A Mixing Time Lower Bound for a Simplified Version of BART

Omer Ronen*[1], Theo Saarinen*[1], Yan Shuo Tan*[4], James Duncan[6], and Bin Yu[1,2,3,5]

[1]Department of Statistics, UC Berkeley
[2]Department of Electrical Engineering and Computer Sciences, UC Berkeley
[3]Center for Computational Biology, UC Berkeley
[4]Department of Statistics and Data Science, National University of Singapore
[5]Microsoft Research
[6]Group in Biostatistics, UC Berkeley

October 19, 2022

Abstract

Bayesian Additive Regression Trees (BART) is a popular Bayesian non-parametric regression algorithm. The posterior is a distribution over sums of decision trees, and predictions are made by averaging approximate samples from the posterior. The combination of strong predictive performance and the ability to provide uncertainty measures has led BART to be commonly used in the social sciences, biostatistics, and causal inference. BART uses Markov Chain Monte Carlo (MCMC) to obtain approximate posterior samples over a parameterized space of sums of trees, but it has often been observed that the chains are slow to mix. In this paper, we provide the first lower bound on the mixing time for a simplified version of BART in which we reduce the sum to a single tree and use a subset of the possible moves for the MCMC proposal distribution. Our lower bound for the mixing time grows exponentially with the number of data points. Inspired by this new connection between the mixing time and the number of data points, we perform rigorous simulations on BART. We show qualitatively that BART's mixing time increases with the number of data points. The slow mixing time of the simplified BART suggests a large variation between different runs of the simplified BART algorithm and a similar large variation is known for BART in the literature. This large variation could result in a lack of stability in the models, predictions and posterior intervals obtained from the BART MCMC samples. Our lower bound and simulations suggest increasing the number of chains with the number of data points.

*Equal contribution, alphabetical ordering

1 Introduction

Decision tree models such as CART (Breiman et al., 1984) and their ensembles such as Random Forests (Breiman, 2001) and Gradient Boosted Trees (Chen & Guestrin, 2016; Friedman, 2001) have proved to be enormously successful supervised learning algorithms, because they are able to combine non-parametric model fitting with implicit dimension reduction. It is often difficult to quantify the uncertainty of their predictions and due to their greedy local splitting criteria, there is no guarantee for the optimality of the constructed decision trees. An alternative approach is to construct the decision trees in a Bayesian manner (H. A. Chipman et al., 1998; Denison et al., 1998; Wu et al., 2007)

To address these issues, H. A. Chipman et al., 1998 proposed a Bayesian adaptation of CART, Bayesian CART, and later, a sum of Bayesian CART trees, which they called Bayesian Additive Regression Trees (BART) (H. A. Chipman et al., 2010). One perspective views these algorithms as non-greedy stochastic versions of their deterministic equivalents, where the randomness inside the fitting process allows the algorithm to explore the space of possible decision trees in ways the CART algorithm cannot. An alternative perspective views these algorithms as Bayesian non-parametric regression models, in which we put a prior on the space of decision trees, assume a likelihood for the observed data, and then obtain a posterior distribution over the possible decision trees based on the training data. The posterior distribution can be used to provide posterior predictive credible intervals and other forms of uncertainty quantification. Due to strong predictive performance (H. Chipman et al., 2006; H. A. Chipman et al., 2010) and the ability to quantify the uncertainty of predictions, BART has spawned a number of variants (Hahn et al., 2020; J. L. Hill, 2011; Linero & Yang, 2018; Murray, 2021; Pratola et al., 2020; Sparapani et al., 2016) and has become increasingly popular in fields such as the social sciences (Green & Kern, 2010; Yeager et al., 2019), biostatistics (Starling et al., 2020; Wendling et al., 2018), and causal inference (Dorie et al., 2019; Green & Kern, 2012; Hahn et al., 2019; J. Hill et al., 2020; J. L. Hill, 2011; Kern et al., 2016).

Several recent works have theoretically analyzed BART variants from a frequentist perspective. Concentration results have been shown for the BART posterior under assumptions on the smoothness of the underlying regression function and the prior used for the BART algorithm. Ročková and Saha, 2019 show that when the BART prior is modified to decrease the mass on deeper trees as the number of training points increases, the posterior distribution of this BART variant concentrates around the true regression function at nearly the optimal rate. When the true regression function exhibits smoothness, Linero and Yang, 2018 introduce a smoothing variant of BART and show that the modified BART posterior concentrates at nearly the optimal rate for that class of functions. In addition to concentration, there are several results for the feature selection consistency of the BART posterior. When the regression function is smooth and sparse, Ročková and van der Pas, 2020 show that BART with a sparsity inducing prior can perform effective feature selection. Liu et al., 2021 examine an approximate Bayesian computation algorithm to combine with BART for feature selection in high dimensional data. However, all of these theoretical results rely on the ability to sample from the BART posterior.

1.1 Prior Work on BART MCMC Mixing

Since there is no closed form expression for the BART posterior, the standard approach is to sample from it approximately via a Markov chain Monte Carlo (MCMC) algorithm designed by H. A. Chipman et al., 2010. Despite an abundance of empirical evidence, described following this, that the posterior samples from the BART algorithm do not mix well, researchers in many fields have regularly used the BART posterior for uncertainty quantification (Bisbee, 2019; Carlson, 2020; Dorie et al., 2019; Green & Kern, 2012; J. Hill & Su, 2013; Waldmann, 2016; Yeager et al., 2019; Zhang et al., 2020). Further, there has been very little theoretical analysis of how quickly the samples from the BART MCMC algorithm converge to the posterior distribution. This is problematic for several reasons: First, credible intervals obtained from the MCMC samples will not actually reflect the posterior, and are therefore of questionable meaning for inference. Next, it means that different runs of the algorithm may produce somewhat different results, thereby lacking stability and reproducibility. Finally, a necessary condition for the recent results on posterior concentration and model selection consistency, is the ability to sample from the posterior distribution. When the BART MCMC algorithm is slow to mix, it is difficult to satisfy this condition in practice.

Since the introduction of the BART algorithm, the problem of mixing has been observed, (H. A. Chipman et al., 2010; He & Hahn, 2021; Pratola, 2016) leading to a number of works evaluating and attempting to address this issue. The difficulties in mixing have been explored empirically in several works (H. A. Chipman et al., 1998; Pratola, 2016; Wu et al., 2007) and mentioned in several recent survey papers (J. Hill et al., 2020; Linero, 2017). There have also been several algorithmic suggestions including modifying the MCMC proposal moves (Pratola, 2016; Wu et al., 2007), initializing the trees in the algorithm from greedily constructed trees as a "warm start" (He & Hahn, 2021), and running multiple chains to quantify the uncertainty in the predictions (Carnegie, 2019; Dorie et al., 2019). Despite the prevalence of works mentioning the mixing of the BART MCMC algorithm, little theoretical work has been done to understand why the mixing is slow.

1.2 BART and Bayesian CART

In this paper we will discuss several variants of the BART algorithm. The Bayesian CART algorithm was originally introduced as an algorithm for fitting a single decision tree in a Bayesian setting. The BART algorithm fits a sum of these Bayesian CART trees and when this sum has one term, devolves to the Bayesian CART algorithm. BART enjoys better prediction accuracy than Bayesian CART (H. Chipman et al., 2006; H. A. Chipman et al., 2010; J. Hill et al., 2020), and is hence used more often in practice. Because Bayesian CART and each individual tree in the BART sum share the same set of MCMC proposal moves, Bayesian CART has been used to study the empirical effect of these proposal moves on the mixing of both algorithms (H. A. Chipman et al., 2010; J. Hill et al., 2020; Pratola, 2016; Wu et al., 2007). In our theoretical work, we begin by using a simplified version of Bayesian CART towards the goal of explaining the properties of BART. For theoretical tractability, we restrict the MCMC proposals available to the Bayesian CART algorithm to a subset of the standard moves and call

the resulting algorithm *simplified BART*. We present our theoretical mixing time lower bound for the simplified BART algorithm. We compare the mixing time of BART to simplified BART in simulations in Section 4.1, and compare Bayesian CART to simplified BART in Section 4.2 in order to compare the propensity of the two algorithms to select poor initial splitting features.

Table 1: A summary of the different algorithms studied in this paper.

Algorithm \ Property	Number of Trees	MCMC moves
BART	200	**Grow**, **Prune**, **Change**, **Swap**
Bayesian CART	1	**Grow**, **Prune**, **Change**, **Swap**
Simplified BART	1	**Grow**, **Prune**

1.3 Our contributions

We present a theoretical result for the failure of the simplified BART MCMC algorithm to mix and show through extensive simulations that the BART MCMC algorithm has a similar mixing issue. As far as we know, our work is the first attempt at theoretically analyzing the mixing behavior of a simplified version of the Bayesian CART or BART algorithms, with insights that also hold qualitatively for BART. Our specific contributions are as follows:

1. **Mixing time lower bounds:** We obtain a mixing time lower bound for a simplified version of the BART algorithm. Assuming that all features are discrete, we are able to show that the total variation mixing time of the simplified BART MCMC chain on the space of tree structures is bounded from below by $\exp(\Omega(n))$, where n is the number of training data points. This theoretical result holds for any data generating process with features from any random distribution on a discrete feature space of dimension at least 2 and a random outcome vector y. In addition, our proof identifies one reason for the slow mixing: It is difficult for the chain to switch the split at the root of the tree.

2. **Simulations:** Taking several large real datasets from the Penn Machine Learning Benchmark repository (PMLB) (Olson et al., 2017), we run 8 independent instances of the MCMC chain for BART and for simplified BART and study the mixing of the chains according to several metrics. We use the root mean squared error (RMSE) from a held out test set as a summary statistic of each sampled regression function, and compute the distribution of this quantity over each chain after discarding a conservative number of burn-in samples. The failure of the chain to mix can be observed visually by comparing the density plots for the RMSE of the different chains (see Fig 3). We also compute the Gelman-Rubin (GR) diagnostic values across the different chains, and observe that in many cases these exceed the threshold of 1.1 that was recommended by Gelman and Rubin, 1992 as a quantitative indicator of the failure of mixing. Through our

simulations, we find that *the BART algorithm shows worse mixing as the number of training data points increases*, as indicated by mixing diagnostics as well as qualitative measures of the regression function. We also find that Bayesian CART is susceptible to the bottleneck that we exploit for simplified BART in the proof of our theoretical result: Difficulty in reversing the split at the root of the tree. We did not find strong evidence that this bottleneck affects the BART algorithm to the same degree.

To the extent of our knowledge, our theoretical result is the first that studies the mixing time of either the BART or Bayesian CART algorithms. Our simulations suggest that the problems identified theoretically for simplified BART still affect the original BART and Bayesian CART algorithms. In particular, the issue of mixing time increasing with the number of data points suggests that BART practitioners may benefit by running more chains for data with many observations, an observation not yet explored in the literature.

2 Preliminaries

In this section, we describe the elements of the Bayesian CART model underlying the simplified BART algorithm as well as the MCMC sampling algorithm, as described in the original paper (H. A. Chipman et al., 1998). We define the Bayesian CART model formally because it only differs from the simplified BART algorithm we analyze in the set of MCMC moves.

2.1 The Bayesian model specification

Parameter space: Recall that the Bayesian CART model is a non-parametric regression model, which means that the regression function f is a random function. Unlike Gaussian process regression, we constrain it to take values on the space of decision tree functions. We say that $f\colon \mathcal{X}^d \to \mathbb{R}$ is a decision tree function if it is a piecewise constant function on a partition of $\mathcal{X}^d$, where the partition is generated by recursively splitting $\mathcal{X}^d$ along the coordinate directions. Naturally, f can be parameterized by a tuple comprising the binary tree structure $\mathcal{T}$, and $\mathbf{\Theta} \in \mathbb{R}^b$, where $b = |\mathcal{T}|$ is the number of leaves in $\mathcal{T}$, and $\mathbf{\Theta}_i$ is the value of f on leaf i (after choosing a canonical ordering of the leaves). In turn, the tree structure $\mathcal{T}$ can be thought of as a labeled graph, and thus further parameterized by the underlying ordered rooted binary tree, as well as labels on each internal node of the form (v_j, τ_j), which respectively denote the feature and threshold for the split associated with node j. We assume that our covariate space is discrete, i.e. $\mathcal{X} = \{1, \ldots, m\}$ for some integer m.[1] This implies that a tree structure $\mathcal{T}$ can have at most m^d leaves, which, together with a finite choice of labels at each node, implies that Ω, the collection of all tree structures, is discrete and finite.

[1] This is almost WLOG as in practice, splits for continuous features are chosen from a grid of possible values corresponding to the quantiles of the features in the training data.

Prior for $\mathcal{T}$: The prior on the tree structure is defined in terms of a stochastic process: Starting with a trivial tree with a single node, each newly generated node is split with probability $\alpha(1+\delta)^{-\beta}$, where δ is the depth of the node, while α and β are universal hyperparameters. If a node is split, the feature it splits on is drawn uniformly from all available features, and then the threshold is drawn uniformly from all available values of the feature, if it is discrete, and from all available values that have been observed in the training data, if it is continuous.

Prior for Θ: We put independent Gaussian priors for the leaf parameters: $\Theta|\mathcal{T} \sim \mathcal{N}(\bar{\mu}\mathbf{1}, \sigma^2\mathbf{I}_b)$. In the original Bayesian CART algorithm, σ^2 is treated as a Bayesian parameter drawn from an inverse Gamma distribution. For the sake of theoretical tractability, we shall treat it instead as a fixed hyperparameter. We believe that this is relatively innocuous as, in practice, σ^2 is known to quickly concentrate around a fixed value.

Data likelihood: We put an independent Gaussian likelihood function on the errors in the responses. In other words, given observed data $\mathbf{X} = (\mathbf{x}_i)_{i=1}^n$, $\mathbf{y} = (y_i)_{i=1}^n$, we assume

$$\mathbf{y}|\mathbf{X}, \Theta, \mathcal{T} \sim \mathcal{N}\big((f(\mathbf{x}_i))_{i=1}^n, a\sigma^2\mathbf{I}_n\big),$$

where $a > 0$ is a fixed number.

The relationships between the variables described in this section can be summarized in the following Bayesian network diagram.

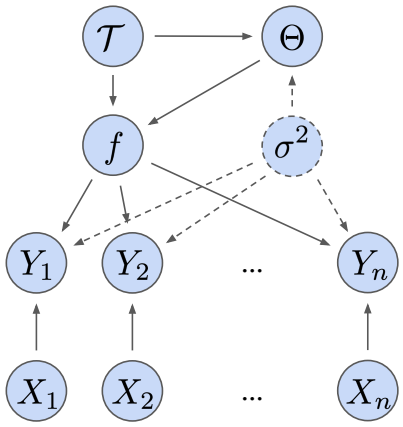

Figure 1: A Bayesian network showing the dependency relationships between the random variables in the simplified BART model. σ^2 is treated as a parameter in Bayesian CART, but will be treated as a fixed hyperparameter in our theoretical analysis.

2.2 Sampling from the Bayesian CART posterior

To sample from the posterior $p(\mathcal{T}, \mathbf{\Theta}|\mathbf{X}, \mathbf{y})$, we first decompose it as $p(\mathbf{\Theta}|\mathcal{T}, \mathbf{X}, \mathbf{y})p(\mathcal{T}|\mathbf{X}, \mathbf{y})$. The Gaussian specification of Θ and the data likelihood allow us to compute the first term in closed form, and so, we only need to use MCMC to sample from $p(\mathcal{T}|\mathbf{X}, \mathbf{y})$, the marginal posterior on the space of tree structures. We do this using the Metropolis-Hastings algorithm. When the chain is at a given tree $\mathcal{T}$, the proposed next tree $\mathcal{T}'$ is obtained by randomly choosing from one the following four moves: 1. **Grow** the tree by splitting a leaf node chosen uniformly at random. The split feature and threshold are also chosen uniformly at random, as in the prior. 2. **Prune** the tree by collapsing a pair of adjacent leaf nodes chosen uniformly at random. 3. **Change** the split feature and threshold (draw them again from the uniform distribution) of an internal node selected uniformly at random. 4. **Swap** the split features and thresholds of a parent-child node pair chosen uniformly at random. An accept-reject filter is then applied to ensure that the chain has the marginal posterior as the stationary distribution. The original simplified BART paper proposed selecting each move with equal probability. Later when proposing BART, the authors updated their suggestion to use the probabilites $(0.25, 0.25, 0.4, 0.1)$.

2.3 Simplified BART

We define the simplified BART algorithm as the Bayesian CART algorithm defined in Section 2.2 with the MCMC moves restricted to **Grow** and **Prune** each of which we select with probability $.5$. This is the algorithm that we theoretically analyze in Section 3.

2.4 Contrasting Bayesian CART with BART

BART is Bayesian model for a tree sum that is built on top of Bayesian CART. The same priors on tree structure and leaf parameters are used, while the likelihood function of the noise in the response remains Gaussian. However, the regression function is now the sum of all the tree functions in the sum, and the parameter space thus comprises the product $(\mathcal{T}_i, \mathbf{\Theta}_i)_{i=1}^M$, where M is the number of trees. To sample from the resulting posterior, we use a combination of Gibbs sampling and the Metropolis-Hastings algorithm. More precisely, we cycle through the trees in the model and update them as follows: Given a tree $\mathcal{T}_i$ in the model, conditioned on the values of parameters from all the other trees, we update $\mathcal{T}_i$ using the same procedure as in Bayesian CART, except that we replace the responses with residuals in the likelihood function. We then make a single draw from the conditional posterior for $\mathbf{\Theta}_i$, and then repeat the process for the next tree in the sum. A single update of the MCMC chain comprises an update for all M trees in the sum.

3 Mixing Time Lower Bound

Notation on probabilities: We will use $p(-)$ to denote the marginal and conditional probabilities associated with the Bayesian model described in Sec 2.1. We use $Q(-)$ to

denote probabilities associated with the Markov chain for the simplified BART algorithm on Ω, the space of tree structures described in Sec 2.2. As a shorthand, we will also denote the stationary distribution, the posterior marginal on Ω, as π. Finally, we assume that our observed data $(\mathbf{X}, \mathbf{y})$ comprise n i.i.d. data points from a generative regression model with regression function $f_0(\mathbf{x}) = \mathbb{E}\{y \mid \mathbf{x}\}$. We will denote probabilities, expectations, and variances with respect to the generative process using $\mathbb{P}$, $\mathbb{E}$, and Var. Note that the generative model is not necessarily the same as the data likelihood in the fitted Bayesian model.

The mixing time (see also Levin et al., 2006) of the Markov chain of the simplified BART algorithm, Q, is defined to be

$$t_{mix} \coloneqq \min\{t : \max_{\mathcal{T} \in \Omega} \|Q^t(-|\mathcal{T}) - \pi\|_{\text{TV}} \leq 0.25\}. \tag{1}$$

The quantity being maximized over in the right hand side of the equation is the total variation distance between the time t distribution of the chain initialized at a tree structure $\mathcal{T}$ and the stationary distribution. A larger mixing time means that the chain takes a long time to reach the stationary distribution from a worst case initialization.[2]

The main theoretical result of our paper is the following theorem.

Theorem 1 (Mixing time lower bound)**.** *Suppose $d \geq 2$ or $m \geq 2$. Also assume that y has a bounded distribution, i.e. $|y| \leq K$. Then with probability at least $1 - 1/n$, the mixing time of the simplified BART Markov chain, Q, as described in Sec 2.3, satisfies*

$$t_{mix} \geq \exp\left(\frac{n}{2a\sigma^2}\text{Var}\{f_0(\mathbf{x})\} - O(\sqrt{n \log n})\right). \tag{2}$$

The theorem says that under trivial assumptions, the mixing time of the simplified BART MCMC algorithm grows exponentially in the number of data points. This is surprising and unfortunate because we generally expect the performance of prediction algorithms to improve with more data points.

Our proof of the lower bound proceeds by constructing two tree structures $\mathcal{T}$ and $\mathcal{T}'$ that each give rise to a partition on which f_0 is piecewise constant, but whose splits at the root node differ from each other. This means that both $\mathcal{T}$ and $\mathcal{T}'$ possess large posterior mass, but are separated from each other in the state space by a bottleneck for the chain – the trivial tree with only a root node. This bottleneck becomes increasingly difficult to traverse as the number of data points increases.

While the result concerns mixing for the tree structure $\mathcal{T}$ and not the regression function f, in most cases we may select $\mathcal{T}$ and $\mathcal{T}'$ so that the partition associated with $\mathcal{T}$ is a refinement of that associated with $\mathcal{T}'$. In this case, the resulting conditional posterior distributions $p(f \mid \mathcal{T}, \mathbf{X}, \mathbf{y})$ and $p(f \mid \mathcal{T}', \mathbf{X}, \mathbf{y})$ are different, which strongly suggests that the stochastic process induced by Q on the space of regression functions[3] also mixes slowly with respect to Wasserstein distances. Such a result, while even more revealing, would be relatively technically difficult, and we leave it to future work.

[2]The choice of 0.25 as the threshold is by convention and does not affect the mixing time up to multiplicative constants.

[3]The transition kernel on the full parameter space is given by $\tilde{Q}(\mathcal{T}', \boldsymbol{\Theta}'|\mathcal{T}, \boldsymbol{\Theta}) = Q(\mathcal{T}'|\mathcal{T})p(\boldsymbol{\Theta}'|\mathcal{T}', \mathbf{X}, \mathbf{y})$. Since the tuple $(\mathcal{T}, \boldsymbol{\Theta})$ determines f, this chain induces a stochastic process on the space of regression functions.

The notion of a bottleneck is formalized by the definition of *conductance*. The conductance of a subset $S \subset \Omega$ is defined as the ratio

$$\Phi(S) \coloneqq \frac{Q(S, S^c)}{\min\{\pi(S), \pi(S^c)\}},$$

where

$$Q(S, S^c) = \sum_{\mathcal{T} \in S, \mathcal{T}' \in S^c} \pi(\mathcal{T}) Q(\mathcal{T}' | \mathcal{T})$$

is the probability of starting in S and ending in S^c over one step of Q when applied to the stationary distribution. A small conductance implies that it is difficult for Q to transit from S to S^c. This intuition can be used to derive the following well-known inverse relationship between mixing time and conductance.

Lemma 2 (Conductance and mixing time, Theorem 7.3 in Levin et al., 2006)**.**

$$t_{mix} \geq \frac{1}{4\Phi^*}$$

where t_{mix} is the mixing time of the simplified BART algorithm defined in 1 and

$$\Phi^* \coloneqq \min_{S \subset \Omega} \Phi(S).$$

By using the bottleneck at the trivial tree alluded to earlier, we can upper bound the conductance in terms of the minimum of two likelihood ratios, each of which compares the trivial tree to a nontrivial one making a different root split.

Lemma 3 (Conductance and likelihood ratio)**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be two tree structures whose splits at the root node differ from each other.*

Then

$$\Phi^* \leq C \min\left\{ \frac{p(\mathbf{y}|\mathcal{T}_0, \mathbf{X})}{p(\mathbf{y}|\mathcal{T}, \mathbf{X})}, \frac{p(\mathbf{y}|\mathcal{T}_0, \mathbf{X})}{p(\mathbf{y}|\mathcal{T}', \mathbf{X})} \right\}, \tag{3}$$

where $C = C(\mathcal{T}, \mathcal{T}', \alpha, \beta, m, d)$ is a constant not depending on n, and $\mathcal{T}_0$ is the trivial tree comprising only a root node.

To further bound each likelihood ratio in (3), we formulate a more general result for the log likelihood ratio between the trivial tree and *any* candidate tree $\mathcal{T}$. The result shows that the normalized log likelihood ratio concentrates around the population *impurity decrease* between $\mathcal{T}_0$ and $\mathcal{T}$.

Lemma 4 (Likelihood ratio and impurity decrease)**.** *For any tree $\mathcal{T}$, with probability at least $1 - 1/n$, we have*

$$\begin{aligned} &\left| \log \frac{p(\mathbf{y}|\mathcal{T}_0, \mathbf{X})}{p(\mathbf{y}|\mathcal{T}, \mathbf{X})} + \frac{n\Delta}{2a\sigma^2} \right| \\ \leq\ & CK^2 a^{-1} \sigma^{-2} \sqrt{\log n} \big(\sqrt{n} + b \big) + b \log(n/a). \end{aligned}$$

where

$$\Delta \coloneqq \mathrm{Var}\{f_0(\mathbf{x})\} - \mathbb{E}\{\mathrm{Var}\{f_0(\mathbf{x}) \mid \mathcal{T}(\mathbf{x})\}\}$$

is the decrease in impurity of $f_0(\mathbf{x})$ when conditioning on the partition given by $\mathcal{T}$, and C is an absolute constant.

The empirical impurity decrease is used by CART to choose which splits to make and which to prune, while the likelihood ratio is used by Bayesian CART to accept or reject proposed grow, prune, change, or swap moves. This lemma thereby further illustrates the similarities and differences between the two algorithms. More importantly for this paper, it completes the toolbox we need to prove our main theorem.

Proof of Theorem 1. By assumption, there are at least two possible splits at the root node, (v, τ) and (v', τ'). Starting from either split, it is possible to grow a tree on whose leaves the true regression function is piecewise constant. Call these two trees $\mathcal{T}$ and $\mathcal{T}'$ respectively, and notice that they automatically satisfy the hypotheses of Lemma 3 so that (3) holds. Furthermore, we have

$$\text{Var}\{\mathbb{E}\{f_0(\mathbf{x})|\mathcal{T}(\mathbf{x})\}\} = \text{Var}\{\mathbb{E}\{f_0(\mathbf{x})|\mathcal{T}'(\mathbf{x})\}\} = 0.$$

By Lemma 4, we therefore have

$$\log \frac{p(\mathbf{y}|\mathcal{T}_0, \mathbf{X})}{p(\mathbf{y}|\tilde{\mathcal{T}}, \mathbf{X})} = -\frac{n}{2\sigma^2}\text{Var}\{f(\mathbf{x})\} + O(\sqrt{n \log n})$$

for $\tilde{\mathcal{T}} = \mathcal{T}, \mathcal{T}'$. Exponentiating and applying Lemma 3 followed by Lemma 2 then gives us (2). □

4 Simulations on Mixing for BART and simplified BART

In this section, we perform two simulation experiments to understand how the conclusions from our theoretical analysis of simplified BART in the previous section provide new insights for understanding BART and Bayesian CART. Specifically our simulations suggest that:

- The BART MCMC algorithm often mixes poorly, and the mixing quality decreases as the number of training data points n increases.
- The root split made by the Bayesian CART algorithm is often chosen suboptimally and yet is rarely reversed, thereby creating a bottleneck for the chain.

Code Availability: All the code necessary to reproduce the experiments in this section is publicly available at github.com/theo-s/bart-sims The computing infrastructure used was a Linux cluster managed by Department of Statistics at UC Berkeley. Most runs of the simulation used a single 24-core node with 128 GB of RAM, while the larger datasets required a large-memory node with 792 GB RAM and 96 cores.

Data: We use real-world datasets in order to emulate how BART is used in practice. Specifically, we utilize the four largest datasets from the Penn Machine Learning Benchmarks (PMLB) (Olson et al., 2017) in order to study how the mixing time depends on the number of training data points. Table 2 details the dimensions of these datasets.

Table 2: PMLB datasets utilized in simulations

Name	Samples	Features
Breast tumor (Romano et al., 2020)	116640	9
California housing (Pace & Barry, 1997)	20640	8
Echo months (Romano et al., 2020)	17496	9
Satellite image (Romano et al., 2020)	6435	36

Algorithm settings and hyperparameters: We use the **dbarts** R package (Dorie, 2022) with the following non-default hyperparameters. First, we increase the number of burn-in samples from 100 to 5000 (*nskip*=5000), in order to highlight that mixing does not occur within a reasonable number of iterations. Second, we run 8 chains (*nchain*=8) to facilitate the analysis of the mixing time. Last, we define the proposal distribution of simplified BART to select grow and prune with equal probability (*probs*=(1-1e-5,1e-5,0,.5)[4]). All other hyperparameters are kept at their default values. In particular, for simplified BART we use *ntree*=1 and for BART we use *ntree*=200.

4.1 Experiment 1: Mixing time increases with n

Choice of mixing metric: Each sample in a chain is a function, and we calculate its RMSE on a held-out evaluation set, comprising 10% of the dataset, as a one-dimensional summary statistic. To quantify the degree of mixing, we compute the Gelman-Rubin convergence diagnostic (GR) (Gelman & Rubin, 1992), which compares the within-chain variation to the across-chain variation.[5] It is widely accepted that a GR value which exceeds 1.1 indicates failure to mix (Gelman & Rubin, 1992). We implicitly assume that the degree of mixing at a fixed number of iterations is indicative of the mixing time.

Varying n: To vary n, we draw random sub-samples without replacement from each dataset of sizes $n = 200$ and $n = 2000$ in addition to using the full dataset.

Results: Across 20 experimental replicates, we compute the GR diagnostic value for BART at each configuration described previously, and summarize our results in Fig 2. For comparison, we also display the corresponding results for the simplified BART

[4] 1e-5 is used avoid a numeric error, this term does not affect the proposal probabilities.

[5] The Gelman-Rubin diagnostic is a statistic that uses the observation that multiple MCMC chains should realize similar values under convergence to give a measure of convergence for a collection of MCMC chains. Suppose we have L samples from J different MCMC chains, let x_{ij} denote the ith sample from the jth chain, $\bar{x}_j$ denote the mean sample from the jth chain, and $\bar{x}$ denote the overall mean sample. The Gelman-Rubin diagnostic is defined as the ratio $R = \frac{\frac{L-1}{L}W + \frac{1}{L}B}{W}$ where W denotes the within-chain variance: $W := \frac{1}{J}\sum_{j=1}^{J} \frac{1}{L-1}\sum_{i=1}^{L} (x_{ij} - \bar{x}_j)^2$ and B denotes the between-chain variance: $B := \frac{L}{J-1}\sum_{j=1}^{J} (\bar{x}_j - \bar{x})^2$.

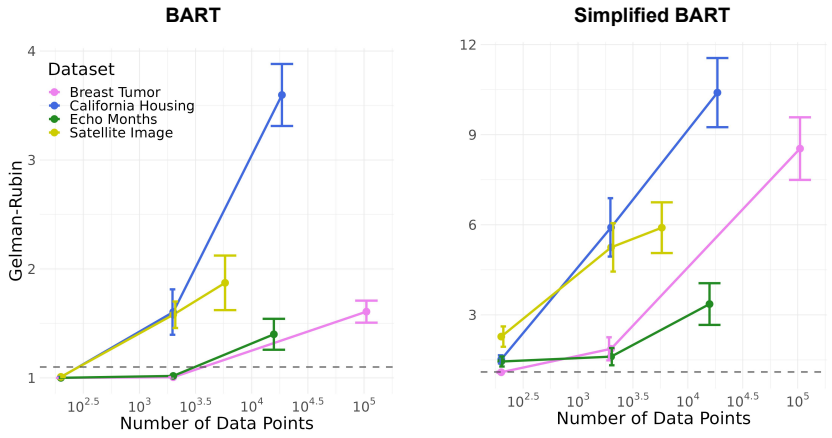

Figure 2: Mixing quality for BART and simplified BART decreases with the number of data points. Mixing quality is measured in terms of the Gelman-Rubin diagnostic applied to the test set RMSE using 8 chains. Error bars are calculated over 20 different Monte carlo runs using the same training data. The left column shows the results for BART, while the right for simplified BART . The horizontal dashed line represent the recommended mixing threshold of 1.1.

algorithm that we analyzed theoretically. We observe that the mixing time increases with n for both BART and simplified BART. Furthermore, even when running the chain with more burn-in samples than recommended, we observe that BART fails to mix on all of the original datasets. This failure to mix for large n is also visually demonstrated for BART and simplified BART respectively in Fig 3 and Appendix A.2.1, through RMSE density plots. As n increases the RMSE densities show visible differences. Observing such differences is extremely unlikely if these chains are comprised of samples from a distribution that concentrates around a fixed function. Furthermore, these plots illustrate a subtle point that mixing reflects the relative magnitude of the between-chain variation compared with the within-chain variation, and not its absolute magnitude. In particular, the between-chain variation for the Breast Tumor dataset is no more than 0.1% of the RMSE value, while the GR value is 1.74. Appendix A.2.2 visualizes the same information using the cusum path plots (Yu & Mykland, 1998), which have also been used for MCMC diagnostics. We observe that the chains follow a smoother path as the number of data points increases, which is an indication of slow mixing (see Appendix A.2.2 for more details).

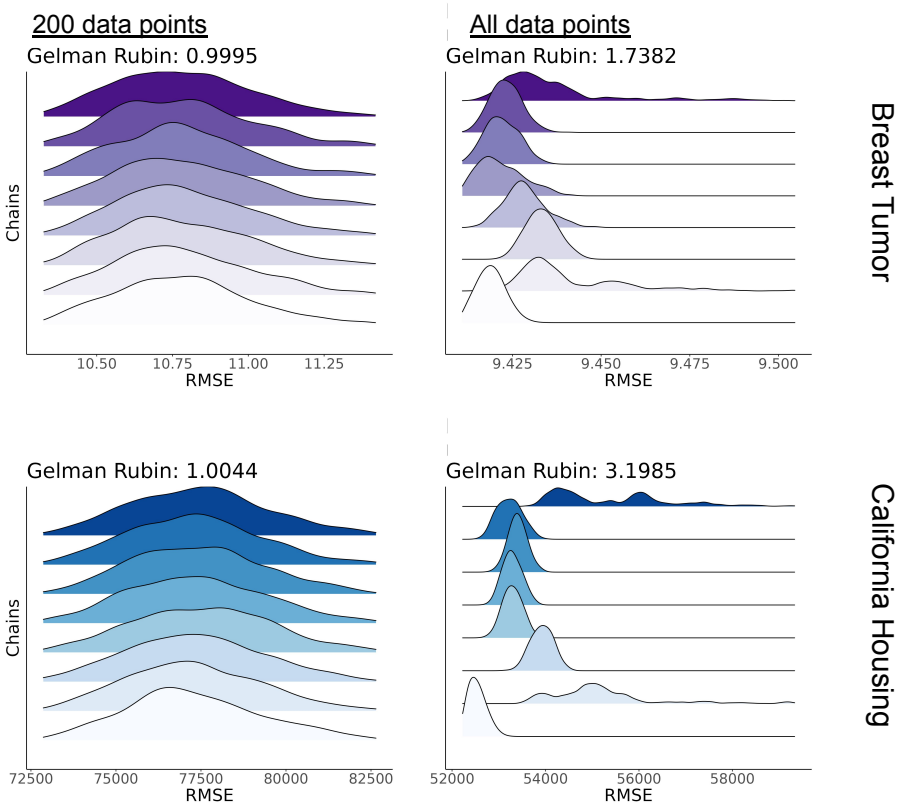

Figure 3: Kernel density plots of BART RMSE values across different chains and data sets and sample sizes. The number of data points used increases from the left to right column. Different rows correspond to different datasets.

4.2 Experiment 2: Root split stuck on suboptimal feature

Additional set-up: In contrast to the previous experiment, we study Bayesian CART instead of BART, in order to understand to what extent the two additional moves, **Change** and **Swap**, can help avoid bottlenecks. For each of the 6000 iterations of each chain (including the burn-in iterations), we record the index of the feature on which the root node splits.

Results: The results suggest that the root node changes in only a tiny fraction of the iterations, which decreases as n increases. Specifically, when each full dataset is used, the root split changes in less than 0.2% of the samples on average across 160 chains. Appendix A.3 plots the number of iterations during which the feature on which the root node splits changes (via a prune, change, or swap move) against the number of data points, and demonstrates that the probability of such change decreases with n.

Therefore, even with the full set of moves, changing the root split remains a bottleneck for the chain.

In Fig 4, we summarize the recorded split indices by counting the number of iterations on which the root node splits on each feature. In each subplot, we display the results separately for each of the 8 independent chains. We observe that for full datasets, an overwhelming majority of the root splits occur on the same feature, and furthermore, this feature is different for different chains. Similar analysis for simplified BART is deffered to Appendix A.3.1.

This further supports our claim about the bottleneck, while demonstrating the instability of the algorithm at the level of tree structures. Moreover, it implies that the chain can get stuck with a root split that is suboptimal.

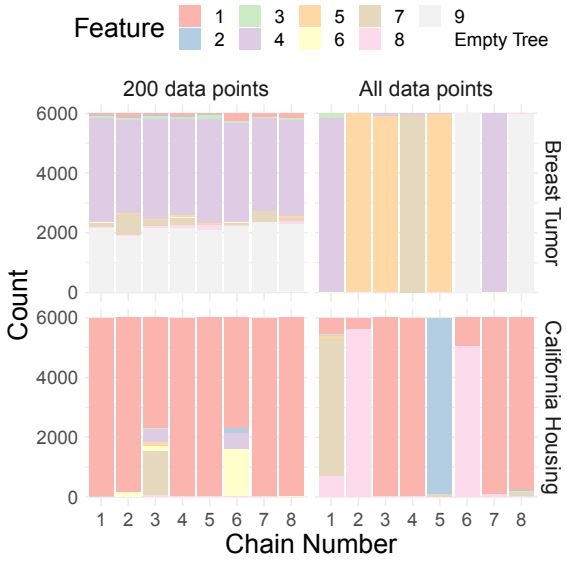

Figure 4: Number of iterations on which the root node splits on each feature for different chains on two datasets. When n is large (right column), almost all MCMC samples for a single chain of Bayesian CART have a root split on the same feature, whose index varies across chains. The different colors correspond to the different features, and empty tree refers to a single leaf (no splits).

5 Discussion

Poor mixing of the BART and Bayesian CART MCMC algorithms has been observed since these algorithms were introduced. In this work we provide the first theoretical analysis that proves a mixing time lower bound scaling with the number of training data points for a simplified version of the BART MCMC algorithm. Our analysis of simplified BART can partially explain why this occurs: The first split in the tree forms a bottleneck for the mixing of the chain, and the bottleneck becomes more difficult

to traverse as the number of training data points increases. While we do not provide theoretical results for BART, our simulations show that the trend of mixing degradation with the number of data points suggested by our theory also holds for BART. We also learn from our simulations that Bayesian CART also encounters a bottleneck when splitting on the first feature; the restricted move set of simplified BART seems not solely responsible for this bottleneck.

The bottleneck formed by the first split in the tree helps explain one cause of the poor mixing of Bayesian CART and possibly BART. This split, which may be sub-optimal, forms a bottleneck for the chain and makes it difficult for it to transition to a tree with a differing root split. Furthermore, both Bayesian CART and BART choose the proposed feature and threshold uniformly at random, and are not guaranteed to make an optimal split initially.

Future Directions: We have made four simplifying assumptions to make the theoretical analysis of BART tractable. Most significantly, we studied a single tree instead of a sum, and only allowed **Grow** and **Prune** moves for the MCMC proposal (thereby excluding the **Change** and **Swap** moves). If either **Change** or **Swap** moves are allowed, the conductance computations would become more complicated and we may not be able to use the same bottleneck set used in the proof of Theorem 1. In future work, we aim to expand our result to hold for a single tree with an unrestricted set of MCMC moves. Analyzing the behavior of a sum of trees would require developing a new set of analytical tools, and appears prima facie to be more difficult. Despite these challenges, a result in this vein would be insightful as it would offer a better understanding of the BART algorithm as used in practice.

The remaining simplifying assumptions are mostly technical, and do not greatly affect the generality of our results. Our assumption of discrete features is relatively benign as a continuous feature can be approximated by its discretization on a sufficiently fine grid. Relaxing this assumption would entangle the prior and the likelihood, since in practice for continuous features, splits are selected from a grid of values dependent on the data. Lastly, our assumption that the variance of the Gaussian prior placed on the leaf parameters, σ^2, is fixed is not divergent from practice because it converges to a fixed value within a small fraction of the iterations that the algorithm runs for (H. A. Chipman et al., 2010). Allowing σ^2 to be random, as is done in practice, would complicate the form of the integrated likelihood and make an equivalent version of Lemma 4 more difficult to prove.

We believe our work should be viewed as only a first step towards rigorously understanding the mixing properties of BART and suggesting principled improvements to this class of MCMC algorithms. Our work suggests several natural ways to improve the mixing of BART in practice: The proposal distribution for selecting splits should be data-adaptive rather than uniform at random, the chain should be initialized at an intelligent guess for the possible trees, and the number of MCMC chains run for BART should increase with the number of training data points. In our future work, we plan to relax the assumptions on moves in simplified BART and develop concrete methods for data-driven splits, tree structure initialization and scaling the number of chains depending on the number of training data points.

Acknowledgements

We would like to thank the Statistical Computing Facility (SCF) of the Department of Statistics at University of California, Berkeley for computing support, and in particular Professor Jacob Steinhardt for access to his group's large-memory nodes within the SCF Linux cluster.
We gratefully acknowledge partial support from NSF TRIPODS Grant 1740855, DMS-1613002, 1953191, 2015341, 2209975, IIS 1741340, ONR grant N00014-17-1-2176, NSF grant 2023505 on Collaborative Research: Foundations of Data Science Institute (FODSI), the NSF and the Simons Foundation for the Collaboration on the Theoretical Foundations of Deep Learning through awards DMS-2031883 and 814639, and a Weill Neurohub grant. YT was partially supported by NUS Start-up Grant A-8000448-00-00.

References

Bisbee, J. (2019). Barp: Improving mister p using bayesian additive regression trees. *American Political Science Review*, *113*(4), 1060–1065.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. J. (1984). *Classification and regression trees*. Chapman; Hall/CRC.

Carlson, C. J. (2020). Embarcadero: Species distribution modelling with bayesian additive regression trees in r. *Methods in Ecology and Evolution*, *11*(7), 850–858.

Carnegie, N. B. (2019). Comment: Contributions of Model Features to BART Causal Inference Performance Using ACIC 2016 Competition Data. *Statistical Science*, *34*(1), 90–93. https://doi.org/10.1214/18-STS682

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Chipman, H., George, E., & McCulloch, R. (2006). Bayesian ensemble learning. *Advances in neural information processing systems*, *19*.

Chipman, H. A., George, E. I., & McCulloch, R. E. (1998). Bayesian cart model search. *Journal of the American Statistical Association*, *93*(443), 935–948.

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, *4*(1), 266–298.

Denison, D. G., Mallick, B. K., & Smith, A. F. (1998). A bayesian cart algorithm. *Biometrika*, *85*(2), 363–377.

Dorie, V. (2022). Dbarts: Discrete bayesian additive regression trees sampler [R package version 0.9-22]. https://CRAN.R-project.org/package=dbarts

Dorie, V., Hill, J., Shalit, U., Scott, M., & Cervone, D. (2019). Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, *34*(1), 43–68.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of statistics*, 1189–1232.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 457–472.

Green, D. P., & Kern, H. L. (2010). Modeling heterogeneous treatment effects in large-scale experiments using bayesian additive regression trees. *The annual summer meeting of the society of political methodology*, 100–110.

Green, D. P., & Kern, H. L. (2012). Modeling heterogeneous treatment effects in survey experiments with bayesian additive regression trees. *Public opinion quarterly*, *76*(3), 491–511.

Hahn, P. R., Dorie, V., & Murray, J. S. (2019). Atlantic causal inference conference (acic) data analysis challenge 2017. *arXiv preprint arXiv:1905.09515*.

Hahn, P. R., Murray, J. S., & Carvalho, C. M. (2020). Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion). *Bayesian Analysis*, *15*(3), 965–1056.

He, J., & Hahn, P. R. (2021). Stochastic tree ensembles for regularized nonlinear regression. *Journal of the American Statistical Association*, 1–20.

Hill, J., Linero, A., & Murray, J. (2020). Bayesian additive regression trees: A review and look forward. *Annual Review of Statistics and Its Application*, *7*(1).

Hill, J., & Su, Y.-S. (2013). Assessing lack of common support in causal inference using bayesian nonparametrics: Implications for evaluating the effect of breastfeeding on children's cognitive outcomes. *The Annals of Applied Statistics*, 1386–1420.

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, *20*(1), 217–240.

Kern, H. L., Stuart, E. A., Hill, J., & Green, D. P. (2016). Assessing methods for generalizing experimental impact estimates to target populations. *Journal of research on educational effectiveness*, *9*(1), 103–127.

Levin, D. A., Peres, Y., & Wilmer, E. L. (2006). *Markov chains and mixing times*. American Mathematical Society. http://scholar.google.com/scholar.bib?q=info:3wf9IU94tyMJ:scholar.google.com/&output=citation&hl=en&as_sdt=2000&ct=citation&cd=0

Linero, A. R. (2017). A review of tree-based bayesian methods. *Communications for Statistical Applications and Methods*, *24*(6), 543–559.

Linero, A. R., & Yang, Y. (2018). Bayesian regression tree ensembles that adapt to smoothness and sparsity. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *80*(5), 1087–1110.

Liu, Y., Ročková, V., & Wang, Y. (2021). Variable selection with abc bayesian forests. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *83*(3), 453–481.

Murphy, K. P. (2007). Conjugate bayesian analysis of the gaussian distribution. *def*, *1*($2\sigma 2$), 16.

Murray, J. S. (2021). Log-linear bayesian additive regression trees for multinomial logistic and count regression models. *Journal of the American Statistical Association*, *116*(534), 756–769.

Olson, R. S., La Cava, W., Orzechowski, P., Urbanowicz, R. J., & Moore, J. H. (2017). Pmlb: A large benchmark suite for machine learning evaluation and comparison. *BioData mining*, *10*(1), 36.

Pace, R. K., & Barry, R. (1997). Sparse spatial autoregressions. *Statistics & Probability Letters*, *33*(3), 291–297.

Pratola, M. T. (2016). Efficient metropolis–hastings proposal mechanisms for bayesian regression tree models. *Bayesian analysis*, *11*(3), 885–911.

Pratola, M. T., Chipman, H. A., George, E. I., & McCulloch, R. E. (2020). Heteroscedastic bart via multiplicative regression trees. *Journal of Computational and Graphical Statistics*, *29*(2), 405–417.

Ročková, V., & Saha, E. (2019). On theory for bart. *The 22nd international conference on artificial intelligence and statistics*, 2839–2848.

Ročková, V., & van der Pas, S. (2020). Posterior concentration for bayesian regression trees and forests. *The Annals of Statistics*, *48*(4), 2108–2131.

Romano, J. D., Le, T. T., La Cava, W., Gregg, J. T., Goldberg, D. J., Ray, N. L., Chakraborty, P., Himmelstein, D., Fu, W., & Moore, J. H. (2020). Pmlb v1. 0: An open source dataset collection for benchmarking machine learning methods. *arXiv preprint arXiv:2012.00058*.

Sparapani, R. A., Logan, B. R., McCulloch, R. E., & Laud, P. W. (2016). Nonparametric survival analysis using bayesian additive regression trees (bart). *Statistics in medicine*, *35*(16), 2741–2753.

Starling, J. E., Murray, J. S., Carvalho, C. M., Bukowski, R. K., & Scott, J. G. (2020). Bart with targeted smoothing: An analysis of patient-specific stillbirth risk. *The Annals of Applied Statistics*, *14*(1), 28–50.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science* (Vol. 47). Cambridge university press.

Waldmann, P. (2016). Genome-wide prediction using bayesian additive regression trees. *Genetics Selection Evolution*, *48*(1), 1–12.

Wendling, T., Jung, K., Callahan, A., Schuler, A., Shah, N. H., & Gallego, B. (2018). Comparing methods for estimation of heterogeneous treatment effects using observational data from health care databases. *Statistics in medicine*, *37*(23), 3309–3324.

Wu, Y., Tjelmeland, H., & West, M. (2007). Bayesian cart: Prior specification and posterior simulation. *Journal of Computational and Graphical Statistics*, *16*(1), 44–66.

Yeager, D. S., Hanselman, P., Walton, G. M., Murray, J. S., Crosnoe, R., Muller, C., Tipton, E., Schneider, B., Hulleman, C. S., Hinojosa, C. P., et al. (2019). A national experiment reveals where a growth mindset improves achievement. *Nature*, *573*(7774), 364–369.

Yu, B., & Mykland, P. (1998). Looking at markov samplers through cusum path plots: A simple diagnostic idea. *Statistics and Computing*, *8*(3), 275–286.

Zhang, T., Geng, G., Liu, Y., & Chang, H. H. (2020). Application of bayesian additive regression trees for estimating daily concentrations of pm2. 5 components. *Atmosphere*, *11*(11), 1233.

A Appendix

A.1 Proof details for Sec 3

A.1.1 Proof of Lemma 3 (Conductance and likelihood ratio)

Proof. Let (v, τ) be the feature and threshold on which the root node splits in tree $\mathcal{T}$. Let $S = S(v, \tau)$ denote the set of tree structures whose root node splits on (v, τ). Then by assumption, we have $\mathcal{T} \in S$ and $\mathcal{T}', \mathcal{T}_0 \in S^c$, where $\mathcal{T}_0$ is the empty tree with only a root node. Assume for now that $\pi(S) \leq \frac{1}{2}$.

Because all possible moves for the Markov chain either add or remove a single terminal split, it is easy to see that S and S^c are connected by a single edge between $\mathcal{T}_0$ and $\mathcal{T}''$, the tree of depth 1 whose root node splits on (v, τ).

The conductance of S can thus be written as

$$\Phi(S) = \frac{\pi(\mathcal{T}_0)Q(\mathcal{T}''|\mathcal{T}_0)}{\pi(S)}. \tag{4}$$

Let $\psi(-|-)$ denote the transition kernel for the proposal step of the Metropolis-Hastings chain. We then have $\psi(\mathcal{T}''|\mathcal{T}_0) = \frac{1}{2md}$ (this comes from the probability of choosing a grow move multiplied by the probability of choosing the split (v, d).) The Metropolis-Hastings filter $\alpha(\mathcal{T}''|\mathcal{T}_0)$ is chosen precisely so that

$$\begin{aligned}\pi(\mathcal{T}_0)Q(\mathcal{T}''|\mathcal{T}_0) &= \pi(\mathcal{T}_0)\psi(\mathcal{T}''|\mathcal{T}_0)\alpha(\mathcal{T}''|\mathcal{T}_0) \\ &= \min\{\pi(\mathcal{T}_0)\psi(\mathcal{T}''|\mathcal{T}_0), \pi(\mathcal{T}'')\psi(\mathcal{T}_0|\mathcal{T}'')\}.\end{aligned}$$

Plugging in the formula for $\psi(\mathcal{T}''|\mathcal{T}_0)$ and continuing the equation gives

$$\pi(\mathcal{T}_0)Q(\mathcal{T}''|\mathcal{T}_0) \leq \frac{\pi(\mathcal{T}_0)}{2md}.$$

Next, notice that

$$\pi(S) = \sum_{\tilde{\mathcal{T}} \in S} \pi(\tilde{\mathcal{T}}) \geq \pi(\mathcal{T}).$$

As such, the conductance is bounded by a constant factor of the ratio of posterior probabilities as follows:

$$\Phi(S) \leq \frac{\pi(\mathcal{T}_0)}{2md\pi(\mathcal{T})}. \tag{5}$$

Since the posterior probability for any tree $\tilde{\mathcal{T}}$ can be written as

$$p(\tilde{\mathcal{T}}|\mathbf{X}, \mathbf{y}) = \frac{p(\tilde{\mathcal{T}})p(\mathbf{y}|\tilde{\mathcal{T}}, \mathbf{X})}{\sum_{\check{\mathcal{T}} \in \Omega} p(\check{\mathcal{T}})p(\mathbf{y}|\check{\mathcal{T}}, \mathbf{X})},$$

the right hand side of (5) is equal to

$$\frac{p(\mathcal{T}_0)}{2mdp(\mathcal{T})} \frac{p(\mathbf{y}|\mathcal{T}_0, \mathbf{X})}{p(\mathbf{y}|\mathcal{T}, \mathbf{X})}.$$

Since prior probabilities depend only on α, β, m, and d, the desired conclusion follows under the assumption that $\pi(S) \leq \frac{1}{2}$. If $\pi(S) > \frac{1}{2}$, we repeat the same string of calculations but with S replaced by S^c in (4) and with $\mathcal{T}$ replaced by $\mathcal{T}'$ thereafter.

□

A.1.2 Proof of Lemma 4 (Likelihood ratio and impurity decrease)

Proof. We first introduce some notation. Give the leaves of $\mathcal{T}$ an ordering, and denote them using $L_1, L_2, \ldots, L_b$. For $i = 1, \ldots, b$, let I_i denote the set of indices of the data points with $\mathbf{x}_j \in L_i$. Let n_i denote the count of data points in leaf i, i.e. $n_i := |I_i|$. Let $\mathcal{T}(x)$ denote the leaf node that contains an observation x, i.e. $\mathcal{T}(x_i) := L_j$ if $i \in I_j$. Finally, we will use C to denote absolute constants (independent of both the regression function f_0 and the tree $\mathcal{T}$) that may vary from line to line.

Step 1: Likelihood ratio to impurity decrease. By equation (42) in (Murphy, 2007), we have

$$\begin{aligned}\frac{p(\mathbf{y}|\mathcal{T}_0, \mathbf{X})}{p(\mathbf{y}|\mathcal{T}, \mathbf{X})} &= \sqrt{\frac{\prod_{i=1}^b (n_i\sigma^2 + a\sigma^2)}{(n\sigma^2 + a\sigma^2)(a\sigma^2)^{b-1}}} \exp\left\{ \frac{\sigma^2}{2a\sigma^2(n\sigma^2 + a\sigma^2)} \cdot Z^2 - \sum_{i=1}^b \frac{\sigma^2}{2a\sigma^2(n_i\sigma^2 + a\sigma^2)} Z_i^2 \right\} \\ &= \sqrt{\frac{\prod_{i=1}^b (n_i + a)}{(n+a)a^{b-1}}} \exp\left\{ \frac{1}{2a\sigma^2} \left(\frac{1}{n+a} Z^2 - \sum_{i=1}^b \frac{1}{n_i + a} Z_i^2 \right) \right\}\end{aligned}$$

where $Z = \sum_{j=1}^n y_j$ $Z_i = \sum_{j \in I_j} y_j$ for $i = 1, \ldots, b$. We would like to convert the expression in the exponential into an impurity decrease by removing the appearance of a in the denominators. To do so, we compute the bound:

$$\begin{aligned}\left| \left(\frac{1}{n+a} Z^2 - \sum_{i=1}^b \frac{1}{n_i + a} Z_i^2 \right) - \left(\frac{1}{n} Z^2 - \sum_{i=1}^b \frac{1}{n_i} Z_i^2 \right) \right| &= \left| \frac{a}{n(n+a)} Z^2 - \sum_{i=1}^b \frac{a}{n_i(n_i+a)} Z_i^2 \right| \\ &\leq \frac{an^2K^2}{n(n+a)} + \sum_{i=1}^b \frac{an_i^2K^2}{n_i(n_i+a)} \\ &\leq a(b+1)K^2. \qquad (6)\end{aligned}$$

Note that this is indeed an impurity decrease as

$$\begin{aligned}\frac{1}{n} Z^2 - \sum_{i=1}^b \frac{1}{n_i} Z_i^2 &= \frac{1}{n} Z^2 - \sum_{j=1}^n y_j^2 - \sum_{i=1}^b \left(\frac{1}{n_i} Z_i^2 - \sum_{\mathbf{x}_j \in L_i} y_j^2 \right) \\ &= -\sum_{j=1}^n (y_j - \bar{y})^2 + \sum_{i=1}^b \sum_{\mathbf{x}_j \in L_i} (y_j - \bar{y}_{L_i})^2,\end{aligned}$$

where $\bar{y}_{L_i} = n_i^{-1} Z_i$.

Step 2: Concentration of impurity decrease conditioned on counts. Condition on $n_1, \ldots, n_b$ and denote $\tilde{q}_i = \frac{n_i}{n}$ for $i = 1, \ldots, b$. The impurity decrease may be written as a variance:

$$-\frac{1}{n} Z^2 + \sum_{i=1}^b \frac{1}{n_i} Z_i^2 = -n \left(\sum_{i=1}^b \tilde{q}_i \bar{y}_{L_i}^2 - \left(\sum_{i=1}^b \tilde{q}_i \bar{y}_{L_i} \right)^2 \right).$$

We may rewrite the expression in parentheses on the right in matrix form:

$$\sum_{i=1}^{b} \tilde{q}_i \bar{y}_{L_i}^2 - \left(\sum_{i=1}^{b} \tilde{q}_i \bar{y}_{L_i} \right)^2 = \mathbf{w}^T \Big(\mathbf{I}_b - \sqrt{\tilde{\mathbf{q}}} \sqrt{\tilde{\mathbf{q}}}^T \Big) \mathbf{w}$$

where $w_i = \sqrt{\tilde{q}_i} \bar{y}_{L_i}$ for $i = 1, \ldots, b$.

Denote

$$\tilde{\Delta} = \sum_{i=1}^{b} \tilde{q}_i \mathbb{E}\{f_0(\mathbf{x}) | \mathbf{x} \in L_i\}^2 - \left(\sum_{i=1}^{b} \tilde{q}_i \mathbb{E}\{f_0(\mathbf{x}) | \mathbf{x} \in L_i\} \right)^2,$$

and notice that we may also write

$$\tilde{\Delta} = \mathbb{E}\{\mathbf{w}\}^T \Big(\mathbf{I}_b - \sqrt{\tilde{\mathbf{q}}} \sqrt{\tilde{\mathbf{q}}}^T \Big) \mathbb{E}\{\mathbf{w}\}.$$

We therefore have

$$\begin{aligned} &\left(\sum_{i=1}^{b} \tilde{q}_i \bar{y}_{L_i}^2 - \left(\sum_{i=1}^{b} \tilde{q}_i \bar{y}_{L_i} \right)^2 \right) - \tilde{\Delta} \\ &= (\mathbf{w} - \mathbb{E}\{\mathbf{w}\})^T \Big(\mathbf{I}_b - \sqrt{\tilde{\mathbf{q}}} \sqrt{\tilde{\mathbf{q}}}^T \Big) (\mathbf{w} - \mathbb{E}\{\mathbf{w}\}) + 2 (\mathbf{w} - \mathbb{E}\{\mathbf{w}\})^T \Big(\mathbf{I}_b - \sqrt{\tilde{\mathbf{q}}} \sqrt{\tilde{\mathbf{q}}}^T \Big) \mathbb{E}\{\mathbf{w}\}. \end{aligned} \tag{7}$$

We shall bound these two terms separately. To this end, we compute

$$\begin{aligned} \left\| \mathbf{I}_b - \sqrt{\tilde{\mathbf{q}}} \sqrt{\tilde{\mathbf{q}}}^T \right\|_F^2 &= \|\mathbf{I}_b\|_F^2 - 2\mathrm{Tr}\Big(\sqrt{\tilde{\mathbf{q}}}^T \mathbf{I}_b \sqrt{\tilde{\mathbf{q}}} \Big) + \left\| \sqrt{\tilde{\mathbf{q}}} \sqrt{\tilde{\mathbf{q}}^T} \right\|_F^2 \\ &= b - 1. \end{aligned}$$

Similarly,

$$\| \mathbf{I}_b - \sqrt{\tilde{\mathbf{q}}} \sqrt{\tilde{\mathbf{q}}}^T \|_{op} = 1.$$

Note also that, by Hoeffding's lemma (Vershynin, 2018), $\mathbf{w}$ has independent sub-Gaussian entries, each with squared sub-Gaussian norm at most

$$\left\| \sqrt{\tilde{q}_i} \bar{y}_{L_i} \right\|_{\psi_2}^2 \leq \frac{\tilde{q}_i K^2}{n_i} = \frac{K^2}{n}.$$

Applying the Hanson-Wright inequality (Vershynin, 2018), we therefore get

$$(\mathbf{w} - \mathbb{E}\{\mathbf{w}\})^T \Big(\mathbf{I}_b - \sqrt{\tilde{\mathbf{q}}} \sqrt{\tilde{\mathbf{q}}}^T \Big) (\mathbf{w} - \mathbb{E}\{\mathbf{w}\}) \leq \frac{CbK^2 \sqrt{\log n}}{n} \tag{8}$$

with probability at least $1 - 1/4n$.

Meanwhile, we have

$$\begin{aligned}\|\mathbb{E}\{\mathbf{w}\}\|_2^2 &= \sum_{i=1}^{b} \tilde{q}_i (\mathbb{E}\{f_0(\mathbf{x}) | \mathbf{x} \in L_i\})^2 \\ &\leq K^2.\end{aligned}$$

Hence, using Hoeffding's inequality, we have

$$(\mathbf{w} - \mathbb{E}\{\mathbf{w}\})^T \left(\mathbf{I}_b - \sqrt{\tilde{\mathbf{q}}}\sqrt{\tilde{\mathbf{q}}}^T\right)\mathbb{E}\{\mathbf{w}\} \leq CK^2\sqrt{\frac{\log n}{n}} \tag{9}$$

with probability at least $1 - 4/n$. Plugging (8) and (9) into (7) gives us

$$\left(\sum_{i=1}^{b} \tilde{q}_i \bar{y}_{L_i}^2 - \left(\sum_{i=1}^{b} \tilde{q}_i \bar{y}_{L_i}\right)^2\right) - \tilde{\Delta} \leq CK^2\sqrt{\log n}\left(\frac{b}{n} + \frac{1}{\sqrt{n}}\right) \tag{10}$$

with probability at least $1 - 2/n$.

Step 3: Decondition on counts. Using the law of total variance, write

$$\begin{aligned}\Delta &= \text{Var}\{f_0(\mathbf{x})\} - \mathbb{E}\{\text{Var}\{f_0(\mathbf{x})|\mathcal{T}(\mathbf{x})\}\} \\ &= \text{Var}\{\mathbb{E}\{f_0(\mathbf{x}) \mid \mathcal{T}(\mathbf{x})\}\} \\ &= \sum_{i=1}^{b} q_i \mathbb{E}\{f_0(\mathbf{x}) \mid \mathbf{x} \in L_i\}^2 - \left(\sum_{i=1}^{b} q_i \mathbb{E}\{f_0(\mathbf{x}) \mid \mathbf{x} \in L_i\}\right)^2\end{aligned}$$

where $q_i = \mathbb{P}\{y \in L_i\}$ for $i = 1, \ldots, b$. Since $n\tilde{\mathbf{q}}$ is a multinomial distribution with parameters n and $\mathbf{q}$, we may apply Hoeffding's inequality two more times to get

$$\left|\sum_{i=1}(\tilde{q}_i - q_i)\mathbb{E}\{f_0(\mathbf{x}) \mid \mathbf{x} \in L_i\}\right| \leq CK\sqrt{\frac{\log n}{n}}$$

and

$$\left|\sum_{i=1}(\tilde{q}_i - q_i)\mathbb{E}\{f_0(\mathbf{x}) \mid \mathbf{x} \in L_i\}^2\right| \leq CK^2\sqrt{\frac{\log n}{n}} \tag{11}$$

jointly with probability at least $1 - 1/2n$. On this event, we also have

$$\begin{aligned}&\left|\left(\sum_{i=1}^{b} \tilde{q}_i \mathbb{E}\{f_0(\mathbf{x}) \mid \mathbf{x} \in L_i\}\right)^2 - \left(\sum_{i=1}^{b} q_i \mathbb{E}\{f_0(\mathbf{x}) \mid \mathbf{x} \in L_i\}\right)^2\right| \\ = &\left|\sum_{i=1}^{b}(\tilde{q}_i - q_i)\mathbb{E}\{f_0(\mathbf{x}) \mid \mathbf{x} \in L_i\}\right| \cdot \left|\sum_{i=1}^{b}(\tilde{q}_i + q_i)\mathbb{E}\{f_0(\mathbf{x}) \mid \mathbf{x} \in L_i\}\right| \\ &\leq 2K\left|\sum_{i=1}^{b}(\tilde{q}_i - q_i)\mathbb{E}\{f_0(\mathbf{x}) \mid \mathbf{x} \in L_i\}\right| \\ &\leq CK^2\sqrt{\frac{\log n}{n}}.\end{aligned} \tag{12}$$

Using (11) and (12), we get

$$\left|\tilde{\Delta} - \Delta\right| \leq CK^2\sqrt{\frac{\log n}{n}}.$$

Conditioning on this event and further conditioning on the $1 - 1/2n$ event guaranteeing (7) then gives us:

$$\begin{aligned}\left|\left(\frac{1}{n+a}Z^2 - \sum_{i=1}^{b}\frac{1}{n_i+a}Z_i^2\right) + n\Delta\right| &\leq a(b+1)K^2 + CK^2\sqrt{\log n}\left(b+\sqrt{n}\right) + CK^2\sqrt{n\log n} \\ &\leq CK^2\sqrt{\log n}\left(\sqrt{n}+b\right)\end{aligned}$$

since it only makes sense to choose $a \leq 1$.

Finally, we have

$$\begin{aligned}\log\sqrt{\frac{\prod_{i=1}^{b}(n_i+a)}{(n+a)a^{b-1}}} &= \frac{1}{2}\sum_{i=1}^{b}\log(n_i+a) - \frac{1}{2}\log(n+a) - (b-1)\log a \\ &\leq b\log(n/a).\end{aligned}$$

□

A.2 Additional Simulation Results

A.2.1 Simplified BART RMSE Kernel Density

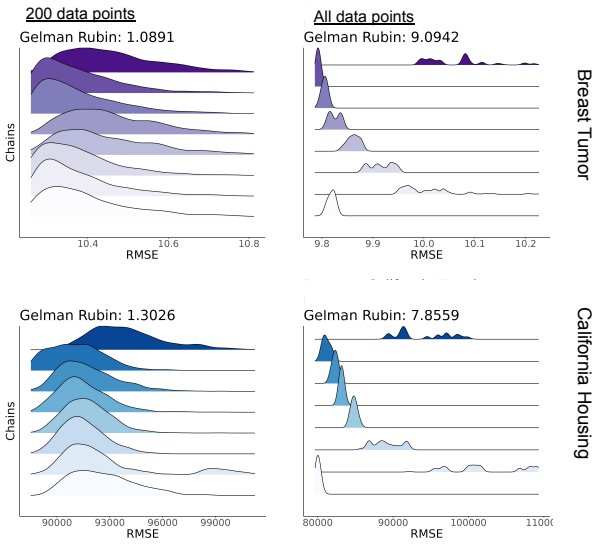

Figure 5: Kernel density plots of simplified BART RMSE values across different chains and data sets and sample sizes. The number of data points used increases from the left to right column. Different rows correspond to different datasets.

A.2.2 Cusum Diagnostics for BART and simplified BART

In this section we visualize the BART and simplified BART mixing using the cusum path plot proposed by Yu and Mykland, 1998. We now describe the cusum plot formally: Suppose we have L samples from J different MCMC chains obtained after a burn-in period. Let x_{ij} denote the RMSE value of the ith sample from the jth chain, $\bar{x}_j$ denote the mean RMSE sample from the jth chain, and let $S_t^{(j)} = \sum_{i=1}^{t}(x_{ij} - \bar{x}_j)$ be the cumulative sum of deviations from the mean. In a cusum plot, we plot $S_t^{(j)}$ against t. A "hairy" cusum plot, in which the cumulative sum varies randomly around a mean of zero, represents fast mixing. A smooth cusum plot, represents shifts in the sampling process, and is an indicating of slow mixing. In our case, a smooth cusum path would translate into a sub-sequence of tree functions within a given chain, whose RMSE value is higher or lower than the chain average.

Figs. 6 and 7 show cusum plot for BART and simplified BART respectively, across 8 different chains.

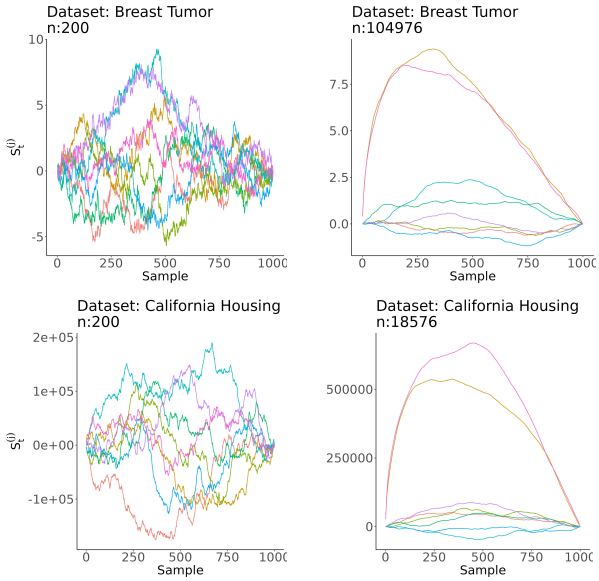

Figure 6: BART cusum path plot become smoother as more data points are used. $S_t^{(j)}$ is plotted against t for 8 different chains. The number of data points used increases from the left to right column. Different rows correspond to different datasets, and different colors correspond to different chains.

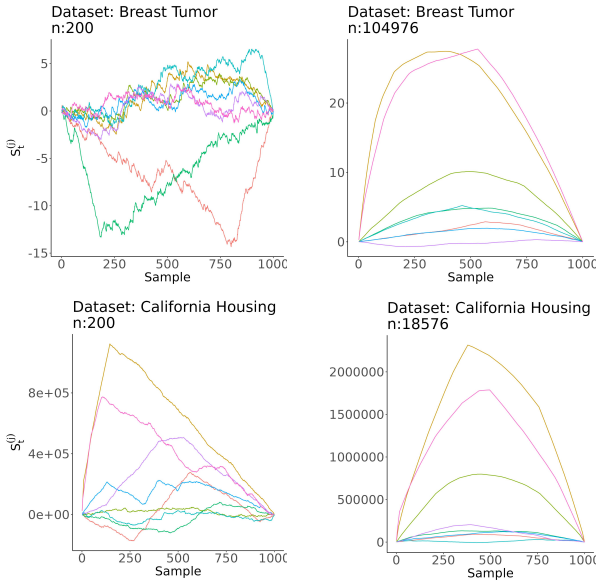

Figure 7: Simplified BART cusum path plot become smoother as more data points are used. $S_t^{(j)}$ is plotted against t for 8 different chains. The number of data points used increases from the left to right column. Different rows correspond to different datasets, and different colors correspond to different chains.

A.3 Reversing the Root Split

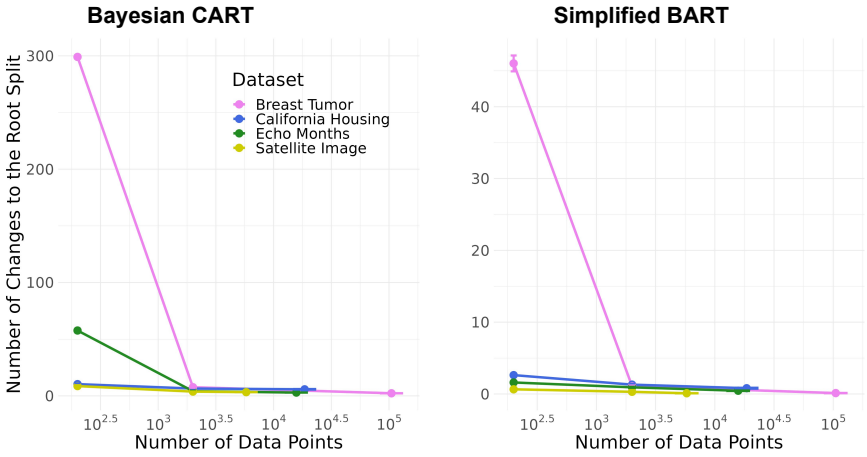

Figure 8: The number of MCMC iterations during which the feature on which the root node splits changes is small, and decreases as the number of training data points increases. Right panel shows the results for B-CART and left for simplified BART . Error bars are calculated over 160 different runs using the same training data.

A.3.1 Simplified BART Root Node Split Indices

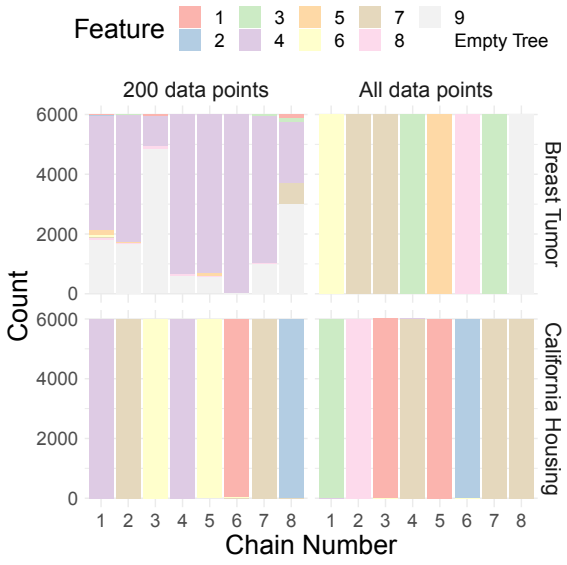

Figure 9: Number of iterations on which the root node splits on each feature for different chains on several datasets. When n is large, almost all MCMC samples for a single chain of simplified BART have a root split on the same feature, whose index varies across chains. The different colors correspond to the different features, and empty tree refers to a single leaf (no splits).