 meter moving window. The first three PC scores explained more than 99% of the variation in the vehicle speed and we found that adding PC scores beyond these did not improve prediction. We included a fourth predictor, the number of times the vehicle has stopped up to the current location. This fourth predictor adjusts for the likely correlation within each turn.

We conducted a preliminary data analysis using logistic regression, BART, and a Super Learner ensemble method (van der Laan and Polley, 2010) that considered elastic net (Friedman et al., 2010), logistic regression, K-Nearest Neighbor, Generalized Additive Models (Hastie and Tibshirani, 1990), mean of the outcomes, and BART. Super Learner and BART had similar prediction performance as measured by AUC, but BART was far more stable.

We fit riBART with a random effect at the driver level which incorporates within-driver correlation. We used the 3 different prior distributions for $\tau$ to check the sensitivity of riBART. For comparison, we also ran BART, which ignores within-driver correlation; linear logistic regression, which ignores within-driver correlation, non-linearity, and complex interactions; and a random intercept linear logistic regression, which incorporates within-driver correlation but ignores non-linearity and complex interactions. For these models, we used the same distance-varying predictors and outcome as riBART. The linear logistic regression was obtained using the \texttt{glm} function in R while the random intercept linear logistic regression was obtained using the \texttt{glmer} function from the \textit{R} package \texttt{lme4}. We also computed
the 95% CI of the AUCs using the method of Hanley and McNeil [1982], which uses a linear approximation of the AUC to the Somer’s D statistic to obtain an estimate of the variance of AUC.

5.3 Results

Figure 6 shows the the estimated intra-class correlation (ICC, \(\frac{\tau^2}{\tau^2 + 1}\)) profile of riBART under \(P(\tau^2) \propto 1\) and half-Cauchy prior on \(\tau^2\), the AUC profiles of riBART under \(P(\tau^2) \propto 1\) and half-Cauchy prior on \(\tau^2\), BART, linear logistic regression, and random intercept linear logistic regression, and the AUC profile difference between riBART versus BART, and riBART versus random intercept linear logistic regression. We omitted the presentation of riBART under IG(1,1) because when the posterior draws of \(\tau\) are small (< 0.5), the posterior draw of \(\tau\) under riBART with IG(1,1) may be positively biased.

The posterior mean profile of ICC (leftmost) was small, between about 0.02 and 0.09, and decreasing as the vehicle approaches the center of an intersection. This suggests firstly that the within subject correlation is small for left turn stops and secondly that as the vehicle approaches the center of the intersection, individual ‘habits’ of the driver do not matter as much compared to when the vehicle is -94m away from the center of an intersection. For the AUC profile (middle), we see some evidence that riBART performs better than BART, and BART in turn performs better than the linear logistic regression model for our dataset. There seems to be no difference in terms of AUC performance between BART and random intercept linear logistic regression. BART produced an AUC estimate of about 0.74 with an estimated 95% CI of (0.72, 0.76) at -94m away from the center of intersection. For both riBART specifications, the AUC was 0.78 [95% C.I. (0.76, 0.80)] at -94m away from
the center of intersection. The difference in AUC profile (rightmost) between riBART and BART remained positive until -10m for both $P(\tau^2) \propto 1$ and half-Cauchy prior on $\tau^2$. For the AUC difference between riBART and random intercept linear logistic regression, it remained positive throughout.

Our results suggest that a substantial improvement in prediction compared to BART can be obtained when we take into account that different drivers have different ‘propensities to stop’ before executing a left turn at an intersection; that is, the inclusion of a random intercept improves prediction performance for our dataset compared to a model without a random intercept. This implies that future development of processing for autonomous vehicles should try to accommodate the similarities of stopping behavior for a given human driver. In addition, devices that are able to transmit information about a driver’s propensity to stop could be installed on vehicles to improve the performance of both the self driving vehicle and human driven vehicle.

6 Discussion

In this paper, we developed a model, riBART, to help engineers developing self driving vehicles predict whether a human-driven vehicle would stop at an intersection before executing a left turn. We achieved this by utilizing the model that did well in our preliminary analyses, BART, and extending it to account for the key feature in our dataset, clustered observations. Although existing methods extending BART to longitudinal datasets were available, our approach was more straightforward and our implementation can be done easily using existing BART software packages. Applying riBART to our dataset, we found substantial
improvements in prediction performance compared to BART, linear logistic regression, and random intercept linear logistic regression. This implies that each driver has an implicit propensity to stop and knowledge of this information can greatly improve the prediction of whether a human-driven vehicle would stop before executing a left turn in addition to the speed of the human-driven vehicle.

When applying to the prediction of binary outcomes in longitudinal settings, riBART will perform better than BART when the number of repeated measurements is large, about 20. This increase in performance would be maximized when the correlation within the repeated measurements is high. Although we proposed a novel idea of how riBART may be implemented without a major re-write of existing BART implementations or implementing riBART from scratch, an issue is the computational burden of our strategy. Because we had to use 100 posterior draws to mimic drawing from the posterior distribution of $(T_1, M_1), \ldots, (T_m, M_m)$, computational efficiency was greatly reduced.

Our proposed model only included a random intercept but, there may be situations where the researcher believes that there may be more complicated linear random effect mechanisms occurring in the real world setting. In our application, estimating a “turn-level” random effect nested within the driver-level random effect could have been done but would be of little value for predicting future turns. However, in other settings, estimating and splitting of these variance components might be useful. Other plausible areas for future research include extending BART to outcomes of other forms, for example, ordinal outcomes or counts.
Supplementary materials

The zip file contains two folders and a portable document format (pdf) file. “Codes” contain the codes we used while the pdf file “Posterior distributions” contain the derivations of the posterior distributions for BART and riBART.

Acknowledgments

This work was supported jointly by Dr. Michael Elliott and an ATLAS Research Excellence Program project awarded to Dr. Carol Flannagan. This work was also funded in part by the Toyota Class Action Settlement Safety Research and Education Program. The conclusions are those of the authors and have not been sponsored, approved, or endorsed by Toyota or plaintiffs’ class counsel. We would like to thank Dr. Jian Kang and Dr. Brisa Sánchez for their valuable suggestions.

References

Albert, J. and Chib, S. (1993). Bayesian Analysis of Binary and Polychotomous Response Data. *Journal of the American Statistical Association* **88**, 669–679.

Albert, J. and Chib, S. (1996). Bayesian modeling of binary repeated measures data with application to crossover trials. In Bayesian Biostatistics, D. A. Berry and D. K. Stangl, eds. New York: Marcel Dekker.
Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and regression Trees*. Wadsworth, Belmont, CA.

Chipman, H., George, E., and McCulloch, R. (1998). Bayesian CART Model Search. *Journal of the American Statistical Association* **93**, 935–948.

Chipman, H., George, E., and McCulloch, R. (2010). BART: Bayesian Additive Regression Trees. *The Annals of Applied Statistics* **4**, 266–298.

Chipman, H., McCulloch, R., and Dorie, V. (2015). Discrete Bayesian Additive Regression Trees Sampler, [online], Available at https://github.com/vdorie/dbarts

Davies, A. (2015). GM Has ‘Aggressive’ Plans for Self-Driving Cars, *Wired Magazine* [online], Available at https://www.wired.com/2015/10/gm-has-aggressive-plans-for-self-driving-cars/

Dorie, V., Harada, M., Carnegie, N., and Hill, J. (2016). A flexible, interpretable framework for assessing sensitivity to unmeasured confounding. *Statistics in Medicine* **35**(20), 3453–3470.

Franke, R. (1982). Smooth interpolation of scattered data by local thin plate splines. *Computers and Mathematics with Applications* **8**, 273–281.

Friedman, J. (1991). Multivariate Adaptive Regression Splines (with discussion and a rejoinder by the author). *The Annals of Statistics* **19**, 1–67.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* **33**, 1–22.
Gamermann, A. (2000). Support vector machine learning algorithm and transduction. *Computational Statistics* **5**, 31–39.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (Comment on Article by Browne and Draper). *Bayesian Analysis* **1**, 515–534.

Google (2015). What were up to [online], Available at [http://www.google.com/selfdrivingcar/](http://www.google.com/selfdrivingcar/)

Hanley, J. and McNeil, B. (1982). The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology* **143**, 29–36.

Hastie, T. and Tibshirani, R. (1990). *Generalized additive models*. CRC Press: Boca Raton, FL.

Kapelner, A. and Bleich, J. (2016). bartMachine: Machine Learning with Bayesian Additive Regression Trees. *Journal of Statistical Software* **70**, 1–40.

Low-Kam, C., Telesca, D., Ji, Z., Zhang, H., Xia, T., Zink, J., and Nel, A. (2015). A Bayesian regression tree approach to identify the effect of nanoparticles’ properties on toxicity profiles. *The Annals of Applied Statistics* **9**, 383–401.

Mchugh, M. (2015). Teslas Cars Now Drive Themselves, Kinda, *Wired Magazine* [online], Available at [http://www.wired.com/2015/10/tesla-self-driving-over-air-update-live/](http://www.wired.com/2015/10/tesla-self-driving-over-air-update-live/)

Pratola, M. (2016). Efficient Metropolis-Hastings Proposal Mechanisms for Bayesian Regression Tree Models. *Bayesian Analysis* **11**(3), 885–911.
R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rässler, S. (2002). *Statistical matching: A frequentist theory, practical applications and alternative bayesian approaches*. Lecture Notes in Statistics, Springer Verlag, New York.

Ruppert, D., Wand, M., and Carrol, R. (2003). *Semiparametric regression*. Cambridge University Press: Cambridge, UK.

Sayer, J., Bogard, S., Buonarosa, M., LeBlanc, D., Funkhouser, D., Bao, S., Blanke-spool, A., and Winkler, C. (2011). Integrated Vehicle-Based Safety Systems Light-Vehicle Field Operational Test Key Findings Report DOT HS 811 416, [online], Available at http://umtri.umich.edu/content/IVBSS_LV_Key_Findings.pdf

Smith, D., Bailey, T. C., and Munford, A. (1993). Robust classification of artificial neural networks. *Statistics and Computing* **3**, 71–81.

Tanner, M. and Wong, W. (1987). The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association* **82**, 528–540.

van der Laan, M. and Polley, E. C. (2010). Super Learner in Prediction. *U.C. Berkeley Division of Biostatistics Working Paper Series Working Paper 266*, http://biostats.bepress.com/ucbbiostat/paper266

Zhang, S., Shih, Y., and Müller, P. (2007). A Spatially-adjusted Bayesian Additive Regression Tree Model to Merge Two Datasets. *Bayesian Analysis* **2**, 611–634.
Figure 1: Example of a regression tree where $\mu_{ij}$ is the mean of the $i^{th}$ node for the $j^{th}$ regression tree.

$$X_2 < 100$$

$\mu_{1j} = 1.19$

$$X_4 < 200$$

$$X_3 < 150$$

$\mu_{2j} = 2.37$

$$X_5 < 50$$

$\mu_{3j} = 2.93$

$\mu_{5j} = 4.5$

$\mu_{A_j} = 4$
Table 1: Bias and coverage of $\hat{Y}_{ik}$, $\hat{\pi}_{ik}$, $\sigma \equiv 1$, and $\tau$ for BART, riBART with $P(\tau^2) \propto 1$ (flat), half-Cauchy prior on $\tau^2$, and $\tau^2 \sim IG(1, 1)$ (proper).

| Scenario 1: continuous, $n_k = 5$, $K = 50$, $\tau = 1$ | Scenario 5: binary, $n_k = 5$, $K = 50$, $\tau = 1$ |
|---------------------------------|---------------------------------|
|                                | $\hat{Y}_{ik}$                   | $\hat{\pi}_{ik}$               |
|                                | Bias ($10^{-4}$) | Coverage (%) | Bias Coverage (%) | Bias Coverage (%) | Bias Coverage (%) | Bias Coverage (%) |
| BART                           | 1.57                         | 100          | 0.16            | 63               | -               | -                |
| Flat                           | 2.06                         | 100          | -0.05           | 99               | -0.02           | 95               |
| Half-Cauchy                    | 1.54                         | 100          | -0.05           | 99               | -0.04           | 95               |
| Proper                         | 0.96                         | 100          | -0.05           | 99               | -0.05           | 96               |
|                                | 5.39                         | 66           | -               | -                |
| Flat                           | 4.29                         | 91           | -0.005          | 96               |
| Half-Cauchy                    | 4.44                         | 91           | -0.05           | 94               |
| Proper                         | 4.22                         | 91           | -0.06           | 96               |

| Scenario 2: continuous, $n_k = 20$, $K = 100$, $\tau = 1$ | Scenario 6: binary, $n_k = 20$, $K = 100$, $\tau = 1$ |
|---------------------------------|---------------------------------|
|                                | $\hat{Y}_{ik}$                   | $\hat{\pi}_{ik}$               |
|                                | Bias ($10^{-4}$) | Coverage (%) | Bias Coverage (%) | Bias Coverage (%) | Bias Coverage (%) | Bias Coverage (%) |
| BART                           | 0.57                         | 98           | 0.35            | 0                | -               | -                |
| Flat                           | -0.02                        | 99           | 0.0003          | 98               | 0.01           | 94               |
| Half-Cauchy                    | 0.29                         | 99           | 0.0002          | 98               | 0.01           | 94               |
| Proper                         | -0.39                        | 99           | 0.0002          | 98               | 0.001          | 95               |
|                                | -7.72                        | 45           | -               | -                |
| Flat                           | -9.10                        | 95           | 0.006           | 97               |
| Half-Cauchy                    | -8.92                        | 95           | -0.002          | 97               |
| Proper                         | -8.98                        | 95           | -0.01           | 96               |

| Scenario 3: continuous, $n_k = 5$, $K = 50$, $\tau = 0.5$ | Scenario 7: binary, $n_k = 5$, $K = 50$, $\tau = 0.5$ |
|---------------------------------|---------------------------------|
|                                | $\hat{Y}_{ik}$                   | $\hat{\pi}_{ik}$               |
|                                | Bias ($10^{-4}$) | Coverage (%) | Bias Coverage (%) | Bias Coverage (%) | Bias Coverage (%) | Bias Coverage (%) |
| BART                           | 1.44                         | 100          | -0.14           | 64               | -               | -                |
| Flat                           | 1.20                         | 100          | -0.07           | 96               | -0.05           | 99               |
| Half-Cauchy                    | 1.05                         | 100          | -0.06           | 97               | -0.13           | 95               |
| Proper                         | 0.71                         | 100          | -0.08           | 94               | 0.08           | 98               |
|                                | -7.24                        | 89           | -               | -                |
| Flat                           | -11.55                       | 90           | 0.01            | 96               |
| Half-Cauchy                    | -11.08                       | 88           | -0.06           | 93               |
| Proper                         | -10.27                       | 93           | 0.11            | 95               |

| Scenario 4: continuous, $n_k = 20$, $K = 100$, $\tau = 0.5$ | Scenario 8: binary, $n_k = 20$, $K = 100$, $\tau = 0.5$ |
|---------------------------------|---------------------------------|
|                                | $\hat{Y}_{ik}$                   | $\hat{\pi}_{ik}$               |
|                                | Bias ($10^{-4}$) | Coverage (%) | Bias Coverage (%) | Bias Coverage (%) | Bias Coverage (%) | Bias Coverage (%) |
| BART                           | 0.28                         | 98           | 0.06            | 13               | -               | -                |
| Flat                           | -0.35                        | 99           | 0.0001         | 98               | 0.005          | 95               |
| Half-Cauchy                    | -0.20                        | 99           | 0.0004         | 98               | 0.001          | 95               |
| Proper                         | -0.59                        | 99           | 0.0003         | 98               | 0.02           | 94               |
|                                | -0.29                        | 75           | -               | -                |
| Flat                           | -1.12                        | 95           | -0.002         | 95               |
| Half-Cauchy                    | -1.20                        | 94           | -0.008         | 95               |
| Proper                         | -1.05                        | 95           | 0.02           | 96               |

Figure 2: Comparing 10 posterior draws of $\sum_{j=1}^{m} g(X_k, T_j, M_j)$ for subject 1 and $\sigma = 1$ with BART (red) and simulated BART using degrees of freedom set at 100,000 (blue).

From left to right: continuous outcome, binary outcome, $\sigma$, thinned using every 5th observation.
Figure 3: Histogram of the upper 95% confidence interval Gelman-Rubin (GR) combining 5 MCMC chains from $R$ bart package BayesTree and 5 MCMC chains using our proposed method that separates draws of $\sigma$. Continuous outcomes on the left and binary outcomes on the right.
Figure 4: Boxplots of mean squared error (MSE) for continuous outcomes produced by BART, riBART with $P(\tau^2) \propto 1$, half-Cauchy prior on $\tau^2$, and $\tau^2 \sim IG(1, 1)$.

(a) $n_k = 5$, $K = 50$, $\tau = 1$

(b) $n_k = 20$, $K = 100$, $\tau = 1$

(c) $n_k = 5$, $K = 50$, $\tau = 0.5$

(d) $n_k = 20$, $K = 100$, $\tau = 0.5$
Figure 5: Boxplots of area under the receiver operating characteristic curve (AUC) for binary outcomes produced by BART, riBART with \( P(\tau^2) \propto 1 \), half-Cauchy prior on \( \tau^2 \), and \( \tau^2 \sim IG(1, 1) \).

(a) \( n_k = 5, K = 50, \tau = 1 \)  
(b) \( n_k = 20, K = 100, \tau = 1 \)

(c) \( n_k = 5, K = 50, \tau = 0.5 \)  
(d) \( n_k = 20, K = 100, \tau = 0.5 \)
Figure 6: Comparing results between riBART under $P(\tau^2) \propto 1$ or half-Cauchy prior on $\tau^2$, BART, linear logistic regression, and random intercept logistic regression for (a) intra-class correlation (ICC) profile, (b) area under the receiver operating characteristic curve (AUC) profile, and (c) AUC difference profile for the prediction of driver stop before executing a left turn.

(a) ICC

(b) AUC

(c) AUC difference