Statistics and Its Interface Volume 11 (2018) 557–572

Predicting human-driving behavior to help driverless vehicles drive: random intercept Bayesian additive regression trees

Yaoyuan Vincent Tan*, Carol A. C. Flannagan, and Michael R. Elliott

The development of driverless vehicles has spurred the need to predict human driving behavior to facilitate interaction between driverless and human-driven vehicles. This interaction is possible when both driverless and human-driven vehicles are connected through vehicle-to-vehicle communication. 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. To handle the possible non-linear effects and interactions, we used Bayesian additive regression trees (BART). 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 lack readily available software and can only be applied to clustered continuous outcomes. We extend BART to handle correlated binary observations by adding a random intercept and used a simulation study to investigate its properties. We then successfully implemented our proposed model to our clustered dataset and found substantial improvements in prediction performance compared to BART, BART adjusted for driver level effects, random intercept linear logistic regression, and linear logistic regression.

Keywords and phrases: 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. A number of companies have deployed or announced plans for deployment of such vehicles [1, 2, 3]. A major hurdle for self-driving vehicles 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.

Currently, driverless vehicles developed generally use onboard sensors to gather data from their surrounding environment to make driving decisions. We envision in the future that vehicles (both human driven and driverless) would be connected such that a driving intent model could first be evaluated on the human driver's vehicle and subsequently "communicated" to the driverless vehicle enabling it to make a better driving decision. Such vehicle-to-vehicle communication would become increasingly available as technology improves resulting in a connected environment. Under such a connected environment, developing a good prediction model for human driving behavior would make sense especially when the driving pattern of a human driven vehicle depends heavily on the unique tendencies of the human driver.

Building a prediction model that addresses all or most of the human driving behavior and driving intent is a massive and complex task. To keep this paper concise, we focus on the the development of a prediction model for 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, the US), 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. Placing this prediction scenario in the context of a connected environment, the driverless vehicle will be evaluating data from the human-driven vehicle, supplied from an adapted version of existing "black-box" technology that would broadcast speed and location information to driverless vehicles. The connected driverless vehicle would then

arXiv: 1609.07464

*Corresponding author.

combine this transmitted information together with the data it has gathered from its surrounding environment to make a driving decision.

To develop such a prediction model, we used a naturalistic driving study, the Integrated Vehicle Based Safety System (IVBSS) study [4]. 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 [5, 6]. Modeling of non-linear interactions between two or more predictors using thin-plate splines [7] 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; 8] as well as more sophisticated machine learning techniques such as artificial neural networks [ANN; 9] and support vector machines [SVM; 10] 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, we prefer regression methods that are able to account for non-linear main and multiple-way interaction effects. Bayesian additive regression trees [BART; 11] is one such model which allows flexible estiamtion of non-linear main and multiple-way interaction effects without much input from the researcher. Hence, we employed BART to predict whether a human-driven vehicle would stop before executing a left turn at an intersection. However, BART was designed for independent subjects, but we would like to evaluate the tendencies of each driver and decide whether including their tendency would improve the prediction of whether a human-driven vehicle would stop before executing a left turn. We are aware of two papers that extended BART to handle longitudinal or clustered observations: [12] used a spatial random intercept BART to merge two datasets, and [13] did so in a dose-finding toxicity study. [12] developed an imputation model for a statistical matching problem [14] that used BART with a conditional autoregressive distribution for the random intercept. Since the correlation our dataset was induced by repeated measurements and not spatial effects, the distribution [12] placed on the random intercept may not be appropriate. Moreover, they did not discuss how their model could be extended to clustered binary outcomes. [13] 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 clustered binary outcomes, we propose an extension of BART to account for longitudinal binary observations. Our proposed method accounts for clustering by adding a random intercept to BART and we call this random intercept BART (riBART). We proceed by first providing a review of BART in the next section followed by a discussion of how we extended BART to riBART in Section 3. In Section 4, we use a simulation study to compare the performance of riBART against BART, fixed effects BART, and linear regression models when applied to clustered datasets. We implement riBART on our dataset and compare its prediction performance with BART, fixed effects BART, random intercept linear logistic regression, and multiple linear logistic regression in Section 5. 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 $\mathbf{X}_k = (X_{k1}, \ldots, X_{kp})^T$ for $k = 1, \ldots, n$ subjects. BART models the outcome as

$$(1) \qquad Y_k = \sum_{j=1}^{m} g(\mathbf{X}_k, T_j, \mathbf{M}_j) + \epsilon_k \quad \epsilon_k \overset{i.i.d.}{\sim} N(0, \sigma^2)$$

where T_j is the j^{th} binary tree structure and $\mathbf{M}_j = (\mu_{1j}, \ldots, \mu_{b_j j})^T$ is the set of b_j terminal node parameters associated with tree structure T_j [11]. $g(\mathbf{X}_k, T_j, \mathbf{M}_j)$ can be viewed as the j^{th} function that assigns the mean μ_{ij} to the k^{th} outcome, Y_k. Typically, the number of trees m is fixed and no prior distribution is placed on m. [11] suggested setting $m = 200$ as this performs well in many situations. Alternatively, cross-validation could be used to determine m [11].

The joint prior distribution for Eq. (1) is $P[(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m), \sigma]$. Note that by the independence of ϵ_k and $(T_j, \mathbf{M}_j)$ as well as the independence between all m tree structures and terminal node parameters, the joint prior distribution $P[(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m), \sigma]$ can be decomposed as

$$
\begin{aligned}
P[(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m), \sigma] &= [\prod_{j=1}^{m} P(T_j, \mathbf{M}_j)]P(\sigma) \\
&= [\prod_{j=1}^{m} P(\mathbf{M}_j|T_j)P(T_j)]P(\sigma) \\
&= [\prod_{j=1}^{m}\{\prod_{i=1}^{b_j} P(\mu_{ij}|T_j)\}P(T_j)] \\
&\quad \times P(\sigma).
\end{aligned}
$$

where $i = 1, \ldots, b_j$ indexes the terminal node parameters in tree j. This implies that we need to assign priors to T_j, $\mu_{ij}|T_j$, and σ in order to obtain the posterior distributions of T_j, μ_{ij}, and σ. [11] suggested the following prior distributions on $\mu_{ij}|T_j$ and σ:

$$
\begin{aligned}
\mu_{ij}|T_j &\sim N(\mu_\mu, \sigma_\mu^2), \\
\sigma^2 &\sim IG(\frac{\nu}{2}, \frac{\nu\lambda}{2}),
\end{aligned}
$$

where $IG(\alpha, \beta)$ is the inverse gamma distribution with shape parameter α and rate parameter β. 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 α controls how likely a terminal node in the tree would split, with a smaller α implying lesser likelihood a terminal node would split, and β controls the number of terminal nodes, and a larger β decreasing 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. [11] 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 [15].

In [11], $\alpha = 0.95$ and $\beta = 2$. For μ_μ and σ_μ, they are set such that $N(m\mu_\mu, m\sigma_\mu^2)$ assigns high probability to the interval $(\min_k(Y_k), \max_k(Y_k))$. This can be achieved by defining v such that $\min_k(Y_k) = m\mu_\mu - v\sqrt{m}\sigma_\mu$ and $\max_k(Y_k) = m\mu_\mu + v\sqrt{m}\sigma_\mu$. For convenience when implementing the posterior draws of T_j and μ_{ij}, [11] suggested transforming the observed Y_k to $\tilde{Y}_k = \frac{Y_k - \frac{\min_k(Y_k)+\max_k(Y_k)}{2}}{\max_k(Y_k)-\min_k(Y_k)}$, and then treating $\tilde{Y}_k$ as the outcome. This has the effect of allowing the hyperparameter of μ_μ to be set as $\mu_\mu = 0$ and σ_μ to be set as $\sigma_\mu = \frac{0.5}{v\sqrt{m}}$ where v is to be chosen. For $v = 2$, $N(m\mu_\mu, m\sigma_\mu^2)$ assigns a prior probability of 0.95 to the interval $(\min_k(Y), \max_k(Y))$ and is the suggested value. Finally, for ν and λ, [11] suggested setting $\nu = 3$ and λ 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 $\mathbf{X}_k$ as the covariates.

This setup induces the posterior distribution $P[(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m), \sigma|Y_k]$ which can be simplified to two major posterior draws using Gibbs sampling. First, draw m successive

$$
P[(T_j, \mathbf{M}_j)|T_{(j)}, \mathbf{M}_{(j)}, Y_k, \sigma] \tag{2}
$$

for $j = 1, \ldots, m$, where $T_{(j)}$ and $\mathbf{M}_{(j)}$ consist of all the tree structures and terminal nodes except for the j^{th} tree structure and terminal node; and then, draw $P[\sigma|(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m), Y_k]$.

To obtain a draw from Eq. (2), note that this distribution depends on $(T_{(j)}, \mathbf{M}_{(j)}, Y_k, \sigma)$ through

$$
R_{kj} = Y_k - \sum_{w \neq j} g(\mathbf{X}_k, T_w, \mathbf{M}_w), \tag{3}
$$

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

$$
P[(T_j, \mathbf{M}_j)|R_{kj}, \sigma]. \tag{4}
$$

We can obtain a draw from Eq. (4) by first drawing from $P(T_j|R_{kj}, \sigma)$ using a Metropolis-Hastings (MH) algorithm outlined in [16]. A new tree T_j^* can be proposed given the previous tree T_j by four steps: (i) grow, where a terminal node is split into two new child nodes; (ii) prune, where two terminal child nodes immediately under the same non-terminal node is combined together such that their parent non-terminal node becomes a terminal node; (iii) swap, where the splitting criteria of two non-terminal nodes are swapped; (iv) change, where the splitting criteria of a single non-terminal node is changed. Once we draw $P(T_j|R_{kj}, \sigma)$, we then draw $P(\mu_{ij}|T_j, R_{kj}, \sigma) \sim N(\frac{\sigma_\mu^2 \sum_i^{n_i} r_{ij} + \sigma^2\mu_\mu}{n_i\sigma_\mu^2 + \sigma^2}, \frac{\sigma^2\sigma_\mu^2}{n_i\sigma_\mu^2 + \sigma^2})$, where r_{ij} is the subset of elements in R_{kj} allocated to the terminal node with parameter μ_{ij} and n_i is the number of r_{ij}s in R_{kj} allocated to μ_{ij}. Note that $\mu_\mu = 0$ after transformation. Complete details for the derivation of $P(\mu_{ij}|T_j, R_{kj}, \sigma)$ and $P[\sigma|(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m), Y_k]$ are provided in the supplementary materials available online. Explicit MH algorithm details for Eq. (4) can be found in Appendix A of [15].

2.2 Binary outcomes

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

$$G(\mathbf{X}_k) = \sum_{j=1}^{m} g(\mathbf{X}_k, T_j, \mathbf{M}_j). \tag{5}$$

Using the probit formulation, the binary outcomes Y_k can be linked to Eq. (5) using $P(Y_k = 1|\mathbf{X}_k) = \Phi[G(\mathbf{X}_k)]$ where $\Phi[.]$ is the cumulative density function of a standard normal distribution. This formulation implicitly assumes that $\sigma \equiv 1$. 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 $\mu_{ij}|T_j$. [11] assumes that priors for T_j and μ_{ij} as well as the hyperparameters for α and β are the same as BART for continuous outcomes. However, for the hyperparameters of μ_μ and σ_μ, [11] suggested that μ_μ and σ_μ should be chosen such that $G(\mathbf{X}_k)$ 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, [11] suggested $v = 2$.

To draw from the posterior distribution $P[(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m)|Y_k]$, [11] proposed the use of data augmentation [17, 18]. This method proceeds by first generating a latent variable Z_k according to

$$(Z_k|Y_k = 1, \mathbf{X}_k) \sim N_{(0,\infty)}(G(\mathbf{X}_k), 1)$$
$$(Z_k|Y_k = 0, \mathbf{X}_k) \sim N_{(-\infty,0)}(G(\mathbf{X}_k), 1),$$

where $N_{(a,b)}(\mu, \sigma^2)$ is the truncated normal distribution with mean μ and variance σ^2 truncated to the range (a, b). Once Z_k is drawn, $P[(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m)|Z_k]$ is drawn next as in Eq. (2) to Eq. (4) with the latent variables Z_k replacing Y_k in Eq. (2) and σ fixed at 1. Note that at each iteration, $G(\mathbf{X}_k)$ will be updated with the new $(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m)$ draws from $P[(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m)|Z_k]$ so that an updated draw of the latent variable Z_k can be obtained.

3. RANDOM INTERCEPT BART

3.1 Continuous outcomes

We now extend BART to account for repeated measurements. We start with the clustered continuous outcomes. We introduce to Eq. (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, Eq. (1) becomes

$$Y_{ik} = \sum_{j=1}^{m} g(\mathbf{X}_{ik}, T_j, \mathbf{M}_j) + a_k + \epsilon_{ik}, \tag{6}$$

where $\epsilon_{ik} \overset{i.i.d.}{\sim} N(0, \sigma^2)$, $a_k \overset{i.i.d.}{\sim} N(0, \tau^2)$, and $a_k \perp \epsilon_{ik}$. We decompose the joint prior distribution (assuming σ^2 and τ^2 are a priori independent) as

$$P[(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m), \sigma, \tau] = [\prod_{j=1}^{m} \{\prod_{l=1}^{b_j} P(\mu_{lj}|T_j)\} P(T_j)] \times P(\sigma)P(\tau).$$

Next, we place the same prior distributions as the independent BART model for T_j, $\mu_{lj}|T_j$ (this is μ_{ij} for the independent BART model), and σ^2. The prior distribution of τ^2 could be set as $\sim IG(1, 1)$ although other specifications are definitely possible. We explore some alternatives in our supplementary materials available online. We use the same hyperparameter values for α, β, μ_μ, and ν that [11] suggested for the independent BART model. For σ_μ, we found that $\sigma_\mu = \frac{1.96}{v\sqrt{m}}$ worked better for reasons we shall discuss later in this section. For λ, we first estimated the outcomes Y_{ik} using multivariate adaptive regression splines [MARS; 19] with $\mathbf{X}_k$ as the predictors. 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 σ^2 using $s^{(0)2} = \frac{\sum_{k=1}^{K}\sum_{i=1}^{n_k}(Y_{ik} - \hat{Y}_{ik}^{(0)} - \hat{a}_k^{(0)})^2}{N - N(1 - \sqrt{\frac{RSS}{GCV \times N}})}$, where $N = \sum_{k=1}^{K} n_k$, RSS and GCV are the residual sum of squares and generalized cross-validation value from MARS respectively, and $N(1 - \sqrt{\frac{RSS}{GCV \times N}})$ is the effective number of parameters in MARS. Then λ can be set as the value such that $P(\sigma^2 < s^{(0)2}; \nu, \lambda) = 0.9$. We call this model the random intercept BART (riBART).

To draw from the posterior distribution of riBART, we employ a Metropolis within Gibbs procedure. We first draw the Gibbs sample of σ, τ, and a_k separately from their respective posterior distribution. Then, using the updated a_k, we obtain $\tilde{Y}_{ik} = Y_{ik} - a_k$. Now $\tilde{Y}_{ik}|\mathbf{X}_k$ can be viewed as a BART model. The idea of viewing $\tilde{Y}_{ik}|\mathbf{X}_k$ as a BART model has been discussed in [12] and [20]. To allow for convenient implementation of the posterior draws of T_j and $\mu_{lj}|T_j$, we transform the outcomes $\tilde{Y}_{ik}$ to $\check{Y}_{ik} = \frac{(2 \times 1.96)[\tilde{Y}_{ik} - \frac{\min_{i,k}(\tilde{Y}_{ik}) + \max_{i,k}(\tilde{Y}_{ik})}{2}]}{\max_{i,k}(\tilde{Y}_{ik}) - \min_{i,k}(\tilde{Y}_{ik})}$. This transformation produced posterior draws for σ and τ with better repeated sampling properties across the range of our simulation studies compared to the usual transformation employed in BART, and suggests setting $\sigma_\mu = \frac{1.96}{2\sqrt{m}}$ so that $(\min_{i,k}(\tilde{Y}_{ik}), \max_{i,k}(\tilde{Y}_{ik}))$ has a prior probability of 0.95. We suspect this transformation produces better repeated sampling properties for the posterior draws of σ and τ because it controls the range of values $\check{Y}_{ik}$ would vary in. Further investigation beyond the scope of this paper is needed in order to determine why this is the case. After obtaining $\check{Y}_{ik}$, we use $\check{Y}_{ik}$ as the outcome in the BART algorithm to obtain the posterior distribution of T_j.

In our implementation, we employed the grow and prune steps for the proposal of a new tree T_j^* for computational ease. Given T_j, we then draw μ_{lj}. Derivation of the Gibbs sampling distributions of σ, a_k, and τ are provided in the supplementary materials available online.

3.2 Binary outcomes

Extending riBART to binary outcomes proceed in a similar fashion. We add a_k to Eq. (5) to obtain

$$G_a(\mathbf{X}_{ik}) = \sum_{j=1}^{m} g(\mathbf{X}_{ik}, T_j, \mathbf{M}_j) + a_k. \tag{7}$$

We once again assume $a_k \sim N(0, \tau^2)$. To link the sum of trees to the binary outcomes Y_{ik}, we use the probit link and write $P(Y_{ik} = 1|\mathbf{X}_{ik}) = \Phi[G_a(\mathbf{X}_{ik})]$. We suggest prior distributions similar to the continuous outcomes riBART for T_j, μ_{lj}, and τ^2. The same hyperparameters in BART for binary outcome can be used for α, β, μ_μ, and σ_μ. To obtain the posterior draws of T_j, $\mathbf{M}_j$, a_k, and τ^2, we employ the data augmentation method suggested by [21]. First, we draw a latent variable Z_{ik} according to

$$\begin{aligned}(Z_{ik}|Y_{ik} = 1, \mathbf{X}_{ik}) &\sim N_{(0,\infty)}(G_a(\mathbf{X}_{ik}), 1)\\ (Z_{ik}|Y_{ik} = 0, \mathbf{X}_{ik}) &\sim N_{(-\infty,0)}(G_a(X_{ik}), 1).\end{aligned}$$

We then draw τ followed by a_k. Next, we remove a_k from Z_{ik} to obtain $\tilde{Z}_{ik} = Z_{ik} - a_k$. $\tilde{Z}_{ik}|\mathbf{X}_{ik}$ can now be viewed as a continuous BART model and the usual BART algorithm can be applied with σ fixed at 1. In our implementation, we employed a further transformation of $\tilde{Z}_{ik}$ to $\check{Z}_{ik} = \frac{6[\tilde{Z}_{ik} - \frac{\min_{i,k}(\tilde{Z}_{ik}) + \max_{i,k}(\tilde{Z}_{ik})}{2}]}{\max_{i,k}(\tilde{Z}_{ik}) - \min_{i,k}(\tilde{Z}_{ik})}$. This keeps $\check{Z}_{ik}$ within the range of $(-3, 3)$, which we found produces posterior draws for τ with better repeated sampling properties across the range of our simulation studies. The posterior draw is then completed by updating Z_{ik} using the most recent posterior draws of $(T_1, \mathbf{M}_1), \ldots, (T_m, \mathbf{M}_m)$, and a_k.

4. SIMULATION STUDY

We conducted a simulation study to determine the in-sample performance of riBART compared to three alternative methods on a longitudinal dataset with correlated outcomes. The methods we considered were: (I) BART, (II) riBART, (III) fixed effects BART where variables indicating which row belonged to which subject was added as a predictor in BART, and (IV) multiple linear regression (MLR) for continuous outcomes or multiple linear logistic regression (MLLR) for binary outcomes. We focused on the prediction performance of the models by using the mean squared error (MSE; continuous) and area under the receiver operating characteristic curve (AUC; binary) produced by each model. In addition, we investigated the bias, root mean squared error (RMSE), 95% coverage, and average 95% credible interval length (AIL) of $\sum_{j=1}^{m} g(\mathbf{X}_{ik}, T_j, \mathbf{M}_j) + a_k$ abbreviated as $g(x) + a_k$ and σ (for continuous correlated outcomes only).

We generated our correlated outcomes dataset by first drawing the predictors using $X_{ikq} \overset{i.i.d.}{\sim} \text{Uniform}(0, 1)$, $q = 1, \ldots, 10$. For continuous outcomes, we generated

$$\begin{aligned}Y_{ik} &= 10\sin(\pi X_{ik1}X_{ik2}) + 20(X_{ik3} - 0.5)^2 + 10X_{ik4}\\ &\quad + 5X_{ik5} + a_k + \epsilon_{ik}\end{aligned} \tag{8}$$

where $\epsilon_{ik} \overset{i.i.d.}{\sim} N(0, \sigma^2)$, $a_k \overset{i.i.d.}{\sim} N(0, \tau^2)$, and $a_k \perp \epsilon_{ik}$. For binary outcomes, we first generated

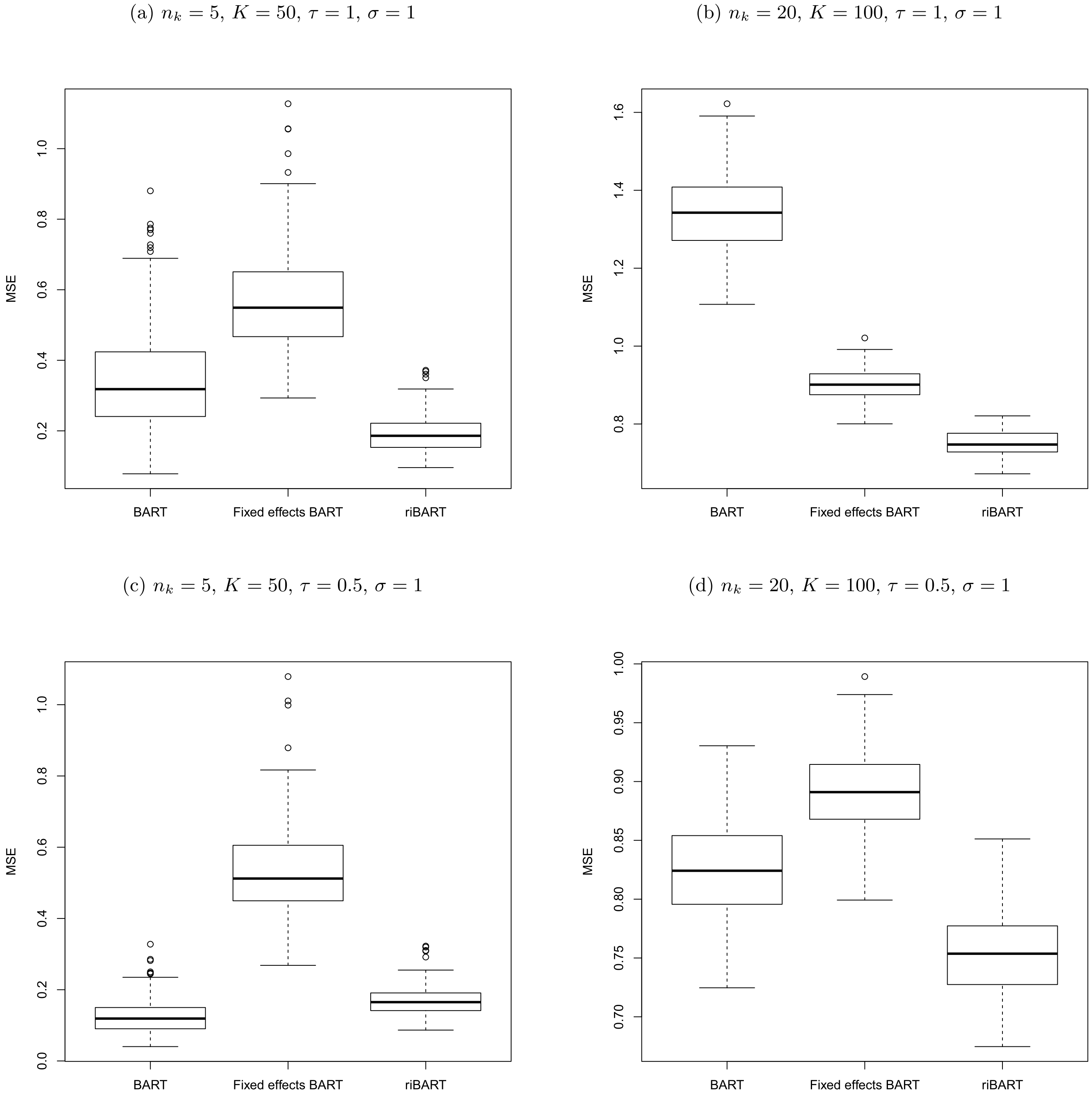
$$\begin{aligned}G_a(X_{ik}) &= 1.35[\sin(\pi X_{ik1}X_{ik2}) + 2(X_{ik3} - 0.5)^2]\\ &\quad - 1.35X_{ik4} - 0.675X_{ik5} + a_k\end{aligned} \tag{9}$$

where $a_k \overset{i.i.d.}{\sim} N(0, \tau^2)$. Then, we generated the binary outcomes Y_{ik} by drawing $Z_{ik} \sim N(G_a(\mathbf{X}_{ik}), 1)$ and setting $Y_{ik} = 1$ if $Z_{ik} > 0$, otherwise $Y_{ik} = 0$. Eq. (8) and Eq. (9) suggest that only the first 5 predictors were important for prediction. The rest of the predictors were “junk” variables.

For the study design, we considered $K = 50$ clusters with $n_k = 5$ observations per cluster and $K = 100$ clusters with $n_k = 20$ observations per cluster. We also considered $\tau = 0.5$ and $\tau = 1$. This produces eight different simulation scenarios summarized in Tables 1 and 2. For each simulation, we conducted 1,000 burn ins followed by 5,000 posterior draws. Bias, RMSE, 95% coverage, AIL, MSE, and AUC were estimated from 200 simulations for each scenario. All our simulations were done in *R 3.1.1* [22].

Figure 1 shows the boxplots of the MSEs for scenarios 1 to 4 while Figure 2 shows the boxplots of the AUCs produced for scenarios 5 to 8. For Figure 1, because the boxplots of the MSE for MLR were much larger compared to the rest of the methods, these boxplots were not presented in the manuscript. Interested readers may refer to our supplementary materials available online for the graphs including MLR results. For continuous correlated outcomes, riBART produces a clear advantage compared to BART and fixed effects BART when $K = 100$, $n_k = 20$, and $\tau = 1$. In other simulation scenarios, riBART does not seem to produce lower MSEs compared to BART and fixed effects BART. For binary correlated outcomes, the advantage of BART in terms of producing a better AUC is more apparent. We observed from Figure 2 that riBART produces the higher AUC compared to BART, fixed effects BART, and MLLR in all our simulation scenarios. This suggests that for continuous correlated outcomes, riBART may not yield an obvious prediction advantage except when the values of K, n_k, and τ are large. However, for binary correlated outcomes, riBART would produce an obvious prediction advantage regardless of K, n_k, and τ.

In terms of the inference for the parameters $\sum_{j=1}^{m} g(\mathbf{X}_{ik}, T_j, \mathbf{M}_j) + a_k$ and σ, Table 1 suggests that for


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

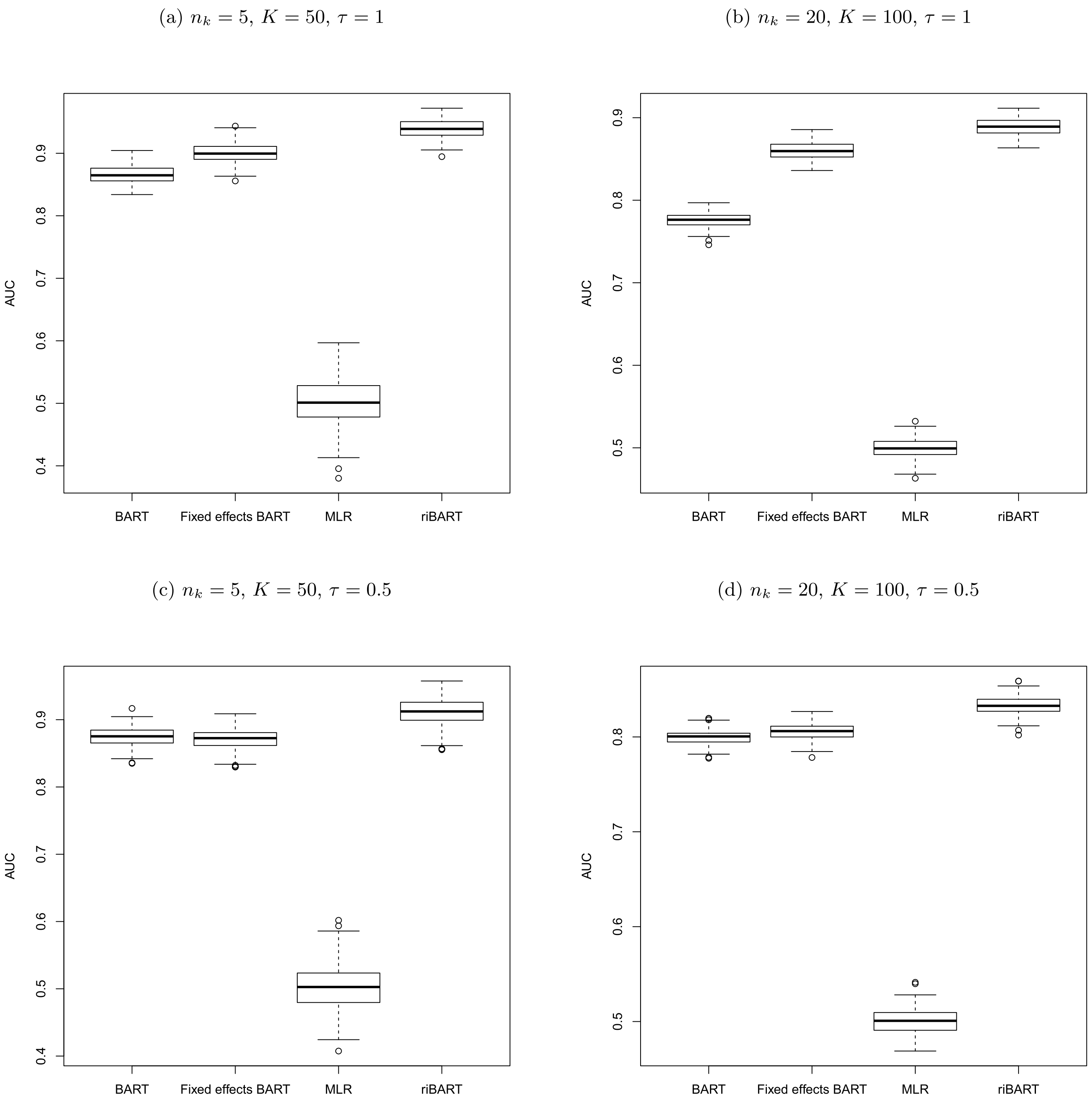
(b) $n_k = 20$, $K = 100$, $\tau = 1$, $\sigma = 1$

(c) $n_k = 5$, $K = 50$, $\tau = 0.5$, $\sigma = 1$

(d) $n_k = 20$, $K = 100$, $\tau = 0.5$, $\sigma = 1$

Figure 1. Boxplots of mean squared error (MSE) for continuous correlated outcomes produced by BART, Fixed effects BART, and riBART.

continuous correlated outcomes, the bias and RMSE for all methods would be similar under all scenarios for $g(x) + a_k$. However, the coverage for riBART would be closer to the nominal coverage of 95% under all scenarios. For σ, the bias produced by riBART was usually the smallest and coverage was usually the highest. These results suggest that riBART should be employed for continuous correlated outcomes if inference for $\sum_{j=1}^{m} g(\mathbf{X}_{ik}, T_j, \mathbf{M}_j) + a_k$ or σ are desired. For binary correlated outcomes, the main focus of our paper, Table 2 suggests that riBART usually has the smallest bias compared with BART, fixed effects BART, and MLLR under all simulation scenarios. riBART also has the better coverage in our simulation scenario compared to the rest of the methods we considered. These results together with the AUC results from Figure 2 suggest that for binary correlated outcomes, riBART should be employed.


(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 2. Boxplots of area under the receiver operating characteristic curve (AUC) for binary correlated outcomes produced by BART, Fixed effects BART, MLR, and riBART.

5. PREDICTING DRIVER STOP BEFORE LEFT TURN EXECUTION

Given the success of riBART in our simulation scenarios, especially for possibly correlated binary outcomes, we now turn to investigate whether this superior performance produced by riBART would propagate to our dataset.

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 [4]. This study collected naturalistic driving data from 108 licensed drivers in Michigan between April 2009 and April 2010. In the study, 16 late-

Table 1. Simulation results for continuous correlated outcomes. Bias and coverage of $\sum_{j=1}^{m} g(\mathbf{X}_k, T_j, \mathbf{M}_j) + a_k$ $(g(x) + a_k)$ and σ for BART, riBART, fixed effects BART, and multiple linear regression (MLR)

Scenario 1: continuous, $n_k = 5$, $K = 50$, $\tau = 1$, $\sigma = 1$								
	$g(x) + a_k$				σ			
	Bias	RMSE	Coverage (%)	AIL*	Bias	RMSE	Coverage (%)	AIL
BART	< 0.01	0.06	95.05	3.40	0.04	0.14	92.00	0.51
riBART	< 0.01	0.06	95.44	3.22	-0.04	0.07	99.50	0.41
Fixed effects BART	< 0.01	0.06	94.68	3.18	0.11	0.15	83.00	0.42
MLR	< 0.01	0.06	48.72	6.92	3.64	3.64	0.00	0.76
Scenario 2: continuous, $n_k = 20$, $K = 100$, $\tau = 1$, $\sigma = 1$								
	$g(x) + a_k$				σ			
	Bias	RMSE	Coverage (%)	AIL	Bias	RMSE	Coverage (%)	AIL
BART	< 0.01	0.02	82.72	2.50	0.32	0.33	0.00	0.10
riBART	< 0.01	0.02	92.77	1.81	-0.01	0.02	92.50	0.08
Fixed effects BART	< 0.01	0.02	89.57	1.78	0.06	0.06	34.50	0.11
MLR	< 0.01	0.02	45.74	6.42	3.69	3.70	0.00	0.27
Scenario 3: continuous, $n_k = 5$, $K = 50$, $\tau = 0.5$, $\sigma = 1$								
	$g(x) + a_k$				σ			
	Bias	RMSE	Coverage (%)	AIL	Bias	RMSE	Coverage (%)	AIL
BART	< 0.01	0.06	89.22	2.64	-0.24	0.25	37.50	0.41
riBART	< 0.01	0.06	94.80	3.05	-0.09	0.10	96.00	0.37
Fixed effects BART	< 0.01	0.06	94.66	3.09	0.07	0.12	90.00	0.40
MLR	< 0.01	0.06	49.32	6.91	3.56	3.56	0.00	0.74
Scenario 4: continuous, $n_k = 20$, $K = 100$, $\tau = 0.5$, $\sigma = 1$								
	$g(x) + a_k$				σ			
	Bias	RMSE	Coverage (%)	AIL	Bias	RMSE	Coverage (%)	AIL
BART	< 0.01	0.02	91.02	2.04	0.05	0.05	36.00	0.08
riBART	< 0.01	0.02	92.69	1.78	-0.01	0.02	91.00	0.08
Fixed effects BART	< 0.01	0.02	90.03	1.76	0.05	0.05	45.50	0.11
MLR	< 0.01	0.02	46.26	6.42	3.61	3.62	0.00	0.27

*AIL = Average interval length.

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 107 drivers made 1,822 left turns (One driver removed because he or she only made one left turn). 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 the prediction performance of our model compared to BART, while simultaneously producing an estimate of a driver's tendency to stop before executing a left turn.

5.2 Data preparation

A detailed description of how we determined and prepared our dataset for analysis using riBART can be found in the Appendix. We provide a brief description in the following paragraphs to aid discussion.

We begin by extracting 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 to provide a distance-varying definition for our binary outcomes of whether a vehicle would stop before executing a left turn in the future. Our outcome was whether a vehicle would eventually stop before executing a left turn, estimated repeatedly at 1 meter intervals before the intersection. We defined $Y_{ikd} = 1$ for the vehicle that would stop eventually before executing a left turn where d is the d^{th} meter from the center of an intersection and i indexes the turns for driver k, $i = 1, \ldots, n_k$. For the vehicles that would not stop before executing a left turn, we defined them as $Y_{ikd} = 0$. 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.

Figure 3 shows the resulting profile of proportion of stops from -100 meters to the center of the intersection (0 meters). We can see that majority (about 65%) of the left turns did not stop before executing a left turn. At -100m, about 35% of the vehicles would stop before executing a left turn. As

Table 2. Simulation results for binary correlated outcomes. Bias and coverage of $\sum_{j=1}^{m} g(\mathbf{X}_k, T_j, \mathbf{M}_j) + a_k$ $(g(x) + a_k)$ for BART, riBART, fixed effects BART, and multiple linear logistic regression (MLLR)

Scenario 5: binary, $n_k = 5$, $K = 50$, $\tau = 1$				
	$g(x) + a_k$			
	Bias	RMSE	Coverage (%)	AIL*
BART	0.02	0.09	73.01	2.12
riBART	0.01	0.10	93.31	2.61
Fixed effects BART	0.03	0.09	62.77	1.61
MLLR	< 0.01	0.11	43.13	1.37
Scenario 6: binary, $n_k = 20$, $K = 100$, $\tau = 1$				
	$g(x) + a_k$			
	Bias	RMSE	Coverage (%)	AIL
BART	0.02	0.04	52.35	1.40
riBART	< 0.01	0.03	94.56	1.62
Fixed effects BART	0.02	0.04	53.60	1.08
MLLR	-0.01	0.04	32.54	1.01
Scenario 7: binary, $n_k = 5$, $K = 50$, $\tau = 0.5$				
	$g(x) + a_k$			
	Bias	RMSE	Coverage (%)	AIL
BART	< 0.01	0.08	92.51	2.13
riBART	< 0.01	0.08	95.32	2.22
Fixed effects BART	0.01	0.08	84.27	1.63
MLLR	-0.02	0.11	62.14	1.53
Scenario 8: binary, $n_k = 20$, $K = 100$, $\tau = 0.5$				
	$g(x) + a_k$			
	Bias	RMSE	Coverage (%)	AIL
BART	< 0.01	0.03	80.72	1.42
riBART	< 0.01	0.03	94.81	1.40
Fixed effects BART	0.01	0.03	78.53	1.05
MLLR	-0.02	0.05	51.40	1.18

*AIL = Average interval length.

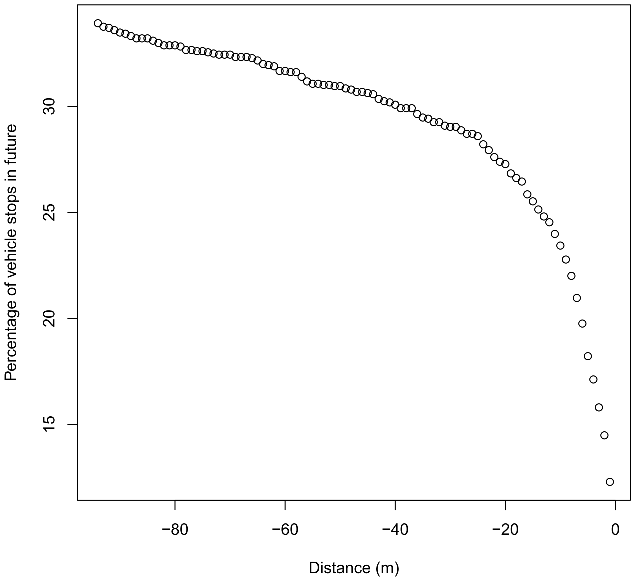


Figure 3. Proportion of vehicles in our study that would be stopped ($\leq$ 1m/s) at some future point for each meter away from the center of an intersection.

vehicles approach the center of an intersection, the proportion of vehicles that eventually stop decreases gradually until about -25m. Beyond -25m, there was a quick drop in the proportion of vehicles that stop suggesting that most vehicles 'decide' to stop about 25m away from the center of an intersection.

At any given distance, we could use the full profile of a vehicle's past speeds as the predictors, but these speeds may contain irrelevant information. Thus, we employed Principal Components Analysis (PCA) to summarize the distance series of vehicle speeds. A detailed description of our decision to use PCA can be found in [23]. In brief, we found that the principal components (PCs) of vehicle speed provided us with much more information than just dimension reduction. The first three PC loadings were fairly similar meter by meter as the vehicle approaches the center of an intersection. In addition, these PCs seemed fairly interpretable as first, second, and third derivatives of the vehicle's location relative to the center of the intersection. The first PC could be loosely interpreted as average speed, second PC as acceleration, and third PC as jerk, change in acceleration. We only included the first two PCs as our predictors because the first two PC scores explained more than 99% of the variation in vehicle speed at all distances (See Figure 4). In addition, we found that adding PC scores beyond these did not produce a large improvement in prediction (See Figure 5).

To decide on our preliminary prediction method, we compared the AUC performance of the following models: logistic regression with polynomial transformation on the predictors, logistic regression with splines for the predictors, BART, and SuperLearner [24] with elastic net [25], logistic regression, K-Nearest Neighbor [26], generalized additive models [5], mean of the outcomes, and BART as the ensemble learners (results not shown here). BART easily outperformed all of the approaches with respect to AUC except the SuperLearner. For the SuperLearner, it sometimes somewhat outperformed BART at a far distance from the intersection but as the vehicle approaches the intersection, SuperLearner stabilized at or a little below BART. Given the unstable AUC performance of the SuperLearner, we focused our attention on extending BART to account for the clustering in our dataset.

Incorporating information from further distances into the estimation of the PCs might also introduce noise to our two PC predictors. Hence, we estimated 8 sets of the first and second PCs from the moving window of vehicle speeds with lengths 3 meters, 4 meters, . . ., 10 meters. We then computed the 10-fold cross validation AUC profile produced by each set with the first and second PCs as the predictor and BART as the model. We finally compared these 8 different AUC profiles and found that a window length of 6 meters gave us the best balance between AUC value and window

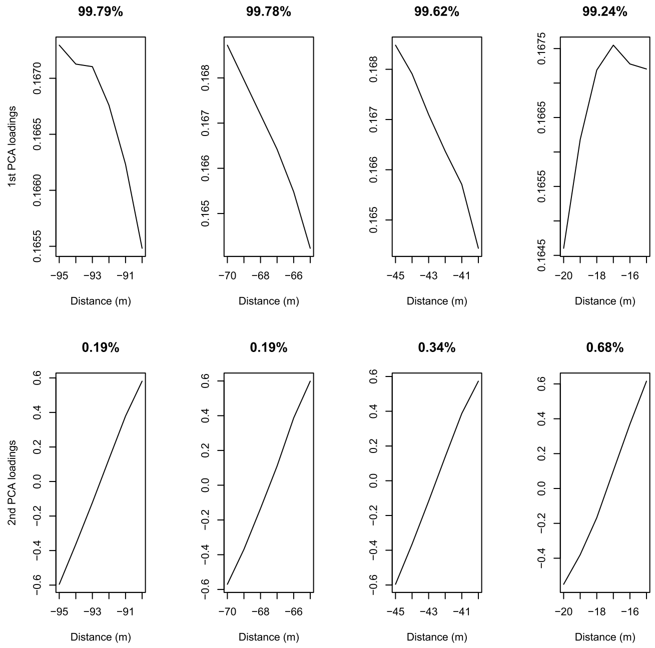


Figure 4. Principal Component loadings for the first and second PC from -95m to -90m, -70m to -65m, -45m to -40m, and -20m to -15m (left to right). The percentages indicate the proportion of variation explained by each PC.

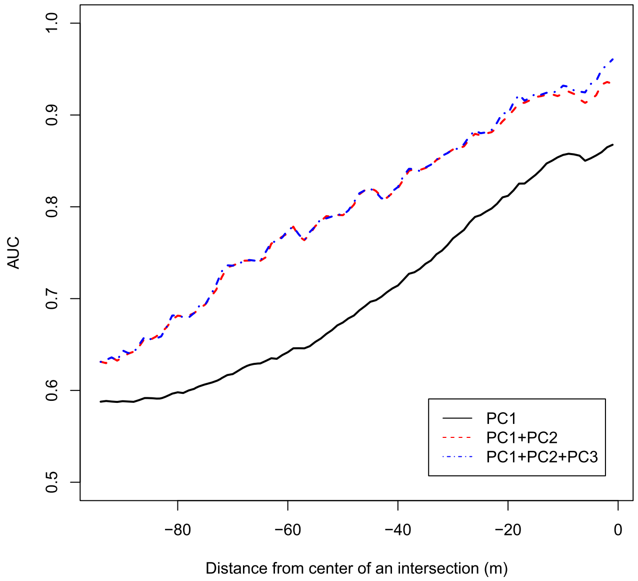


Figure 5. Comparing the Area Under the receiver operating characteristic Curve (AUC) profile gains of including each Principal Component (PC) in the logistic regression model.

length. The result of this comparison can be found in Figure 6 of [23].

Finally, we included a categorical predictor, the number of times the vehicle has stopped up to the current location, to adjust for the likely correlation within each turn. The categories for this predictor were: for -94m to -64m, 0 or ≥ 1; for -63m to -20m, 0, 1, or ≥ 2; and for -19m to -1m, 0, 1, 2, or ≥ 3. Table 3 illustrates the resulting data matrix before analysis.

Table 3. Example of resulting matrix for our IVBSS study dataset

d	k	i	X_{ikd1}	X_{ikd2}	X_{ikd3}
1	1	1	x	x	x
⋮	⋮	⋮	⋮	⋮	⋮
1	1	n_1	x	x	x
1	2	1	x	x	x
⋮	⋮	⋮	⋮	⋮	⋮
1	2	n_2	x	x	x
⋮	⋮	⋮	⋮	⋮	⋮
1	107	n_{107}	x	x	x
2	1	1	x	x	x
⋮	⋮	⋮	⋮	⋮	⋮
2	1	n_1	x	x	x
⋮	⋮	⋮	⋮	⋮	⋮
94	107	n_{107}	x	x	x

5.3 Analysis

We fit riBART with a random effect at the driver level which incorporates within-driver correlation to our dataset. Because we fit riBART meter-by-meter, a slight clarification in notation of the riBART is needed. We model $P(Y_{ikd} = 1|\mathbf{X}_{ikd})$ as

$$P(Y_{ikd} = 1|\mathbf{X}_{ikd}) = \Phi[G(\mathbf{X}_{ikd}],$$

where $\mathbf{X}_{ikd} = (X_{ikd1}, X_{ikd2}, X_{ikd3})^T$, $k = 1, \ldots, K$ indexes the drivers, i indexes the turns for driver k, $i = 1, \ldots, n_k$, and $d = -94, \ldots, -1$ indexes the distance from the center of an intersection. The riBART model is then

$$G(\mathbf{X}_{ikd}) = \sum_{j=1}^{m} g(\mathbf{X}_{ikd}, T_{jd}, \mathbf{M}_{jd}) + a_{kd}, \tag{10}$$

where $a_{kd} \sim N(0, \tau_d^2)$. Note that we are estimating each model at distance d separately and assuming that there is a different random intercept for each driver at each d.

For comparison, we also ran BART, which ignores within-driver correlation; fixed effects BART, which ignores within-driver correlation but adjusts for the driver effect in the model; a random intercept linear logistic regression (riLogistic), which incorporates within-driver correlation but ignores non-linearity and complex interactions; and MLLR, which ignores within-driver correlation, non-linearity, and complex interactions. It may have been more straight forward to use polynomial or splines of our first two PCs together with a random intercept to obtain a model that handles non-linearity and driver correlations. Unfortunately, even simple models with a quadratic main effect or a single knot spline at the mean or median produced convergence errors for the random intercept GLM model. Hence, we did not include them as competitors against riBART. We obtained the linear logistic regression using the *glm* function in *R* while the random intercept linear logistic regressions were obtained using the *glmer* function from the *R* package *lme4*. We compared the in-sample AUC of the six methods and computed the 95% CI of the AUCs using the method of [27], which uses a linear approximation of the AUC to the Somer's D statistic to obtain an estimate of the variance of AUC. In addition, we investigated the proportion of depth of the 200 regression trees over 5,000 iterations for each meter as well as the marginal effects of each main effects and interaction to explore the additional features provided by riBART.

5.4 Results

Figure 6 shows (a) the the estimated intra-class correlation (ICC, $\frac{\tau^2}{\tau^2+1}$) profile; (b) the AUC profiles of riBART, BART, fixed effects BART, riLogistic, and MLLR; and (c) the AUC profile difference between riBART versus BART, riBART versus fixed effects BART, riBART versus riLogistic, and riBART versus MLLR.

The posterior mean profile of ICC was small, between about 0.12 and 0.15, and fairly stable as the vehicle approaches the center of an intersection. This suggests firstly that the variance parameter, τ, for the random intercept, a_k, is small for left turn stops and secondly that as the vehicle approaches the center of the intersection, the effect of individual 'habits' of the driver remained relatively stable throughout the left turn maneuver. For the AUC profile, we see evidence that riBART performed better than BART, fixed effects BART, riLogistic, and MLLR. The difference in AUC profile between riBART versus BART, riBART versus fixed effects BART, riBART versus riLogistic, and riBART versus MLLR remained negative throughout the left turn maneuver suggesting the superior prediction performance of riBART to the other prediction methods we considered.

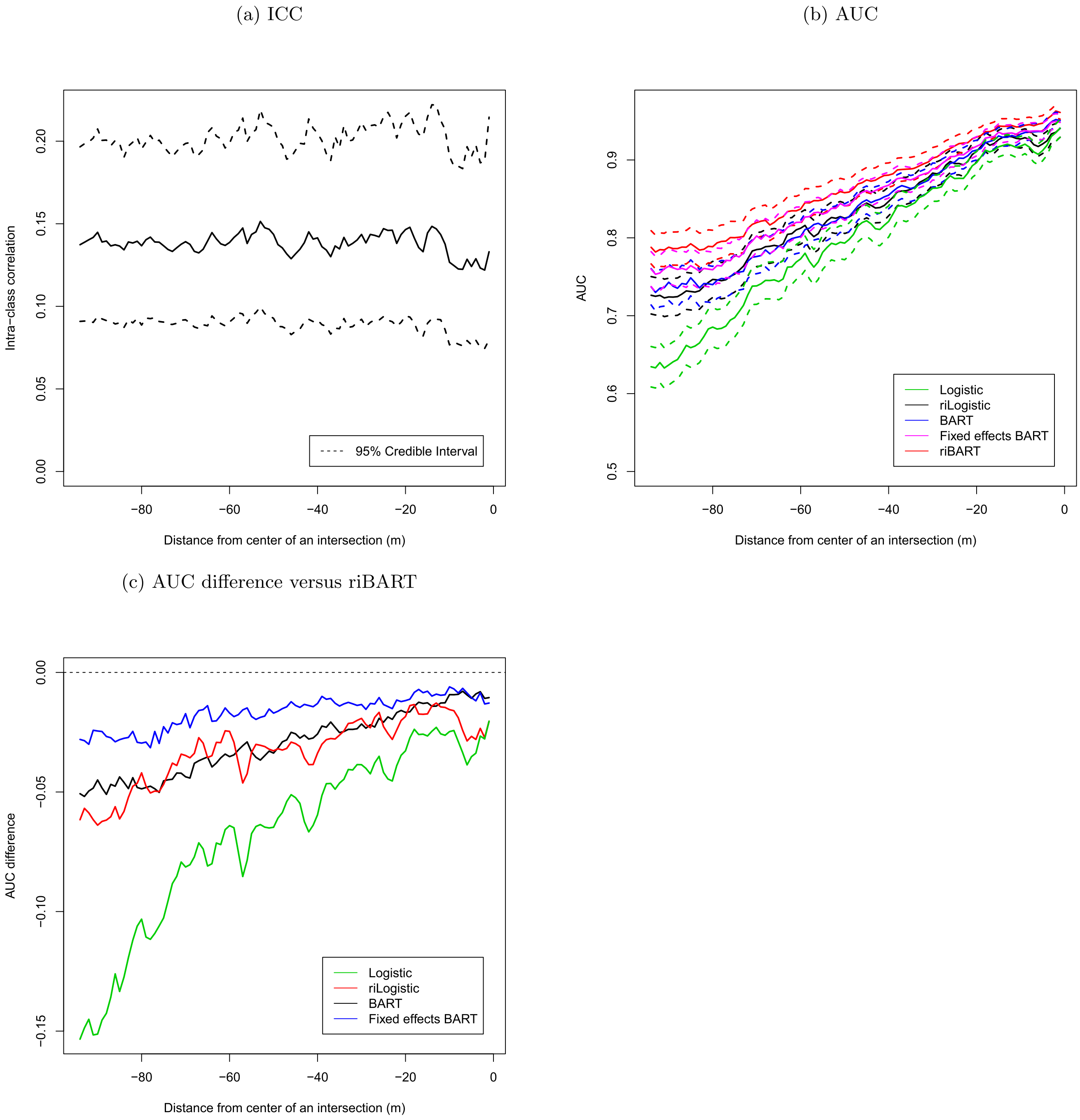
At 94m away from the center of intersection, riBART produced an AUC estimate of 0.79 [95% C.I. (0.77, 0.81)]. Comparatively, fixed effects BART produced an AUC of 0.76 (0.74, 0.78), BART produced an AUC of 0.74 (0.71, 0.76), riLogistic produced an AUC of 0.73 (0.70, 0.75), and MLLR produced an AUC of 0.64 (0.61, 0.66). In situations where last-second decisions are needed for example, Automatic Emergency Braking, an AUC of 0.79 would not be enough. However, the application that we envision for our algorithm is to provide further information to an oncoming driverless vehicle and help it make better decisions in conjunction with its own sensor-based algorithms. As such, almost any AUC value greater than 0.50 should improve the decision made by the driverless vehicle. Most likely, a driverless vehicle would use this information to adjust its own speed (up or down) so that any potential conflict between it and the human-driven turning vehicle is less ambiguous (e.g., speeding up to pass before the turning vehicle would turn or slowing down to let the turning vehicle go).

Figure 7 shows the proportion of depth of each regression tree meter by meter from -94m away from the center of an intersection to -1m away from the center of an intersection. About 90% of the regression trees employed by riBART were single terminal nodes for every meter, 9% were trees with one internal node with two child terminal nodes, and the rest, about 1%, had regression tree depths of more than 1. This suggests a rather strong penalization effect for the tree structure depth which was what the BART portion of riBART was aiming for. We also investigated the frequency of each main and interaction effect being used by each regression tree to give us a sense of which main or interaction effect was most used, hence an indication of effect importance (results not shown here). We found that the main effects were most frequently used (excluding single terminal node trees) followed by the two-way interactions and lastly the three-way interaction. These results suggest that the two most important variables could be the first two PCs.

Figure 8 shows the smoothed marginal effect plots of all the main effects at -45m (approximately halfway through the left turn). The clear non-linearity of the main effects and the reduced use of the interactions by riBART suggests that the substantial improvement provided by riBART over random intercept linear logistic regression came from the non-linear effects. Since PC1 can be loosely interpreted as the average speed, plot (a) suggest that at -45m, a higher average speed suggests a lower probability of stopping with a sharp decline in the probability when the average speed increases to around 12-13 m/s. As the average speed increases to about 17-18 m/s, the probability of stopping increases again. Smoothed marginal effect plots for PC1 from -94m to -1m can be found in the supplementary materials available online.

For PC2, since it could be loosely defined as the acceleration of the vehicle, plot (b) suggests that negative acceleration produces a higher probability of stopping while positive acceleration produces a lower probability of stopping halfway through the left turn maneuver. This result continues as the vehicle approaches the center of an intersection. The smoothed marginal effect plots for PC2 from -94m to -1m can be found in the supplementary materials available online. Note that for PC2, the PC loadings sometimes suggest deceleration instead of acceleration i.e. the slope for PC2 in Figure 4 is negative instead of positive. We have placed a condition (multiplying the loadings by -1 whenever this occurs) in our implementation to ensure that the heuristic interpretation of PC2 will always stay as acceleration.

Plot (c) shows the boxplot of the predicted probability of stopping stratified by the number of times a vehicle has



(a) ICC

(b) AUC

(c) AUC difference versus riBART

Figure 6. (a) The intra-class correlation (ICC) profile of riBART as a factor of distance from the intersection; (b) Area under the receiver operating characteristic curve (AUC) profile of riBART, BART, and random intercept logistic regression (dotted lines are 95% Credible Interval); and (c) AUC difference profile between riBART versus BART and riBART versus random intercept linear logistic regression.

stopped previously before -45m. From the stratified boxplots, we can see that as the number of times the vehicle has stopped previously increases, the vehicle is slightly more likely to be predicted to stop before executing a left turn.

In summary, Figure 8 suggests that vehicles with lower average speed, and/or slowing down quickly, and/or have stopped multiple times previously would be more likely to stop compared to vehicles with higher average speed, accelerating, and has not made a previous stop. This agrees with our understanding of how a vehicle would stop at an intersection before executing a left turn and suggests that riBART is producing sensible results.

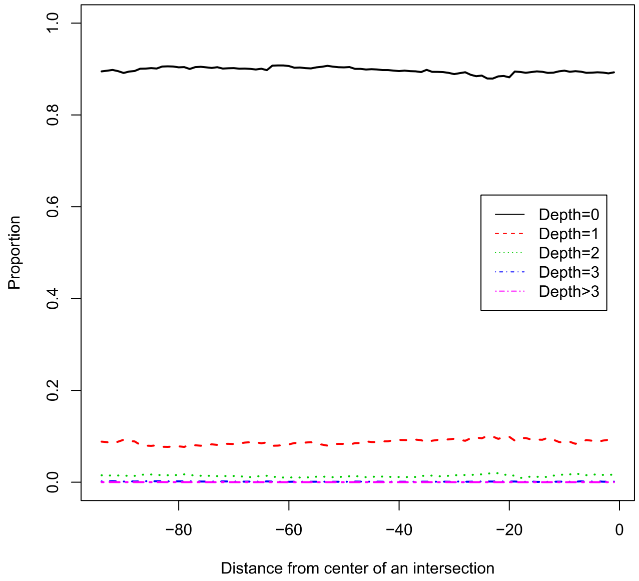


Figure 7. Proportion of depth of regression tree meter by meter.

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 analysis, 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 straight-forward and can be implemented on correlated binary outcomes. We have also provided codes that would implement riBART in our supplementary materials available online. Our codes could be used to explore some of the properties and features that riBART provided over the random intercept linear logistic regression. These results could help the researcher make sense of the marginal effects provided by each variable estimated using riBART.

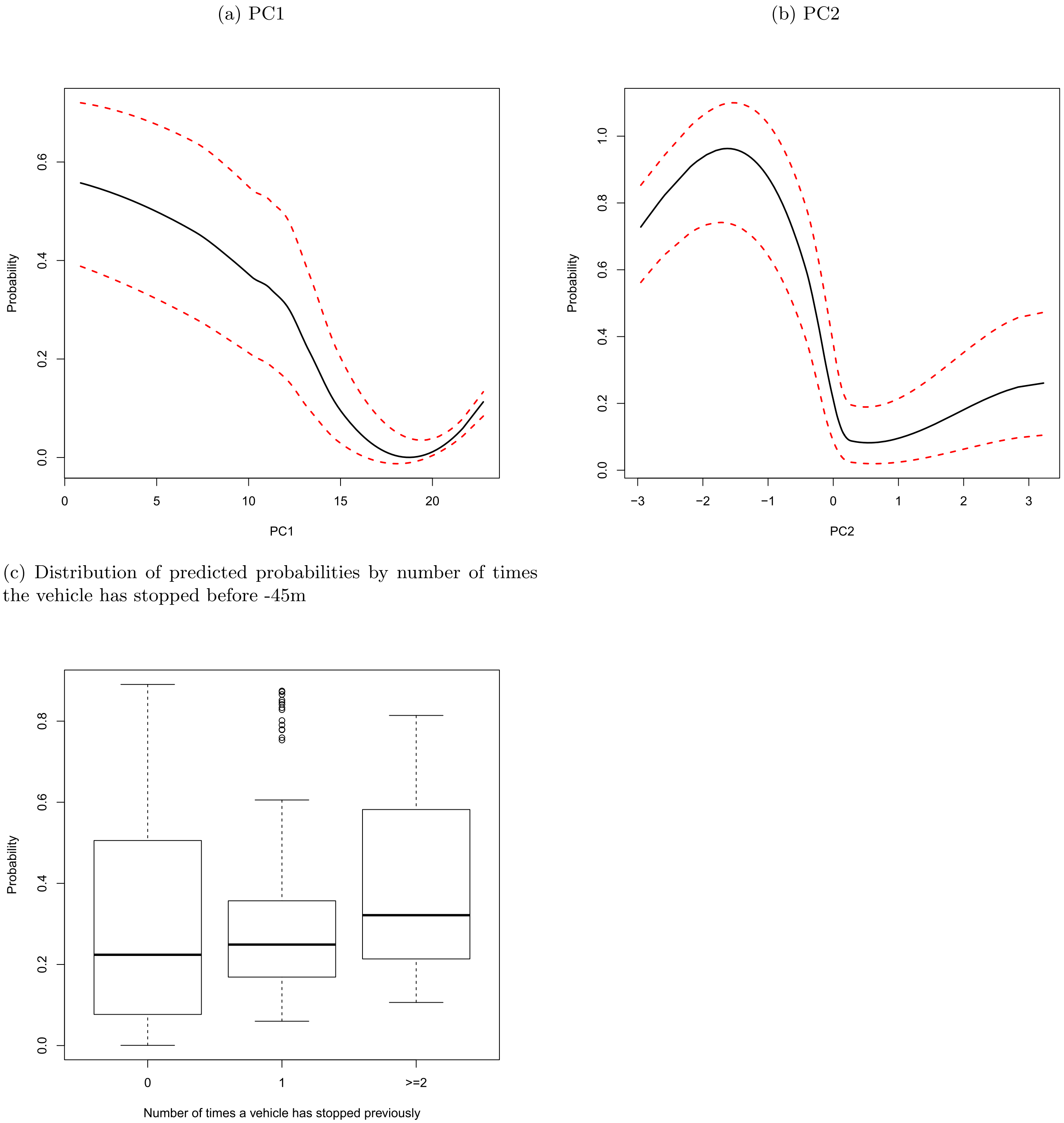
Applying riBART to our dataset, substantial improvement in prediction compared to BART can be obtained when we take into account that different drivers have different 'propensities to stop' before executing a left turn at an intersection; that is, the inclusion of a random intercept improves prediction performance for our dataset compared to a model without a random intercept. This implies that future development of an operational algorithm should try to accommodate the similarities of stopping behavior for a given human driver through a learning algorithm. For example, devices that are able to transmit information about a driver's propensity to stop could be installed on vehicles to improve the decision-making performance of the self driving vehicle.

To elaborate, we are assuming that this method would be used to create a prediction profile that would be broadcast to autonomous vehicles, thus utilizing all of the available information on the turning behavior both across and within vehicles. For a new vehicle, we could treat the posterior means of the random intercepts in our dataset as a "quasi" distribution for the random intercept of the unseen driver. Alternatively, we could draw an initial random intercept distribution using the posterior distribution of the random intercept variance parameter. Once this driver makes a turn, their random intercept can be estimated and updated.

In our simulation study, we found that the 95% coverage for σ was reduced when the number of clusters and the number of observations within a cluster was large ($n_k = 20$, $K = 100$). The likely cause for the poor coverage is due to low variation in the posterior draw of σ resulting in reduced average 95% credible interval length. We believe this low variation in σ is due to the regression trees in BART getting stuck at certain tree structures. This phenomenon of regression trees getting stuck at certain tree structures has been discussed by [28] previously. The difference here is that [28] only reported observing regression trees being stuck when the true σ is small for regression trees. We argue that regression trees might also get stuck when the effective sample size, N, is large. This is because with a large N, deeper trees tend to produce a better fit for R_{kj} in Eq. (3). However, when a regression tree gets deep, the standard grow, prune, change, and swap steps will have trouble proposing new trees with radically different tree structures. This lack of radically different tree structures implies reduced variability in the tree structures, which is indirectly reflected by the lack of variation in σ.

This issue is separate from the development of BART in the correlated data context, and indeed would occur even when observations are independent. We illustrate this with an example using BART implemented via the *BayesTree* package in *R*. We generated $Y_k = 10\sin(\pi X_{k1} X_{k2}) + 20(X_{k3} - 0.5)^2 + 10X_{k4} + 5X_{k5} + \epsilon_k$ with $X_{kq} \overset{i.i.d.}{\sim}$ Uniform$(0,1)$, $q = 1,\ldots,5$ and $\epsilon_{ik} \overset{i.i.d.}{\sim} N(0,1)$. We then ran 200 simulations with $\sigma = 1$ and a sample size of 2,000. The resulting bias, RMSE, 95% coverage, and AIL for σ were -0.04, 0.04, 79%, and 0.09 respectively. We observe once again that although bias and RMSE were small, the 95% coverage for σ was far from nominal because the AIL was small. We think that this issue of a lack in variation of σ when the sample size is large could be solved by either increasing the number of regression trees used, re-calibrating the α and β parameters used to penalize each regression tree, or to include the rotate step proposed by [28] in the proposal of a new regression tree in the MH algorithm of BART. As inference about σ is not the key focus of this paper, we leave investigation of this problem with BART to future work.

Although our analysis of left turn data found that the first two PCs appeared to be the most important predictors


(a) PC1

(b) PC2

(c) Distribution of predicted probabilities by number of times the vehicle has stopped before -45m

Figure 8. Smoothed (a) marginal effect of PC1 (b) marginal effect of PC2; and (c) boxplots of the predicted probability of stopping stratified by the number of times a vehicle has stopped previously. Dotted red lines show smoothed 95% credible interval.

based on the frequency of the trees drawn, caution should be exercised when using riBART to decide whether a variable was important. This is because of the default discrete uniform prior we placed on the variables which forces the model to use the variables uniformly for prediction. If variable selection is desired, spike and slab priors could be considered but such an implementation would go beyond the scope of this work.

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 our application, estimating a "turn-level" random effect nested within the driver-level random effect is possible. Eq. (10) could be modified to become

$$G(\mathbf{X}_{ikd}) = \sum_{j=1}^{m} g(\mathbf{X}_{ikd}, T_{jd}, \mathbf{M}_{jd}) + a_{kd} + l_{ik},$$

where $a_{kd} \sim N(0, \tau_d^2)$, $l_{ik} \sim N(0, \tau^2)$, and $a_{kd} \perp l_{ik}$. To estimate this model, we would employ once again a Gibbs-like sampling by drawing τ or l_{ik} conditional on the rest of the parameters and the observed data. By estimating τ and comparing it with τ_d, we could determine if we require additional variables to account for the dependencies in our outcome. This is because if τ was much larger compared to τ_d, this suggests that not all of the variation is captured by the driver level random intercept and there is still some variation left at the turn level. However, such a model is not practical for our prediction situation. This is because the estimated turn-level effect would only be useful for prediction for that turn – but once that turn is completed, we have no interest in predicting it. Other plausible areas for future research include extending BART and riBART to outcomes of other forms, for example, ordinal outcomes or counts.

APPENDIX A

Our original data contains the time series of speed for the vehicle every 10 milliseconds starting from 100 meters away from the center of an intersection. We rescale the original time series predictors to measure distance-series of vehicle speed from the intersection because, in a turn that is not complete, only the distance from the intersection will be known in advance. We recorded the distance series at every single meter i.e. $d = -100, \ldots, -1$ where 0 is the center of an intersection and -100 is 100 meters from the center of an intersection. To determine the vehicle speed at a certain meter, we searched for the vehicle speed recorded that was closet to the meter mark. In the situation where more than one speed sample point was closest to the meter, we took their average as the speed at that meter.

Because vehicles can stop and restart before reaching the center of the intersection, we define "stopping" as a distance-varying outcome. Let $i = 1, \ldots, n_k$ index the i^{th} turn made by the k^{th} driver where $k = 1, \ldots, K$ index the driver. Let s_{ikd} be the distance series of vehicle speed and y_{ikd} be the distance-varying outcome (1=stopped in future, 0=will not stop in future). We defined y_{ikd} as follows:

1. If $s_{ikd} > 1m/s \, \forall \, d = -100, \ldots, -1$, then set $y_{ikd} = 0$ for all d.
2. If $s_{ikd} \leq 1m/s$ for some $d \in \{-100, \ldots, -1\}$, let $c \in \{-100, \ldots, -1\}$ be the index such that for every $d > c$, $s_{ikd} > 1m/s$. We set $y_{ik,-100} = y_{ik,-99} = \ldots = y_{ik,c} = 1$ and $y_{ik,c+1} = y_{ik,c+2} = \ldots = y_{ik,-1} = 0$.

Next, for every d^{th} meter, we defined the moving window of speeds as,

$$M_{ikd} = \{s_{ik,d-w+1}, s_{ik,d-w+2}, \ldots, s_{ikd}\}$$

where w is the size of the moving window. We the implemented PCA on these M_{ikd}s to reduce the number of covariates in our prediction model. Before reduction, the covariates are $s_{..,k-w+1}, s_{..,k-w+2}, \ldots, s_{..d}$. We let

$$M_d = \begin{bmatrix} s_{11,d-w+1} & s_{11,d-w+2} & \cdots & s_{11j} \\ \vdots & \vdots & \vdots & \vdots \\ s_{1n_1,j-w+1} & s_{1n_1,j-w+2} & \cdots & s_{1n_1j} \\ \vdots & \vdots & \vdots & \vdots \\ s_{Kn_K,j-w+1} & s_{Kn_K,j-w+2} & \cdots & s_{Kn_Kj} \end{bmatrix}$$

and

$$u(d) = \begin{bmatrix} u_{d-w+1} \\ u_{d-w+2} \\ \vdots \\ u_d \end{bmatrix}$$

where M_d is the matrix of moving windows with the first row being M_{11d}, n_1^{th} row being M_{1n_1d},..., and the last row being M_{Kn_Kd}. There are w (number of columns in M_d) orthogonal vectors $u(d)$ that decompose the variance of M_d into w parts under the condition that for each $u(d)$, $||u(d)|| = 1$. To obtain the w decomposed variances, we used the formula: $PC_d = Var[M_d u(d)]$. If we let $PC_{d(q)}$ be the ordered statistic where $q = 1, \ldots, w$ and $u(d)_{(q)}$ be the ordered vector corresponding to $PC_{d(q)}$, then the first PC is $\mathbf{X}_{d1} = M_d u(d)_{(w)}$, the second PC is $\mathbf{X}_{d2} = M_d u(d)_{(w-1)}$, and so on.

We used the first two PCs in our analysis for reasons already covered in our main paper. We then added a third predictor, the number of stops made by the vehicle until distance d to obtain Table 3.

ACKNOWLEDGEMENTS

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. Finally, we would like to thank the anonymous referee and associate editor for their many insightful comments and suggestions to help improve the clarity of our manuscript.

SUPPLEMENTARY MATERIALS

The zip file contains two folders and a portable document format (pdf) file (http://intlpress.com/site/pub/files/_supp/SII-2018-11-4-s1.zip). "Codes" contain the codes we used, the pdf file "Posterior distributions" contain the derivations of the conditional draws for the BART and riBART MCMC algorithm, and the "Results" folder contains the marginal effect plots.

Received 27 June 2017

REFERENCES

[1] Google (2015). *What we're up to*, Retrieved August 26, 2015, from http://www.google.com/selfdrivingcar/. Google, Mountain View, CA.

[2] Mchugh, M. (2015). *Tesla's Cars Now Drive Themselves, Kinda*, Retrieved May 15, 2016, from http://www.wired.com/2015/10/tesla-self-driving-over-air-update-live/. Wired Magazine, New York, NY.

[3] Davies, A. (2015). *GM Has 'Aggressive' Plans for Self-Driving Cars*, Retrieved May 15, 2016, from https://www.wired.com/2015/10/gm-has-aggressive-plans-for-self-driving-cars/. Wired Magazine, New York, NY.

[4] Sayer, J. R., Bogard, S. E., Buonarosa, M. L., LeBlanc, D. J., Funkhouser, D. S., Bao, S., Blankespoor, A. D., and Winkler, C. B. (2011). *Integrated Vehicle-Based Safety Systems Light-Vehicle Field Operational Test Key Findings Report*, DOT HS 811 416, National Center for Statistics and Analysis, NHTSA, U.S. Department of Transportation, Washington, DC.

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

[6] Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric regression*. Cambridge University Press: Cambridge, UK.

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

[8] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and regression Trees*. Wadsworth, Belmont, CA.

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

[10] Gammermann, A. (2000). Support vector machine learning algorithm and transduction. *Computational Statistics* **5** 31–39.

[11] Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian Additive Regression Trees. *The Annals of Applied Statistics* **4(1)** 266–298.

[12] Zhang, S., Shih, Y. T., and Müller, P. (2007). A Spatially-adjusted Bayesian Additive Regression Tree Model to Merge Two Datasets. *Bayesian Analysis* **2(3)** 611–634.

[13] Low-Kam, C., Telesca, D., Ji, Z., Zhang, H., Xia, T., Zink, J. I., and Nel, A. E. (2015). A Bayesian regression tree approach to identify the effect of nanoparticles' properties on toxicity profiles. *The Annals of Applied Statistics* **9(1)** 383–401.

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

[15] Kapelner, A. and Bleich, J. (2016). bartMachine: Machine Learning with Bayesian Additive Regression Trees. *Journal of Statistical Software* **70(4)** 1–40.

[16] Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian CART Model Search. *Journal of the American Statistical Association* **93(433)** 935–948.

[17] Albert, J. H. and Chib, S. (1993). Bayesian Analysis of Binary and Polychotomous Response Data. *Journal of the American Statistical Association* **88(422)** 669–679.

[18] Tanner, M. A. and Wong, W. H. (1987). The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association* **82(398)** 528–540. MR0898357

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

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

[21] Albert, J. H. 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, New York: Marcel Dekker, 577–599.

[22] R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[23] Tan, Y. V., Elliott, M. R., and Flannagan, C. A. C. (2017). Development of a real-time prediction model of driver behavior at intersections using kinematic time series data. *Accident Analysis and Prevention* **106** 428–436.

[24] 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.

[25] Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* **33** 1–22.

[26] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* **46(3)** 175–185.

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

[28] Pratola, M. T. (2016). Efficient Metropolis-Hastings Proposal Mechanisms for Bayesian Regression Tree Models. *Bayesian Analysis* **11** 885–911.

Yaoyuan Vincent Tan
1415 Washington Heights
Ann Arbor, MI 48109
United States
E-mail address: vincetan@umich.edu

Carol A. C. Flannagan
2901 Baxter Rd
Ann Arbor, MI 48109
United States
E-mail address: cacf@umich.edu

Michael R. Elliott
1415 Washington Heights
Ann Arbor, MI 48109
United States
E-mail address: mrelliot@umich.edu