STOCHASTIC TREE ENSEMBLES FOR REGULARIZED NONLINEAR REGRESSION

By Jingyu He and P. Richard Hahn

The University of Chicago

Arizona State University

February 8, 2020

This paper develops a novel stochastic tree ensemble method for nonlinear regression, which we refer to as XBART, short for Accelerated Bayesian Additive Regression Trees. By combining regularization and stochastic search strategies from Bayesian modeling with computationally efficient techniques from recursive partitioning approaches, the new method attains state-of-the-art performance: in many settings it is both faster and more accurate than the widely-used XGBoost algorithm. Via careful simulation studies, we demonstrate that our new approach provides accurate point-wise estimates of the mean function and does so faster than popular alternatives, such as BART, XGBoost and neural networks (using Keras). We also prove a number of basic theoretical results about the new algorithm, including consistency of the single tree version of the model and stationarity of the Markov chain produced by the ensemble version. Furthermore, we demonstrate that initializing standard Bayesian additive regression trees Markov chain Monte Carlo (MCMC) at XBART-fitted trees considerably improves credible interval coverage and reduces total run-time.

Keywords and phrases: Tree ensembles; Machine learning; Markov chain Monte Carlo; Regression trees; Supervised learning; Bayesian

CONTENTS

1. Introduction. Tree-based algorithms for supervised learning, such as Classification and Regression Trees (CART) (Breiman et al., 1984), random forests (Breiman, 1996, 2001), adaBoost (Freund and Schapire, 1997), and gradient boosting (Breiman, 1997; Friedman, 2001, 2002), are widely used for applied supervised learning. As a whole, these methods are popular in applied settings due to their speed and accuracy in mean estimation and out-of-sample prediction tasks. One limitation of such methods is their well-known sensitivity to tuning parameters, which require costly cross-validation to optimize. Bayesian additive regression trees (BART) (Chipman et al., 2007, 2010) is a popular model-based alternative that is often more accurate than other tree-based methods; specifically, BART boasts valuable robustness to the choice of tuning-parameters. However, relative to random forests and boosting, BART's wider adoption has been slowed by its more severe computational demands, owing to its reliance on a random walk Metropolis-Hastings Markov chain Monte Carlo (MCMC) algorithm.

Despite this limitation, BART has inspired a considerable body of research in recent years. Applications to causal inference (Hill, 2011; Hahn et al., 2020; Logan et al., 2019; Starling et al., 2019), extensions to novel model settings (Murray, 2017; Linero and Yang, 2018; Linero et al., 2019; Kindo et al., 2016; Pratola et al., 2017; Starling et al., 2018; van der Pas and Ročková, 2017), computational innovations (Pratola et al., 2014; Pratola, 2016), and posterior consistency theory (Ročková and Saha, 2019; Rocková, 2019) are some of the notable active research areas. For a more comprehensive review of this literature, see Linero (2017) and Hill et al. (2020). Important precursors of the BART model include Chipman et al. (1998), Denison et al. (1998), and Gramacy and Lee (2008).

In this paper, we contribute to this growing literature by developing a novel stochastic tree ensemble method that combines the hyper-parameter robustness of BART with the efficient recursive computational techniques of traditional tree-based methods. Specifically, we propose a novel tree splitting criterion derived from an integrated-likelihood calculation and suggest a parameter-sampling approach (as opposed to a bootstrapping approach, as in random forests) for avoiding over-fitting. These modifications lead to a tree sampling algorithm that is substantially faster than BART while retaining its state-of-the-art predictive accuracy. This new approach to Bayesian tree models both leads to a substantial speed-up of model fitting and also opens the door for new theoretical results adapted from the literature on random forests (Scornet et al., 2015).

After introducing the general algorithm, which we call Accelerated Bayesian Additive Regression

Trees (XBART), we then specialize it to Gaussian nonlinear regression. We prove that the sampling algorithm produces a finite-space Markov chain with stationary distribution, and show that the recent theoretical results of consistency for random forests (Scornet et al., 2015) can be modified to apply to XBART. A wide range of simulation studies demonstrate the efficacy of the new approach. Furthermore, XBART works not only as a stand-alone machine learning algorithm, but can also be used to initialize a BART MCMC sampler (warm-start BART), resulting in faster fully Bayesian inference with improved posterior exploration as indicated by posterior credible intervals (for the mean function) with better coverage (for a fixed number of posterior samples).

2. XBART Framework. Let y denote a continuous outcome in $\mathbb{R}^1$. Our goal is to predict y by a length p covariate vector $\mathbf{x} = (\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(p)})$. We demonstrate the general framework of the algorithm in this section; details of the regression setting are introduced in the following section. We begin with the algorithm for fitting a single tree, then proceed to tree ensembles, or forests.

2.1. *Fitting a single tree recursively and stochastically.* A tree T_l $(1 \leq l \leq L)$ is a set of decision rules defining a rectangular partition of the covariate space $\{\mathcal{A}_{l1}, \cdots, \mathcal{A}_{lB_l}\}$. Each terminal node $\mathcal{A}_{lb}$ is associated with a vector of leaf parameter μ_{lb}. We denote a tree $g(\mathbf{x}; T_l, \mu_l)$ where $\mu_l = (\mu_{l1}, \cdots, \mu_{lB_l})$ is a vector of all leaf parameters. Each pair of (T_l, μ_l) parameterizes a step function on covariate space,

$$g(\mathbf{x}; T_l, \mu_l) = \mu_{lb}, \quad \text{if } \mathbf{x} \in \mathcal{A}_{lb}.$$

Figure 1 depicts a regression tree. The left panel shows a decision rule structure, and the right panel plots the corresponding partition of the space as well as the associated leaf parameters. Ideally, a tree partitions the space into fine irregular mesh where outcome observations within each leaf node (defining a hyperrectangle in covariate space) are nearly homogeneous. Predicting the outcome of a new observation then follows according to the leaf parameter associated with the node it falls within.

Usually, a tree algorithm learns the partition by recursively partitioning the data set one $\mathbf{x}$ variable at a time, and partitions the child nodes similarly as parent node until reaching terminating conditions. Algorithm 1 gives the pseudocode for the essential step of fitting a tree recursively.

Most popular tree-based methods deploy Algorithm 1 varying in terms of their split criteria and stopping conditions. Essentially, split criteria are functions of a cutpoint, measuring homogeneity within the two child nodes produced by the implied split. CART (Breiman et al., 1984), for instance,

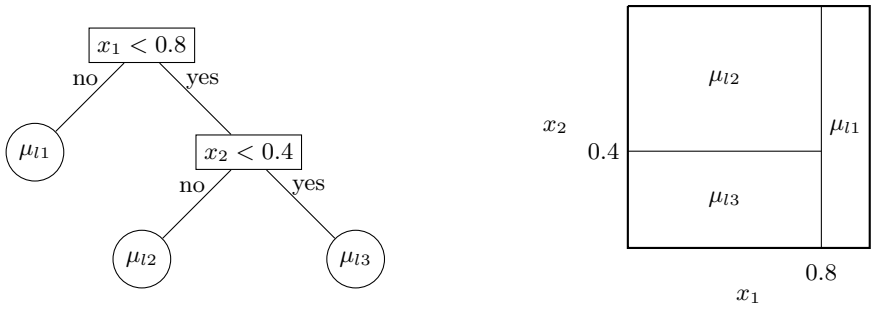


Fig 1: An illustration of tree in two dimensional space.

Algorithm 1 Pseudocode of growing a tree recursively.

1: Start at a root node.
2: Select a cutpoint by some pre-specified split criterion, then partition the root node into two child nodes according to the selected decision rule.
3: If pre-specified stop conditions are satisfied, stop the algorithm and estimate leaf parameters based on data in each leaf node. Otherwise, apply step 2 to each child node.

uses mean squared error to define its split criterion. Note also that CART and other recursive tree algorithms *optimize* their split criteria over the set of all cutpoint candidates in order to select a cutpoint. Frequently used stop conditions include maximum depth of a tree, minimal number of data observations within leaf node or a threshold for percent change of split criterion from parent to child nodes. Despite its simplicity and elegance, the exhaustive search approach tends to grow a tree unnecessarily deep, thereby over-fitting the data, thus pruning (the merge of some leaf nodes via a bottom up process) after model fitting is usual necessary to further throttle model complexity. In contast, BART provides a Bayesian perspective on tree models by using a regularization prior on tree space which preferences smaller trees. The excellent empirical performance of BART suggests that this prior regularization is beneficial. However, BART does not fit trees recursively but rather explores the posterior distribution over trees via a Bayesian backfitting MCMC scheme, which is computationally intensive, especially for large scale data. This computational burden hampers BART's usefulness on large scale data.

Inspired by both CART and BART, our proposed XBART framework combines strength from each. XBART is a sum of trees in which each individual tree is grown according to a model-based split criterion derived from an integrated-likelihood, but draws a cutpoint with probability proportional to split criterion. Furthermore, XBART stochastically terminates the growing process, the so-called *no-split* option; this allows trees to stop growing before reaching stop conditions and helps to prevent over-fitting.

We briefly clarify notation before turning to details of the algorithm. The predictor matrix $\mathbf{X}$, with dimension $n \times p$, defines a set of splitting rule candidates, denoted $\mathcal{C}$, which are indexed as (j, k) where $j = 1, \ldots p$ indexes a variable (column) of $\mathbf{X}$ and k indexes a set of candidate cutpoints. Let $|\mathcal{C}|$ denote the total number of candidate splitting rules (cutpoints). Let Φ denote prior hyper-parameters and Ψ denote model parameters, which are both considered given and fixed when growing a single tree (a distinction that will be clarified in specific examples).

Inspired by the Bayesian approach, we define a likelihood $L(\mathrm{y}_b; \mu_b, \Psi_b)$ on *one* leaf node b with leaf-specific parameter μ_b and other model parameters Φ_b (given and fixed during the tree growing process). In the following text, we omit subscript b for simplicity. For instance, the likelihood can be Gaussian for regression, details of which are given in section 3. The leaf parameter μ is given a prior $\pi(\mu \mid \Phi)$. We derive our split criterion by integrating out the leaf parameter μ:

$$m(s \mid \Phi, \Psi) := \int L(\mathrm{y}; \mu, \Psi)\pi(\mu \mid \Phi)d\mu,$$

where s represents sufficient statistics of data y falling in the current node.

A cutpoint (j, k) partitions the current node to left and right child nodes, with sufficient statistics s^l_{jk} and s^r_{jk} calculated based on y respectively. Assuming that observations in separate leaf nodes are independent, the joint integrated-likelihood is simply the product of the two sides

$$m(s^l_{jk} \mid \Phi, \Psi)m(s^r_{jk} \mid \Phi, \Psi), \tag{1}$$

which defines the split criterion for cutpoint (j, k). Furthermore, the split criterion for *no-split* is defined as

$$|\mathcal{C}| \left(\frac{(1+d)^\beta}{\alpha} - 1 \right) m(s^\emptyset \mid \Phi, \Psi)$$

where d is depth of the current node, $s^\emptyset$ represents sufficient statistics on current node and α, β are hyper-parameters. The weight of no-split increases significantly as a tree grows deeper, strongly penalizing deep trees and thereby favoring "weak learners" in the parlance of the boosting literature. Besides the no-split option, traditional stopping conditions are also imposed, such as setting a maximum depth or a minimal number of observations per node.

Once the split criterion has been evaluated at all cutpoint candidates, as well as the no-split option (retained from the previous split), a cutpoint is randomly sampled with probability proportional to its split criterion value. Unlike non-model-based tree algorithms, XBART's split criterion has a natural probabilistic interpretation as it is derived as an integrated-likelihood and sampling follows

according to Bayes rule. Indeed, the prior probability of the no-split option (after normalizing with respect to all other cutpoint candidates) matches that of the tree prior used in standard BART (Chipman et al., 2010).

THEOREM 1. *The a priori probability of splitting at a node of depth d implied by algorithm 2 (grow-from-root) is $\alpha(1+d)^{-\beta}$.*

PROOF. The proof is by direct calculation. Ignore the data contribution from the marginal likelihood function $m(\cdot)$ by setting these terms to 1 in the expressions in line 8. Accordingly, the probability of any cutpoint $(j,k) \in \mathcal{C}$ has prior probability proportional to 1 and the prior probability of no-split is proportional to $|\mathcal{C}|\left(\frac{(1+d)^{\beta}}{\alpha}-1\right)$. Therefore, the total weight given to splitting is $\sum_{\mathcal{C}} 1 = |\mathcal{C}|$ and normalizing gives the prior probability of splitting as

$$\frac{\text{split weight}}{\text{split weight} + \text{no split weight}} = \frac{|\mathcal{C}|}{|\mathcal{C}|\left(\frac{(1+d)^{\beta}}{\alpha}-1\right)+|\mathcal{C}|} = \alpha(1+d)^{-\beta}.$$

□

Remark We observe that sampling a cutpoint stochastically rather than optimizing the split criterion substantially improves performance, based on simulation experiments. Intuitively, sampling cutpoints rather than optimizing them helps alleviate over-fitting and encourages wider exploration of the tree space.

Once the no-split option is selected or stopping conditions are met, the current node becomes a terminating node or a *leaf*. The leaf parameter μ_{lb} is then updated based on data in the current leaf by standard Bayesian posterior sampling using conjugate likelihood and prior. Algorithm 2 presents the GrowFromRoot function that grows a single tree for XBART.

In summary, XBART implies the same prior probability as BART by the design of the split criterion, while fitting a tree recursively. This framework enjoys great flexibility of potential applications. In this paper we focus on the case where the marginal likelihood arises from a Gaussian mean regression model (section 3), but it is straightforward to substitute an integrated-likelihood function $m(s \mid \Phi, \Psi)$ from other models.

2.2. *Forest.* A forest, as the name sugggests, is an ensemble of trees. Ensemble learning is a widely used technique to combine multiple learning algorithms to improve the overall prediction accuracy. Random forests (Breiman, 1996, 2001) takes an average of trees fitting bootstrap resampled

Algorithm 2 GrowFromRoot

procedure GFR(y, $\mathbf{X}$, Ψ, Φ, d, T, `node`)
outcome Modifies T by adding nodes and sampling associated leaf parameters.
    $s^{\emptyset} \leftarrow s(y, \mathbf{X}, \Psi, \mathcal{C}, \text{all})$. ▷ Compute sufficient statistic of not splitting.
    for $(j,k) \in \mathcal{C}$ **do** ▷ Calculated recursively in a single sweep of the data per variable.
        $s^{l}_{jk} \leftarrow s(y, \mathbf{X}, \Psi, \mathcal{C}, j, k, \text{left})$. ▷ Compute sufficient statistic of left candidate node.
        $s^{r}_{jk} \leftarrow s(y, \mathbf{X}, \Psi, \mathcal{C}, j, k, \text{right})$. ▷ Compute sufficient statistic of right candidate node.
    end for
    Sample cutpoint (j,k) proportional to integrated likelihoods

$$m(s^{l}_{jk})m(s^{r}_{jk})$$

or

$$|\mathcal{C}|\left(\frac{(1+d)^{\beta}}{\alpha}-1\right)m(s^{\emptyset})$$

for the no-split option.
    if no-split is selected or stop conditions are reached **then**
        $\theta_{\texttt{node}} \leftarrow$ SampleParameters($s^{\emptyset}$)
        return.
    else
        Create two new nodes, denoted `left_node` and `right_node`, and growing T by designating them as the current node's (`node`) children.
        Sift the data into left and right parts, according to the selected cutpoint $x_{ij'} \leq x^{*}_{kj}$ and $x_{ij'} > x^{*}_{kj}$, respectively, where x^{*}_{kj} is the value corresponding to the sampled cutpoint (j,k).
        GFR(y_{left}, $\mathbf{X}_{\text{left}}$, Ψ, Φ, $d+1$, T, `left_node`)
        GFR(y_{right}, $\mathbf{X}_{\text{right}}$, Ψ, Φ, $d+1$,T, `right_node`)
    end if
end procedure

data; adaBoost (Freund and Schapire, 1997), gradient boosting (Breiman, 1997; Friedman, 2001, 2002) and BART (Chipman et al., 2010) explicitly fit a sum of trees. XBART Gaussian nonlinear regression takes the same sum of trees form as BART does:

$$f(\mathbf{x}) = \sum_{i=1}^{L} g_l(\mathbf{x}; T_l, \mu_l), \tag{2}$$

where step function $g_l(\mathbf{x}; T_l, \mu_l)$ denotes a tree defined by partition T_l and corresponding leaf parameters μ_l.

The stochastic tree ensemble method proceeds similarly to an MCMC algorithm. Suppose we draw I samples (sweeps) of forests, and each forest contains L trees. When updating the h-th tree in the *iter*-th iteration, a new tree is grown to fit the *partial* residuals $\text{r}^{(iter)}_{h}$, which are defined as the partial residual of the target (y) after subtracting off the contribution of all the other trees. Specifically, the partial residuals are defined as

$$\text{r}^{(iter+1)}_{h} \equiv \text{y} - \sum_{h'<h} g(\mathbf{X}; T_{h'}, \mu_{h'})^{(iter+1)} - \sum_{h'>h} g(\mathbf{X}; T_{h'}, \mu_{h'})^{(iter)},$$

while the total residual is taken with respect to all trees

$$\tilde{\mathrm{r}}_h^{(iter+1)} \equiv \mathrm{r}_h^{(iter+1)} - g(\mathbf{X}; T_h, \mu_h)^{(iter+1)}.$$

Algorithm 3 draws I samples of the forest; we refer to one pass of the algorithm, sampling each tree, as a sweep. Every tree is updated by algorithm 2 GrowFromRoot in each iteration, where the "data" are the partial residuals as calculated at the current iteration. Extra model-dependent non-tree parameters Ψ are updated in between sampling each tree; specifically, the residual standard deviation σ is sampled after each tree. Details are summarized in the following sections.

Algorithm 3 Accelerated Bayesian Additive Regression Trees (XBART)

procedure XBART($\mathrm{y}, \mathbf{X}, \Phi, L, I$)
output I posterior draws of a forest (and associated leaf parameters) comprising L trees.
    Initialize Ψ, partial fit R_h^{iter}.
    for *iter* in 1 to I **do**
        for h in 1 to L **do**
            Create `new_node`.
            Initialize tree T_h^{iter} consisting only of `new_node`.
            GFR(r_h^{iter}, $\mathbf{X}$, Ψ, Φ, T_h^{iter}, $d = 0$, `new_node`)
            Update r_{h+1}^{iter} (or r_1^{iter+1} if $h = L$), the target to fit for the next tree and full residual $\tilde{\mathrm{r}}_h^{(iter+1)}$.
            Sample non-tree parameters of Ψ, probably based on the full residual $\tilde{\mathrm{r}}_h^{(iter+1)}$.
        end for
    end for
end procedure

Next, we study theoretical properties of Algorithm 3.

THEOREM 2. *The algorithm sampling $F = \{T_h\}_{1 \le h \le L}$ is a finite-state Markov chain with stationary distribution.*

PROOF. We consider the process $F = \{T_h\}_{1 \le h \le L}$. Leaf parameters $\mu = \{\mu_h\}_{1 \le h \le L}$ are updated conditional on forest F based on standard conjugate Bayesian posterior draws and are not to be regarded as part of the Markov chain of the forest.

First, observe that each tree has a maximum depth and all cutpoint candidates are defined on a finite covariate matrix $\mathbf{X}$. Therefore a single tree has finite states. The forest is an ensemble of a finite number of trees, thus has a finite number of states as well. The probability of the GrowFromRoot algorithm drawing a single tree is a product of the probabilities of drawing specific cutpoints at each node, thus $p(T_j \mid T_{-j}, \mu_{-j}) > 0$. In addition, the GrowFromRoot algorithm updates T_h^{iter} fitting r_h^{iter}, which is defined by trees and leaf parameters with subscript $1 < j < h$ in *iter*-th sweeps and $h + 1 < j < L$ in $(iter - 1)$-th sweeps. Therefore, the forest process is a finite-state Markov chain.

Second, we claim that because the split criterion is defined by an integrated likelihood, it has non-zero evaluations for all cutpoint candidates (including the no-split option) given fitting data r_h^{iter}. Let $T_{-j} = \{T_h\}_{1\leq h\leq L}/T_j$ and $\mu_{-j} = \{\mu_h\}_{1\leq h\leq L}/\mu_j$ be trees and leaf parameters excepting the j-th one, respectively. We have

$$p(T_j \mid T_{-j}) = \int\int p(T_j \mid \mathrm{y}, T_{-j}, \mu_{-j}, \Psi) f(\mu_{-j} \mid \mathrm{y}, T_{-j}) f(\Psi) d\Psi d\mu_{-j} > 0, \tag{3}$$

since $f(\mu_{-j} \mid \mathrm{y}, T_{-j})$, the usual Bayesian posterior of drawing leaf parameters, is non-zero. Note that this integral arises via the algorithmic implementation that draws T_j by first drawing μ_{-j} and Ψ, and then drawing T_j via GrowFromRoot.

Lastly, consider the transition probability between any two forests, $F^1 = \{T_h^1\}_{1\leq h\leq L}$ and $F^2 = \{T_h^2\}_{1\leq h\leq L}$. Observe that there is at least one way to transition from one forest to another, which is to regrow each tree and replace them one by one. Therefore, we have

$$P(F_2 \mid F_1) \geq \prod_{j=1}^{L} p\left(T_j^2 \mid \{T_h^2\}_{1\leq h<j}, \{T_h^1\}_{j+1\leq h<L}\right) > 0,$$

where the last inequality is from equation (3).

In conclusion, the forest process has a finite number of possible states, and the transition probability between any two states is positive. Therefore, by standard results, it is a finite-state Markov chain with a stationary distribution. □

To obtain a prediction from XBART, we take posterior averages as if the sampled trees were draws from a standard Bayesian Monte Carlo algorithm. That is, given I iterations of the algorithm, the final $I - I_0$ samples are used to compute a point-wise average function evaluation, where $I_0 < I$ denotes the length of the burn-in period. We recommend $I = 40$ and $I_0 = 15$ for routine use. The final estimator is therefore expressible as

$$\bar{f}(\mathbf{X}) = \frac{1}{I - I_0} \sum_{k>I_0}^{I} f^{(k)}(\mathbf{X}).$$

where $f^{(k)}$ denotes a sample of the forest, as in equation 2, drawn by algorithm 3. This would correspond to the Bayes optimal estimator under mean squared error estimation loss, if we regard our samples as coming from a proper posterior distribution. As the GrowFromRoot strategy is not a proper full conditional, this estimator must be considered an approximation of some sort. Nonetheless, simulation results strongly suggest that the approximation is adequate. In subsequent

sections we also provide some theory suggesting that XBART is a consistent estimator in its own right.

As for quantification of estimation uncertainty, note that with only $I = 40$ sweeps, the XBART posterior would certainly understate the estimation uncertainty even if we had independent Monte Carlo draws from a valid posterior distribution. However, the standard BART MCMC is probably not mixing well in most contexts, either, and yet still provides useful, if approximate, uncertainty quantification. It is noteworthy that experiments with a version of an XBART estimate based on only the final sweep (that is, letting $I - I_0 = 1$) perform worse than XBART with $I - I_0 > 1$, suggesting that the posterior exploration, while partial, is still beneficial. In any event, the next section describes how to combine XBART with standard BART MCMC to get full Bayesian inference that appears to be both faster and more accurate than BART MCMC alone.

2.3. *Warm-start BART MCMC.* Standard BART MCMC (Chipman et al., 2010) initializes each tree at the root (i.e., a tree only one node) and explores the posterior over trees via a random-walk Metropolis-Hastings algorithm. This approach works surprisingly well in practice, but it is natural to wonder if it takes unnecessarily long to find favorable regions in tree space. Because XBART provides a fast approximation to the BART posterior, initializing BART MCMC at XBART trees rather than roots is a promising strategy to help speed convergence and also to accelerate posterior exploration by running multiple chains. In fact, we find that this approach yields improved point estimation and posterior credible intervals with substantially higher pointwise frequentist coverage of the mean function, and in a fraction of the total run time. These simulation results are reported in section 5.2.

2.4. *Adaptive variable importance weights.* Our XBART implementation strikes an intermediate balance between the local BART updates, which randomly consider one variable at a time, and the all-variables Bayes rule described above. Specifically, we consider only $m \leq V$ variables at a time when sampling each splitting rule. Rather than drawing these variables uniformly at random as is done in random forests, we introduce a parameter vector w which denotes the prior probability that a given variable is chosen to be split on, as suggested in Linero (2018). Before sampling each splitting rule, we randomly select m variables (without replacement) with probability proportional to w.

2.5. *Computational strategies.* In the remainder of this section, we catalogue implementation details that improve the computational efficiency of the algorithm. These implementational details serve to make the algorithm competitive with state-of-the-art supervised learning algorithms, such as XGBoost. These particular strategies, such as variable presorting and careful handling of categorical covariates, are inapplicable in the standard BART MCMC and XBART's ability to incorporate them is the basis of its improved performance.

2.5.1. *Pre-sorting predictor variables.* Observe that the XBART split criterion depends on sufficient statistics only, namely the sum of the observations in a node (that is, at a given level of the recursion). An important implication of this, for computation, is that with sorted predictor variables, the various cutpoint integrated likelihoods can be computed rapidly via a single sweep through the data (per variable), taking cumulative sums. Let $\mathbf{O}$ denote the V-by-n array such that o_{vh} denotes the index, in the data, of the observation with the h-th smallest value of the v-th predictor variable x_v. Then, taking the cumulative sums gives

$$s(\leq, v, c) = \sum_{h \leq c} \mathrm{r}_{o_{vh}}$$

and

$$s(>, v, c) = \sum_{h=1}^{n} r_{lh} - s(\leq, v, c).$$

The subscript l on the residual indicates that these evaluations pertain to the update of the lth tree.

The above formulation is useful if the data can be presorted and, furthermore, the sorting can be maintained at all levels of the recursive tree-growing process. To achieve this, we must "sift" each of the variables before passing to the next level of the recursion. Specifically, we form two new index matrices $\mathbf{O}^{\leq}$ and $\mathbf{O}^{>}$ that partition the data according to the selected cutpoint. For the selected split variable v and selected split c, this is automatic: $O_v^{\leq} = O_{v,1:c}$ and $O_v^{>} = O_{v,(c+1):n}$. For the other $V-1$ variables, we sift them by looping through all n available observations, populating $O_q^{\leq}$ and $O_q^{>}$, for $q \neq v$, sequentially, with values o_{qj} according to whether $x_{vo_{qj}} \leq c$ or $x_{vo_{qj}} > c$, for $j = 1, \ldots, n$.

Because the data is processed in sorted order, the ordering will be preserved in each of the new matrices $\mathbf{O}^{\leq}$ and $\mathbf{O}^{>}$. This strategy was first presented in Mehta et al. (1996) in the context of classification algorithms and has be rediscovered a number of times since then. The pre-sorting and

sifting **O** strategy is easy to implement for continuous covariates, but not for categorical covariates due to the possibility of ties in the data. Appendix A describes a special data structure for dealing with ties efficiently.

2.5.2. *Adaptive cutpoint grid.* Evaluating the integrated likelihood criterion is straightforward, but the summation and normalization required to sample the cutpoints contributes a substantial computational burden itself. Therefore, it is helpful to consider a restricted number of cutpoints C. This can be achieved simply by taking every jth value (starting from the smallest) as an eligible cutpoint with $j = \lfloor \frac{n_b - 2}{C} \rfloor$. As the tree grows deeper, the amount of data that is skipped over diminishes. Eventually, we get $n_b < C$, and each data point defines a unique cutpoint. In this way, the data could, without regularization, be fit perfectly, even though the number of cutpoints at any given level is given an upper limit. As a default, we set the number of cutpoints to $\min(n, 100)$, where n is the sample size of the entire data set.

Our cutpoint subsampling strategy is more straightforward than the elaborate cutpoint subselection search heuristics used by `XGBoost` (Chen and Guestrin, 2016) and `LightGBM` (Ke et al., 2017), which both consider the gradient evaluated at each cutpoint when determining the next split. Our approach does not consider the response information at all, but rather defines a predictor-dependent prior on the response surface. That is, given a design matrix $\mathbf{X}$, sample functions can be drawn from the prior distribution by sampling trees, splitting uniformly at random among the cutpoints defined by the node-specific quantiles, in a sequential fashion.

2.5.3. *Variable importance weights.* The variable weight parameter w is given a Dirichlet prior with hyper-parameters $\bar{\mathrm{w}}$ that is initialized to all ones. At each iteration of the first sweep through the forest, $\bar{\mathrm{w}}$ is incremented to count the total number of splits across all trees. The split counts are then updated in between each tree sampling/growth step:

$$\bar{\mathrm{w}} \leftarrow \bar{\mathrm{w}} - \bar{\mathrm{w}}_l^{(k-1)} + \bar{\mathrm{w}}_l^{(k)}$$

where $\bar{\mathrm{w}}_l^{(k)}$ denotes the length-V vector recording the number of splits on each variable in tree l at iteration k. The weight parameter is then re-sampled as $\mathrm{w} \sim \mathrm{Dirichlet}(\bar{\mathrm{w}})$. Splits that improve the likelihood function will be chosen more often than those that don't. The parameter w is then updated to reflect that, making chosen variables more likely to be considered in subsequent sweeps. In practice, we find it is helpful to use all V variables during an initialization phase, to more rapidly obtain an accurate initial estimate of w.

3. Tree-based models for nonlinear regression. This section provides details of XBART in the Gaussian nonlinear regression setting. We derive specific split criteria and sampling strategies for leaf parameters μ and non-tree parameters Ψ. We begin by considering a nonlinear mean regression additive error model

$$y = f(\mathbf{x}) + \epsilon, \tag{4}$$

where f is the unknown mean regression function $f(\mathbf{x}) = \mathbb{E}[y \mid \mathbf{x}]$ and $\epsilon \sim N(0, \sigma^2)$. The extra non-tree parameter is residual variance σ^2, which is given a standard inverse-Gamma(a, b) prior and updated in between each tree update. Reviewing notation, $\mathbf{x} = (\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(p)})$ is a p dimensional covariate vector and $y \in \mathbb{R}$ is the real response variable. Capital letters represent a vector or matrix of data, $Y = (y_1, \cdots, y_n)$ is a vector of n observations and $\mathbf{X} = (\mathbf{x}_1', \cdots, \mathbf{x}_n)'$ is a $n \times p$ matrix of covariate data. Leaf parameters are given independent and identical Gaussian priors, $\mu \sim N(0, \tau)$. In the notation from above, these modeling choices correspond to hyper-parameter and model parameters $\Phi = (a, b, \tau)$ and $\Psi = (\sigma)$, respectively.

Assuming that observations in the same leaf node share a common mean parameter, the prior predictive distribution — obtained by integrating out the unknown group-specific mean — is simply a mean-zero multivariate Gaussian distribution with covariance matrix $\mathbf{V}$,

$$p(Y \mid \tau, \sigma^2) = \int N(Y \mid \mu, \sigma^2 \mathbf{I}_n) \mathrm{N}(\mu \mid 0, \tau) d\mu = N(0, \mathbf{V}), \tag{5}$$

where $\mathrm{N}(Y \mid \mu, \sigma^2 \mathbf{I}_n)$ denotes the density of multivariate Gaussian distribution with mean μ and covariance matrix $\sigma^2 \mathbf{I}_n$, n is number of data observations in the current node. We have

$$\mathbf{V} = \tau \mathrm{J}\mathrm{J}^t + \sigma^2 \mathbf{I}_n, \quad \mathbf{V}^{-1} = \sigma^{-2}\mathbf{I} - \frac{\tau}{\sigma^2(\sigma^2 + \tau n)} \mathrm{J}\mathrm{J}^t,$$

where J is a length n column vector of all ones. Observe that the prior predictive density of $Y \sim \mathrm{N}(0, \mathbf{V})$ is

$$p(Y \mid \tau, \sigma^2) = (2\pi)^{-n/2} \det(\mathbf{V})^{-1/2} \exp\left(-\frac{1}{2} Y^t \mathbf{V}^{-1} Y\right),$$

which can be simplified by a direct application of the matrix inversion lemma to $\mathbf{V}^{-1}$. Applying Sylvester's determinant theorem to $\det \mathbf{V}^{-1}$ yields

$$\det \mathbf{V}^{-1} = \sigma^{-2n}\left(1 - \frac{\tau n}{\sigma^2 + \tau n}\right) = \sigma^{-2n}\left(\frac{\sigma^2}{\sigma^2 + \tau n}\right).$$

Taking logarithms yields a marginal log-likelihood of

$$-\frac{n}{2}\log(2\pi) - n\log(\sigma) + \frac{1}{2}\log\left(\frac{\sigma^2}{\sigma^2 + \tau n}\right) - \frac{1}{2}\frac{Y^t Y}{\sigma^2} + \frac{1}{2}\frac{\tau}{\sigma^2(\sigma^2 + \tau n)} s^2,$$

where we write the sufficient statistics $s \equiv Y^t \mathrm{J} = \sum_i y_i$ so that $Y' \mathrm{J}\mathrm{J}^t Y = (\sum_i y_i)^2 = s^2$. This likelihood is applied separately to two child nodes of a single cutpoint (j,k). Because observations in different leaf nodes are independent (conditional on σ^2), the full marginal log-likelihood is given by

$$\begin{aligned}
&\sum_{b=1}^{2} \left\{ -\frac{n_b}{2} \log(2\pi) - n_b \log(\sigma) + \frac{1}{2} \log\left(\frac{\sigma^2}{\sigma^2 + \tau n_b}\right) - \frac{1}{2} \frac{Y_b^t Y_b}{\sigma^2} + \frac{1}{2} \frac{\tau}{\sigma^2(\sigma^2 + \tau n_b)} s_b^2 \right\} \\
&= -\frac{n}{2} \log(2\pi) - n \log(\sigma) - \frac{1}{2} \frac{Y^t Y}{\sigma^2} + \frac{1}{2} \sum_{b=1}^{2} \left\{ \log\left(\frac{\sigma^2}{\sigma^2 + \tau n_b}\right) + \frac{\tau}{\sigma^2(\sigma^2 + \tau n_b)} s_b^2 \right\},
\end{aligned}$$

where index b runs over two child nodes and $\sum_{b=1}^{2} n_b = n$. Notice that the first three terms are not functions of the partition (the tree parameter), and so may be ignored, leaving

$$\frac{1}{2} \sum_{b=1}^{2} \left\{ \log\left(\frac{\sigma^2}{\sigma^2 + \tau n_b}\right) + \frac{\tau}{\sigma^2(\sigma^2 + \tau n_b)} s_b^2 \right\}$$

as the model-based split criterion, where (n_b, s_b) are functions of the data. Therefore, we define the log-integrated-likelihood

$$\log(m(s)) = \log\left(\frac{\sigma^2}{\sigma^2 + \tau n}\right) + \frac{\tau}{\sigma^2(\sigma^2 + \tau n)} s^2. \tag{6}$$

The logarithm of split criterion (j,k) is

$$\begin{aligned}
\log\left(m(s_{jk}^l) m(s_{jk}^r)\right) &= \log\left(\frac{\sigma^2}{\sigma^2 + \tau n_{jk}^l}\right) + \frac{\tau}{\sigma^2(\sigma^2 + \tau n_{jk}^l)} \left(s_{jk}^l\right)^2 \\
&+ \log\left(\frac{\sigma^2}{\sigma^2 + \tau n_{jk}^r}\right) + \frac{\tau}{\sigma^2(\sigma^2 + \tau n_{jk}^r)} \left(s_{jk}^r\right)^2,
\end{aligned} \tag{7}$$

where $n_{jk}^l = |\mathcal{A}_L(j,k)|$ and $n_{jk}^r = |\mathcal{A}_R(j,k)|$ are number of data observations on left or right child node if split at cutpoint (j,k). s_{jk}^l and s_{jk}^r are sufficient statistics

$$s_{jk}^l = \sum_{i : x_i \in \mathcal{A}_L(j,k)} y_i, \quad s_{jk}^r = \sum_{i : x_i \in \mathcal{A}_R(j,k)} y_i.$$

Similarly, the log-probability of no-split is

$$\log\left(|\mathcal{C}| \left(\frac{(1+d)^\beta}{\alpha} - 1\right)\right) + \log\left(\frac{\sigma^2}{\sigma^2 + \tau n}\right) + \frac{\tau}{\sigma^2(\sigma^2 + \tau n)} s^2. \tag{8}$$

For notational simplicity, we overload n as the number of data observations in the *current* node and s is sum of all y in the *current* node. It is apparent that $n = n_{jk}^l + n_{jk}^r$ and $s = s_{jk}^l + s_{jk}^r$ for all cutpoints (j,k). Note that the split criterion involves residual standard error σ, meaning that it is adaptively regularizing within the model fitting process.

If the no-split option is selected or stopping conditions are satsified, we label this node as leaf $\mathcal{A}_{lb}$, the b-th leaf of l-th tree. Leaf parameter μ_{lb} associated with leaf $\mathcal{A}_{lb}$ is updated in step 10 of Algorithm 2. We assume a conjugate Gaussian prior $\mu_{lb} \sim N(0, \tau)$, therefore the posterior to sample from is

$$\mu_{lb} \sim N\left(\frac{s_{lb}}{\sigma^2\left(\frac{1}{\tau} + \frac{n_{lb}}{\sigma^2}\right)}, \frac{1}{\frac{1}{\tau} + \frac{n_{lb}}{\sigma^2}}\right), \tag{9}$$

where n_{lb} is number of data observations and $s_{lb} = \sum_{y \in \mathcal{A}_{lb}} y$ is the sufficient statistic in the leaf node corresponding to leaf parameter μ_{lb}.

Next, we describe the model parameter sampling steps in Algorithm 3. The only non-tree model parameter for Gaussian nonlinear regression is the residual variance σ^2, which updates after one draw of a tree in step 10 of Algorithm 3. For σ^2 we assume a standard inverse-Gamma prior, $\sigma^2 \sim$ inverse-Gamma(a, b), and the posterior is

$$\sigma^2 \sim \text{inverse-Gamma}\left(N + a, \tilde{\mathrm{r}}_h^{(iter)t}\tilde{\mathrm{r}}_h^{(iter)} + b\right), \tag{10}$$

where $\tilde{\mathrm{r}}_h^{(iter)}$ is the *total* residual after updating the h-th tree in the *iter*-th Monte Carlo iteration, defined as

$$\tilde{\mathrm{r}}_h^{(iter)} \equiv \mathrm{y} - \sum_{h' \leq h} g(\mathbf{X}; T_{h'}, \mu_{h'})^{(iter+1)} - \sum_{h' > h} g(\mathbf{X}; T_{h'}, \mu_{h'})^{(iter)}.$$

Remark The derivations above pertain to growing a single tree by Algorithm 2. Note that in the context of the forest, the data y in the above would instead be the residual r_h.

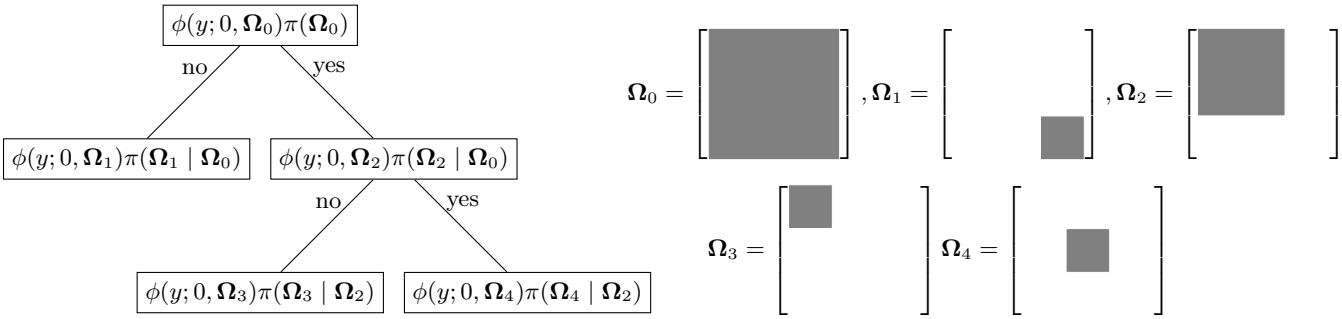
The default parameters, used in all simulations reported here, are $L = 30$ trees and $\tau = \mathrm{Var}(\mathrm{y})/L$.

3.1. *Model implied by the GrowFromRoot algorithm.* The XBART cutpoint sampling, while based on Bayes rule, is *myopic* in the sense that it does not consider the entire tree structure when evaluating its (marginal) likelihood. In particular, the recursive structure of a binary tree implies that the data is "reused" at different levels of the tree.

However, interestingly, in the case of a single tree it is possible to show that GrowFromRoot can be interpreted as a proper Bayesian model, as follows. From equation (5), the integrated likelihood of a single leaf is Gaussian with mean a vector of zeros and precision matrix

$$\mathbf{\Omega} := \mathbf{V}^{-1} = \sigma^{-2}\mathbf{I} - \frac{\tau}{\sigma^2(\sigma^2 + \tau n)}\mathrm{J}\mathrm{J}^t,$$

where $\mathbf{I}$ is a $n \times n$ identity matrix and J is a vector of ones with length n. Now, regard the tree growing algorithm as an exhaustive one, always growing maximum depth trees (relative to $\mathbf{X}$):



(a) Nodes and corresponding precision matrices.

(b) Precision matrices

Fig 2: An illustration of the precision matrices at each node, from root to leaves. Left panel: assignment of precision matrix at each node. Right panel: illustration of precision matrices, where grey block represents non-zero elements and white blocks are 0.

while the tree keeps splitting, the integrated likelihood is a Gaussian likelihood; once a node stops splitting, the likelihood of all the nodes beneath it degenerate to 1.

The posterior of a single tree model is

$$\pi_{gfr}(T \mid \mathrm{y}) \propto \prod_{i=0}^{B} \phi(y;0,\mathbf{\Omega}_i)P(i), \tag{11}$$

where B is number of all nodes in the tree, and $\phi(y;0,\mathbf{\Omega})$ is a multivariate Gaussian PDF with precision matrix $\mathbf{\Omega}$ and $P(i) = \alpha(1+d_i)^{-\beta}$ is the BART prior probability of the i-th node reaching depth d_i. Figure 2 illustrates the assignment and structure of precision matrices. All precision matrices have the same dimension n, the total number of observations. Since each non-root node only has a subset of the data, the precision matrix has a block-diagonal structure with a non-zero sub-matrix on the diagonal indicating correlation of data observations in that node and 0 elsewhere (it is always possible to rearrange order of the data to make the precision matrix block-diagonal). The cumulative product of Gaussian kernels in equation (11) represents another Gaussian kernel up to a normalizing constant

$$\prod_{i=1}^{B} \phi(y;0,\mathbf{\Omega}_i)P(i) = \exp\left(\xi_{i=1,\cdots,B} - \xi_B\right) \exp\left(\xi_B - \frac{1}{2} y^t \mathbf{\Omega}^B y\right) \prod_{i=1}^{B} P(i), \tag{12}$$

where $\mathbf{\Omega}^B = \sum_{i=1}^{B} \mathbf{\Omega}_i$, $\xi_B = -\frac{1}{2}\left(N\log(2\pi) - \log|\mathbf{\Omega}^B|\right)$ and $\xi_{i=1,\cdots,B} = \sum_{i=1}^{B} -\frac{1}{2}\left(N\log(2\pi) - \log|\mathbf{\Omega}_i|\right)$. Therefore, we may consider the GrowFromRoot likelihood to be the single multivariate Gaussian

$$\phi(y;0,\mathbf{\Omega}^B) = \exp\left(\xi_B - \frac{1}{2} y^t \mathbf{\Omega}^B y\right), \tag{13}$$

in which the data only appear once, and the implied prior of GrowFromRoot (which does not include y), is

$$\exp\left(\xi_{i=1,\cdots,B}\right)\prod_{i=1}^{B}P(i). \tag{14}$$

On the other hand, the BART posterior is

$$\pi_{bart}(T \mid \mathrm{y}) \propto \prod_{i\in\text{Leaf}}\phi(y;0,\mathbf{\Omega}_i)\prod_{i=1}^{B}P(i), \tag{15}$$

Here the likelihood is $\prod_{i\in\text{Leaf}}\phi(y;0,\mathbf{\Omega}_i)$ and the prior is $\prod_{i=1}^{B}P(i)$.

Remark The discussion above is only to show that the GrowFromRoot sampling process can be considered a well defined model, but it is *not* the one that is used to sample leaf parameters. Indeed, the multivariate Gaussian model above corresponds to building up the mean function from a weighted average of node-specific mean vectors. We attempted estimating the mean parameters in this fashion and it was dramatically outperformed by using only the leaf parameters, as in BART. Nonetheless, the GrowFromRoot algorithm appears to produce samples of trees that perform well in conjunction with the leaf-only estimation method of the conditional means.

4. Theory of Consistency. Tree-based methods have a substantial, if incomplete, body of theory going back several decades. Gordon and Olshen (1980) analyze the consistency of recursive partitioning irrespective of the specific split criterion; to achieve this they assume that the diameter of leaf node hyperrectangles shrink to zero at a certain rate. Breiman (2001) gives an upper bound of the generalization error of random forest, and Lin and Jeon (2006) show a lower bound of the generalization error of a nonadaptive forest. Biau et al. (2008) and Ishwaran et al. (2008) establish consistency of a simplified random forest model. Scornet et al. (2015) is the first consistency result of the original random forest algorithm, and their theory applies to the consistency of CART directly. Wager and Athey (2018) study the asymptotic sampling distribution of random forest. Zhang et al. (2005) prove consistency and derive the convergence rate for boosting with early stopping, although the result is non-constructive in that their results are not known to apply to any specific stopping-rule. Bartlett and Traskin (2007) establish consistency theory for the adaBoost algorithm.

There has also been a surge of recent theoretical results for BART. Coram et al. (2006) prove consistency for Bayesian histograms of binary regression. Rocková and van der Pas (2017) prove posterior consistency for a variant of the BART prior and Ročková and Saha (2019) study posterior

concentration of the exact BART prior. Linero and Yang (2018) establish posterior consistency for a fractional posterior of soft BART (SBART), whose trees have soft decision rules.

In this section, we prove the consistency of a single tree of the XBART algorithm for the Gaussian nonlinear regression case. First, we establish the connection of our XBART sampling strategy to the optimization approach in CART by applying the perturb-max theorem. Having reconciled the sampling versus optimizing distinction, we are then able to adapt the consistency proof for CART to the XBART split criterion. The key step of the proof is to show that variation of the true function is small in each hyper-rectangular cells associated to a leaf node, as the number of data observations grows large enough, either because the diameter of the cell shrinks to zero or because the true function is flat over that region. We follow the proof of Scornet et al. (2015) closely.

Before diving into the theorem and proofs, we again review notations. Suppose $\mathbf{x} \in [0,1]^p$ is a vector of input variables and $y \in \mathbb{R}^1$ is the corresponding outcome variable. Our goal is to estimate the regression function $f(\mathbf{x}) = \mathbb{E}[y \mid \mathbf{x}]$ as $f_n : (0,1)^p \to \mathbb{R}$ based on data $\{(y_1, \mathbf{x}_1), \cdots, (y_n, \mathbf{x}_n)\}$. Let d_n denote maximum depth of a tree.

A key assumption of Scornet et al. (2015) is that the regression function is additive,

Assumption 1 (A1).

$$y = \sum_{j=1}^{p} f_j(x^{(j)}) + \epsilon$$

where $\mathbf{x} = (\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(p)})$ *is uniformly distributed on* $[0,1]^p$. $\epsilon \sim N(0, \sigma^2)$.

See the remark following Lemma 2 concerning the possibility of relaxing this strong assumption.

We show consistency for the case of regression with Gaussian noise and focus on a variant of XBART algorithm which only contains a single tree. Our main theorem states that a single XBART regression tree approximates the true underlying mean function in $\mathcal{L}^2$ norm if maximum depth goes to infinity slower than a function of the number of data points.

Theorem 3. *Assume (A1) holds. Let* $n \to \infty$, $d_n \to \infty$ *and* $(2^{d_n} - 1)(\log n)^9/n \to 0$, *XBART is consistent in the sense that*

$$\lim_{n \to \infty} \mathbb{E}[f_n(\mathbf{x}) - f(\mathbf{x})]^2 = 0. \tag{16}$$

Both a single-tree XBART and CART learn decision rules by a recursive algorithm, but with a different way of selecting cutpoints. CART optimizes its split criterion while XBART draws

cutpoints randomly with probability proportional to the split criterion. However, sampling from a so-called perturb-max model is equivalent to optimizing an objective function with an additional random draw from Gumbel$(0,1)$ distribution, see Corollary 6.2 from Hazan et al. (2016), restated here for the sake of completeness.

LEMMA 1 (Perturb-max theorem). *Suppose there are $|\mathcal{C}|$ finite cutpoint candidates $\{c_{jk}\}$ at a specific node. We are interested in drawing one of them according to probability $P(c_{jk}) = \frac{\exp(l(c_{jk}))}{\sum_{c_{jk}\in\mathcal{C}}\exp(l(c_{jk}))}$. We have*

$$\frac{\exp(l(c_{jk}))}{\sum_{c_{jk}\in\mathcal{C}}\exp(l(c_{jk}))} = P\left(c_{jk} = \arg\max_{c_{jk}\in\mathcal{C}}\{l(c_{jk}) + \gamma_{jk}\}\right) \tag{17}$$

where $\{\gamma_{jk}\}$ are independent random draws from a Gumbel$(0,1)$ distribution with density $p(x) = \exp(-x + exp(-x))$.

The independent random draws $\{\gamma_{jk}\}$ can be treated as known constants if conditioning on a random seed Θ, as in Scornet et al. (2015). That is, Θ is used to sample Gumbel random draws, and we always assume taking the condition of Θ in the following proof. Lemma 1 states that XBART's sampling cutpoint strategy is equivalent to optimizing an objective function. Thus CART and XBART fitting algorithms only differ in the specific form of the split criterion to optimize. Our proof of consistency is based on the work of Scornet et al. (2015), where only Lemma 1 and Lemma 2 involve the specific function form of split criterion. Therefore we only have to check that Lemma 1 and Lemma 2 of Scornet et al. (2015) are still valid for the XBART split criterion. Recalling equation (7), the logarithm of the split criterion c_{jk} is

$$\begin{aligned}
l(c_{jk}) = &\log\left(\frac{\sigma^2}{\sigma^2+\tau n^l_{jk}}\right) + \frac{\tau}{\sigma^2\left(\sigma^2+\tau n^l_{jk}\right)}\left(\sum_{i:\mathbf{x}_i\in\mathcal{A}_L(j,k)} y_i\right)^2 \\
&+ \log\left(\frac{\sigma^2}{\sigma^2+\tau n^r_{jk}}\right) + \frac{\tau}{\sigma^2\left(\sigma^2+\tau n^r_{jk}\right)}\left(\sum_{i:\mathbf{x}_i\in\mathcal{A}_R(j,k)} y_i\right)^2 \\
= &\frac{\tau}{\sigma^2\left(\sigma^2+\tau n^l_{jk}\right)}\left(n^l_{jk}\sum_{i:\mathbf{x}_i\in\mathcal{A}_L(j,k)} y_i^2 - (n^l_{jk}-1)\sum_{i:\mathbf{x}_i\in\mathcal{A}_L(j,k)}(y_i-\bar{y}_l)^2\right) \\
&+ \frac{\tau}{\sigma^2\left(\sigma^2+\tau n^r_{jk}\right)}\left(n^r_{jk}\sum_{i:\mathbf{x}_i\in\mathcal{A}_R(j,k)} y_i^2 - (n^r_{jk}-1)\sum_{i:\mathbf{x}_i\in\mathcal{A}_R(j,k)}(y_i-\bar{y}_r)^2\right) \\
&+ \log\left(\frac{\sigma^2}{\sigma^2+\tau n^l_{jk}}\right) + \log\left(\frac{\sigma^2}{\sigma^2+\tau n^r_{jk}}\right),
\end{aligned}$$

where $\bar{y}_l = \frac{1}{n^l_{jk}} \sum_{i:\mathbf{x}_i \in \mathcal{A}_L(j,k)} y_i$ and $\bar{y}_r = \frac{1}{n^r_{jk}} \sum_{i:\mathbf{x}_i \in \mathcal{A}_R(j,k)} y_i$ are averagea of y in the left and right children respectively. Following Lemma 1, we optimize

$$c^*_{jk} = \arg\max_{c_{jk} \in \mathcal{C}} \{l(c_{jk}) + \gamma_{jk}\},$$

where γ_i are random draws from Gumbel$(0, 1)$ and can be treated as fixed constant if we condition on random seed Θ. Note that the optimization problem is invariant if the objective function is scaled by a constant n, used here to denote the number of observations in the current node, so that

$$\arg\max_{c_{jk} \in \mathcal{C}} \frac{l(c_{jk})}{n} + \frac{\gamma_{jk}}{n}.$$

and our "empirical" split criterion (in the terminology of Scornet et al. (2015)) is defined as

$$L_n(c_{jk}) = \frac{l(c_{jk})}{n} + \frac{\gamma_x}{n}. \tag{18}$$

Letting $n \to \infty$, our empirical split criterion function $L_n(x)$ converges to the "theoretical" version

$$L^*(j, c_{jk}) = \frac{1}{\sigma^2} P(\mathbf{x}^{(j)} \leq c_{jk}) \left[\mathbb{E}(y \mid \mathbf{x}^{(j)} \leq c_{jk})\right]^2 + \frac{1}{\sigma^2} P(\mathbf{x}^{(j)} > c_{jk}) \left[\mathbb{E}(y \mid \mathbf{x}^{(j)} > c_{jk})\right]^2. \tag{19}$$

Importantly, $L^*(j, c_{jk})$ does not rely on the training data because, by the strong law of large numbers, $L_n(c_{jk}) \to L^*(c_{jk})$ almost surely as $n \to \infty$. Again following Scornet et al. (2015), we refer to a tree grown according to the empirical split criterion $L_n(c_{jk})$ or the theoretical criterion $L^*(c_{jk})$ as an empirical tree or theoretical tree, respectively. It worth emphasizing that the theoretical split criterion of XBART and CART are the same up to a multiplicative constant $1/\sigma^2$.

In the rest of the section, we recap the proof of consistency for CART and random forest by Scornet et al. (2015) and verify that all lemmas involving the CART split criterion are also valid for that of XBART, equation (18).

More notation is needed for the proof. Write $c = (c^{(1)}, c^{(2)})$ to represent a cutpoint, where $c^{(1)} \in \{1, \cdots, p\}$ indicates cut variables and $c^{(2)} \in [0, 1]$ indicates cut values. Let $\mathcal{A}_n(\mathbf{x}, \Theta)$ denote the leaf node of an *empirical tree* built with random parameter Θ that contains $\mathbf{x}$. Let $\mathcal{A}^*_k(\mathbf{x}, \Theta)$ be a cell of the *theoretical tree* at depth k containing $\mathbf{x}$. Additionally, $\mathcal{A}(\mathbf{x}, \mathbf{c}_k)$ is the node containing $\mathbf{x}$ built with sequence of cuts $\mathbf{c}_k$. This node is reached via a sequence of cuts $\mathbf{c}_k = (c_1, \cdots, c_k)$ and we call $\mathbb{A}_k(\mathbf{x})$ the set of all possible $k \geq 1$ cuts used to create the node containing $\mathbf{x}$. The distance between two cut sequences $\mathbf{c}_k, \mathbf{c}'_k \in \mathbb{A}_k(x)$ is defined as

$$||\mathbf{c}_k - \mathbf{c}'_k||_\infty = \sup_{1 \leq j \leq k} \max\left(\left|c^{(1)}_j - c'^{(1)}_j\right|, \left|c^{(2)}_j - c'^{(2)}_j\right|\right).$$

The distance between a cut $\mathbf{c}_k$ and a set $\mathbb{A} \subset \mathbb{A}_k(\mathbf{x})$ is

$$c_\infty(\mathbf{c}_k, \mathbb{A}) = \inf_{\mathbf{c} \in \mathcal{A}} ||\mathbf{c}_k - \mathbf{c}||_\infty.$$

We define the total variation of the true function f within any leaf node $\mathcal{A}$ as

$$\Delta(f, \mathcal{A}) = \sup_{\mathbf{x}, \mathbf{x}' \in \mathcal{A}} |f(\mathbf{x}) - f(\mathbf{x}')|.$$

The proof of Theorem 3 relies critically on the following proposition:

PROPOSITION 1. *Assume (A1) holds. For all $\rho > 0$ and $\xi > 0$, there exists an $N \in \mathbb{N}^*$ such that, for all $n > N$,*

$$\mathbb{P}\left[\Delta(f, \mathcal{A}_n(\mathbf{x}, \Theta)) \leq \xi\right] \geq 1 - \rho. \tag{20}$$

Proposition 1 states that the total variation of the true function f within any leaf node of the empirical tree is small if the number of observations, n, used to fit the tree is large enough. In general, the consistency result controling the behavior of the true function on each of the partitions defined by the leaf nodes, in that either the cell diameter shrinks to zero or else the true function is constant over any non-vanishing cell. The proof of Proposition 1 is based on three lemmas below. Lemma 2 and Lemma 3 are the only two pieces involving the specific functional form of the split criterion in the complete proof of Theorem 3.

LEMMA 2. *Assume that (A1) holds. Then for all $x \in (0,1)^p$,*

$$\Delta(f, \mathcal{A}_k^*(\mathbf{x}, \Theta)) \to 0 \qquad \textit{almost surely as } k \to \infty.$$

Lemma 2 shows that as $n \to \infty$ and tree grows deeper, variation of the true function f tends to zero in the leaf node of a *theoretical* tree.

Remark Assumption (A1) is used in the proof of Lemma 2 only. If the true function f is additive, Lemma 2 is valid. However, a weaker replacement of assumption (A1) is to assume Lemma 2 is valid directly. Although this is perhaps less interpretable than an assumption of an additive model, it is also presumably a weaker assumption in that it may be satisfied by non-additive models.

Next, we show that the cuts of an empirical tree will be close to its associated theoretical tree in a certain sense. Suppose the empirical tree has grown following a sequence of cuts $\mathbf{c}_{k-1}$, and consider splitting node $\mathcal{A}(\mathbf{x}, \mathbf{c}_{k-1})$. Let $\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1}) = \mathcal{A}(\mathbf{x}, \mathbf{c}_{k-1}) \cap \{\mathbf{x} : \mathbf{x}^{(c_k^{(1)})} \leq c_k^{(2)}\}$ and

$\mathcal{A}_R(\mathbf{x}, \mathbf{c}_{k-1}) = \mathcal{A}(\mathbf{x}, \mathbf{c}_{k-1}) \cap \{\mathbf{x} : \mathbf{x}^{(c_k^{(1)})} > c_k^{(2)}\}$ be left and right child nodes of node $\mathcal{A}(\mathbf{x}, \mathbf{c}_{k-1})$ given cut c_k. We write the split criterion equation (18) explicitly for $\mathcal{A}(\mathbf{x}, \mathbf{c}_{k-1})$,

$$
\begin{aligned}
L_{n,k}(\mathbf{x}, \mathbf{c}_k) = \frac{1}{n} \frac{\tau}{\sigma^2 \left(\sigma^2 + \tau \mathrm{N}_n(\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1}))\right)} \Bigg(& \mathrm{N}_n(\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1})) \sum_{i:\mathbf{x}_i \in \mathrm{N}_n(\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1}))} y_i^2 \\
& - (\mathrm{N}_n(\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1})) - 1) \sum_{i:\mathbf{x}_i \in \mathrm{N}_n(\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1}))} (y_i - \bar{y}_{\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1})})^2 \Bigg) \\
+ \frac{1}{n} \frac{\tau}{\sigma^2 \left(\sigma^2 + \tau \mathrm{N}_n(\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1}))\right)} \Bigg(& \mathrm{N}_n(\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1})) \sum_{i:\mathbf{x}_i \in \mathcal{A}_R(\mathbf{x}, \mathbf{c}_{k-1})} y_i^2 \\
& - (\mathrm{N}_n(\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1})) - 1) \sum_{i:\mathbf{x}_i \in \mathcal{A}_R(\mathbf{x}, \mathbf{c}_{k-1})} (y_i - \bar{y}_r)^2 \Bigg) \\
+ \frac{1}{n} \log \left(\frac{\sigma^2}{\sigma^2 + \tau \mathrm{N}_n(\mathcal{A}_L(\mathbf{x}, \mathbf{c}_{k-1}))} \right) & + \frac{1}{n} \log \left(\frac{\sigma^2}{\sigma^2 + \tau \mathrm{N}_n(\mathcal{A}_R(\mathbf{x}, \mathbf{c}_{k-1}))} \right).
\end{aligned}
$$

Lemma 3 below states that $L_{n,k}(\mathbf{x}, \mathbf{c}_k)$ is "stochastically equicontinuous" on $\mathbf{c}_k$ for all $\mathbf{x} \in [0,1]^p$. For all $\xi > 0$ and $\mathbf{x} \in [0,1]^p$, $\mathbb{A}_{k-1}^{\xi}(\mathbf{x}) \subset \mathbb{A}_{k-1}(\mathbf{x})$ denotes the set of all sequences of cuts $\mathbf{c}_{k-1}$ such that the node $\mathcal{A}(\mathbf{x}, \mathbf{c}_{k-1})$ contains a hypercube with edge length ξ. The set $\bar{\mathbb{A}}_k^{\xi}(x) = \{\mathbf{c}_k : \mathbf{c}_{k-1} \in \mathbb{A}_{k-1}^{\xi}(x)\}$ is equipped with norm $||\cdot||_\infty$.

LEMMA 3. *Assume that (A1) holds. Fix $x \in [0,1]^p$, $k \in \mathbb{N}^*$ and let $\xi > 0$. Then $L_{n,k}(x, \dot{)}$ is stochastically equicontinuous on $\bar{\mathbb{A}}_k^{\xi}(x)$, that is, for all α, $\rho > 0$, there exist $\delta > 0$ such that*

$$
\lim_{n \to \infty} \mathbb{P} \left[\sup_{\substack{||\mathbf{c}_k - \mathbf{c}'_k||_\infty \leq \delta \\ \mathbf{c}_k, \mathbf{c}'_k \in \bar{\mathbb{A}}_k^{\xi}(\mathbf{x})}} \left| L_{n,k}(\mathbf{x}, \mathbf{c}_k) - L_{n,k}(\mathbf{x}, \mathbf{c}'_k) \right| > \alpha \right] \leq \rho.
$$

Lemma 3 is used in the proof of Lemma 4 below.

LEMMA 4. *Assume that (A1) holds. Fix $\xi > 0$, $\rho > 0$, and $k \in \mathbb{N}^*$. Then there exists $N \in \mathbb{N}^*$ such that for all $n \geq N$,*

$$
\mathbb{P} \left[c_\infty(\widehat{\mathbf{c}}_{k,n}(\mathbf{x}, \Theta), \mathcal{A}_k^*(\mathbf{x}, \Theta)) \leq \xi \right] \geq 1 - \rho. \tag{21}
$$

Lemma 4 states that the empirical tree converges to the theoretical tree in probability. The proof of Proposition 1 is the same as in Scornet et al. (2015) and so is omitted here. Only proofs of Lemma 2 and 3 rely on the specific form of split criterion; complete proofs are presented in the Appendix.

Finally, we are equipped to prove Theorem 3. Proposition 1 offers good control of approximation error if the tree is grown by the XBART split criterion. There are two steps for the proof. First, the result is proved for the case of a truncated estimator, which is based on Theorem 10.2 of Györfi et al. (2006), presented as Theorem 4 below. Then, the truncation is released to prove the untruncated case.

The truncation T_{β_n} is defined as

$$\begin{cases} T_{\beta_n}(u) = u & \text{if} \quad |u| \leq \beta_n \\ T_{\beta_n}(u) = \text{sign}(u)\beta_n & \text{if} \quad |u| > \beta_n \end{cases}$$

where $\{\beta_n\}$ is a sequence of positive real numbers. The partition obtained with random variable Θ and data set $\mathcal{D}_n$ is denoted by $\mathcal{P}_n$. Let $\mathcal{M}_n(\Theta)$ is the set of all functions $m : [0,1]^p \to \mathbb{R}$ which is piecewise constant on each node of the partition $\mathcal{P}_n(\Theta)$.

THEOREM 4 (Györfi et al. (2006)). *Assume that*

1. $\lim_{n\to\infty} \beta_n = \infty$;
2. $\lim_{n\to\infty} \mathbb{E}\left[\inf_{\substack{m\in\mathcal{M}_n(\Theta) \\ ||m||_\infty \leq \beta_n}} \mathbb{E}_X \left[m(\mathbf{x}) - f(\mathbf{x})\right]^2\right] = 0$;
3. *for all truncations at* $L > 0$,

$$\lim_{n\to\infty} \mathbb{E}\left[\sup_{\substack{m\in\mathcal{M}_n(\Theta) \\ ||f||_\infty \leq \beta_n}} \left| \frac{1}{a_n} \sum_{i=1}^{n} \left[m(\mathbf{x}_i) - T_L\left(y_i\right)\right]^2 - \mathbb{E}\left[m(\mathbf{x}) - T_L(y)\right]^2 \right|\right] = 0.$$

Then

$$\lim_{n\to\infty} \mathbb{E}[T_{\beta_n}\left(f_n(X,\Theta)\right) - f(X)]^2 = 0.$$

It is sufficient to verify the three assumptions of Theorem 4 to show that the truncated estimator is consistent. Intuitively, the first condition says that the truncation is relaxed as $n \to \infty$, and the next two conditions control the approximation error and the estimation error, respectively. For the sake of brevity, we skip the proof. Interested readers may refer to Scornet et al. (2015) for details.

5. Simulation Studies.

5.1. *Time-accuracy comparisons to other popular machine learning methods.*

5.1.1. *Synthetic regression data.* To demonstrate the performance of XBART, we estimate function evaluations with a hold-out set that is a quarter of the training sample size and judge accuracy according to root mean squared (estimation) error (RMSE). We consider four different challenging functions, f, as defined in Table 1. In all cases, $x_j \overset{\text{iid}}{\sim} N(0,1)$ for $j = 1, \ldots, d = 30$. The data is generated according to the additive error mode, with $\epsilon_i \overset{\text{iid}}{\sim} N(0,1)$. We consider $\sigma = \kappa \text{Var}(f)$ for $\kappa \in \{1, 10\}$.

TABLE 1
Four true f functions

Name	Function
Linear	$\mathbf{x}^t\gamma;\ \gamma_j = -2 + \frac{4(j-1)}{d-1}$
Single index	$10\sqrt{a} + \sin(5a);\ a = \sum_{j=1}^{10}(x_j - \gamma_j)^2;\ \gamma_j = -1.5 + \frac{j-1}{3}.$
Trig + poly	$5\sin(3x_1) + 2x_2^2 + 3x_3x_4$
Max	$\max(x_1, x_2, x_3)$

We compare to leading machine learning algorithms: random forests, gradient boosting machines, neural networks, and BART. All implementations had an `R` interface and were the current fastest implementations to our knowledge: `ranger` (Wright and Ziegler, 2015), `xgboost` (Chen and Guestrin, 2016), and `Keras` (Chollet et al., 2015), `dbarts` respectively. For `Keras` we used a single architecture but varied the number of training epochs depending on the noise level of the problem. For `xgboost` we consider two specifications, one using the software defaults and another determined by a 5-fold cross-validated grid optimization (see Table 2); a reduced grid of parameter values was used at sample sizes $n > 10,000$. Comparison with `ranger` and `dbarts` are shown in supplementary material.

TABLE 2
Hyperparameter Grid for XGBoost

Parameter name	$N = 10$K	$N > 10$K
`eta`	$\{0.1, 0.3\}$	$\{0.1, 0.3\}$
`max_depth`	$\{4, 8, 12\}$	$\{4, 12\}$
`colsample_bytree`	$\{0.7, 1\}$	$\{0.7, 1\}$
`min_child_weight`	$\{1, 10, 15\}$	10
`subsample`	0.8	0.8
`gamma`	0.1	0.1

The software used is `R` version 3.4.4 with XGBoost 0.71.2, `dbarts` version 0.9.1, `ranger` 0.10.1 and `keras` 2.2.0. The default hyperparameters for XGBoost are `eta` = 0.3, `colsample_bytree` = 1, `min_child_weight` = 1 and `max_depth` = 6. Ranger was fit with `num.trees` = 500 and `mtry` = $5 \approx \sqrt{d}$. BART, with the package `dbarts`, was fit with the defaults of `ntrees` = 200, `alpha`

= 0.95, `beta` = 2, with a burn-in of 5,000 samples (`nskip` = 5000) and 2,000 retrained posterior samples (`ndpost` = 2000).

The default `dbarts` algorithm uses an evenly spaced grid of 100 cutpoint candidates along the observed range of each variable (`numcuts` = 100, `usequants = FALSE`). For `Keras` we build a network with two fully connected hidden layers (15 nodes each) using ReLU activation function, ℓ_1 regularization at 0.01, and with 50/20 epochs depending on the signal to noise ratio.

5.1.2. *Results.* The performance of the new XBART algorithm was excellent, showing superior speed and performance relative to all the considered alternatives on virtually every data generating process. The full results, averaged across five Monte Carlo replications, are reported in Table 3. Neural networks perform as well as XBART in the low noise settings under the Max and Linear functions. Unsurprisingly, neural networks outperform XBART under the linear function with low noise. Across all data generating processes and sample sizes, XBART was 31% more accurate than the cross-validated XGBoost method and typically faster. Specifically, the supplement examines the empirical examples given in Chipman et al. (2010).

The XBART method was slower than the untuned default XGBoost method but was 350% more accurate. This pattern points to one of the main benefits of the proposed method, which is that it has excellent performance using the same hyperparameter settings across all data generating processes. Importantly, these default hyperparameter settings were decided on the basis of prior elicitation experiments using different true functions than were used in the reported simulations. While XGBoost is quite fast, the tuning processes are left to the user and can increase the total computational burden by orders of magnitude.

Random forests and BART were prohibitively slow at larger sample sizes. However, at $n = 10,000$ several notable patterns did emerge; see the supplementary material for full details. First was that BART and XBART typically gave very similar results, as would be expected. BART performed slightly better in the low noise setting and quite a bit worse in the high noise setting (likely due to inadequate burn-in period). Similarly, random forests do well in higher noise settings, while XGBoost and neural networks perform better in lower noise settings.

5.2. *Warm-start BART MCMC.* In this section, we demonstrate the advantage of initializing BART MCMC at XBART draws. The data generating process is the same as section 5.1.1, and the data size is fixed at 10,000 while noise level κ varies. We fit 40 XBART forests, the first 15 are

TABLE 3

Root mean squared error (RMSE) of each method. Column XGBoost +CV is result of XGBoost with tuning parameter by cross validation and column NN is result of neural networks. The number in parenthesis is running time in seconds. First column is number of data observations (in thousands).

$\kappa = 1$

n	XBART	XGBoost +CV	XGBoost	NN
		Linear		
10k	1.74 (20)	2.63 (64)	3.23 (0)	1.39 (26)
50k	1.04 (180)	1.99 (142)	2.56 (4)	0.66 (28)
250k	0.67 (1774)	1.50 (1399)	2.00 (55)	0.28 (40)
		Max		
10k	0.39 (16)	0.42 (62)	0.79 (0)	0.40 (30)
50k	0.25 (134)	0.29 (140)	0.58 (4)	0.20 (32)
250k	0.14 (1188)	0.21 (1554)	0.41 (60)	0.16 (44)
		Single Index		
10k	2.27 (17)	2.65 (61)	3.65 (0)	2.76 (28)
50k	1.54 (153)	1.61 (141)	2.81 (4)	1.93 (31)
250k	1.14 (1484)	1.18 (1424)	2.16 (55)	1.67 (41)
		Trig + Poly		
10k	1.31 (17)	2.08 (61)	2.70 (0)	3.96 (26)
50k	0.74 (147)	1.29 (141)	1.67 (4)	3.33 (29)
250k	0.45 (1324)	0.82 (1474)	1.11 (59)	2.56 (41)

$\kappa = 10$

n	XBART	XGBoost +CV	XGBoost	NN
		Linear		
10k	5.07 (16)	8.04 (61)	21.25 (0)	7.39 (12)
50k	3.16 (135)	5.47 (140)	16.17 (4)	3.62 (14)
250k	2.03 (1228)	3.15 (1473)	11.49 (54)	1.89 (19)
		Max		
10k	1.94 (16)	2.76 (60)	7.18 (0)	2.98 (15)
50k	1.22 (133)	1.85 (139)	5.49 (4)	1.63 (16)
250k	0.75 (1196)	1.05 (1485)	3.85 (54)	0.85 (22)
		Single Index		
10k	7.13 (16)	10.61 (61)	28.68 (0)	9.43 (14)
50k	4.51 (133)	6.91 (139)	21.18 (4)	6.42 (16)
250k	3.06 (1214)	4.10 (1547)	14.82 (54)	4.72 (21)
		Trig + Poly		
10k	4.94 (16)	7.16 (61)	17.97 (0)	8.20 (13)
50k	3.01 (132)	4.92 (139)	13.30 (4)	5.53 (14)
250k	1.87 (1216)	3.17 (1462)	9.37 (49)	4.13 (20)

thrown out as burn-in draws, and 25 forest draws are retained. BART was fit with a burn-in of 1,000 samples, and 2,500 retrained posterior samples. For the warm-start BART, 25 *independent* BART MCMC chains were initialized at the 25 forest draws obtained from XBART and each was run for 100 iterations with no burn-in. Note that the total number of posterior draws is 2,500, the same as the number of posterior draws by BART. We repeat drawing synthetic data and computing intervals

100 times, all measurement below were taken average with respect to those 100 replications.

TABLE 4

Coverage and length of credible interval of f at 95% level for warm-start BART MCMC. The table also shows running time (in seconds) and root mean squared error (RMSE) of all approaches.

$\kappa = 1$

		XBART	BART	Warm-start BART
Max	coverage	0.86	0.78	0.95
	interval length	0.36	0.35	0.46
	running time	1.57	44.41	1.21 (31.82)
	RMSE	0.11	0.14	0.11
Trig + Poly	coverage	0.90	0.74	0.96
	interval length	3.61	2.89	4.23
	running time	4.68	92.75	3.02 (80.18)
	RMSE	1.03	1.27	1.01
Single Index	coverage	0.77	0.73	0.87
	interval length	4.84	4.62	5.88
	running time	5.10	102.87	2.97 (79.35)
	RMSE	1.94	2.08	1.92
Linear	coverage	0.78	0.77	0.99
	interval length	7.82	6.14	9.92
	running time	5.61	131.17	3.85 (101.86)
	RMSE	3.11	2.51	1.81

$\kappa = 2$

		XBART	BART	Warm-start BART
Max	coverage	0.88	0.84	0.97
	interval length	0.58	0.64	0.76
	running time	1.35	40.23	1.22 (31.85)
	RMSE	0.17	0.22	0.17
Trig + Poly	coverage	0.90	0.82	0.96
	interval length	5.62	5.06	6.86
	running time	3.68	86.81	2.90 (74.17)
	RMSE	1.65	1.87	1.60
Single Index	coverage	0.81	0.83	0.91
	interval length	6.81	7.67	8.49
	running time	3.92	90.70	2.81 (74.0490)
	RMSE	2.51	2.73	2.47
Linear	coverage	0.50	0.83	0.98
	interval length	6.53	8.82	11.84
	running time	3.61	109.43	3.33 (86.86)
	RMSE	4.74	4.13	2.53

Table 4 shows the credible interval coverage, length, RMSE of the point estimate, and running time of the three approaches. The running time for warm-start BART is reported as time in seconds for a single *indepedent* BART MCMC, while the number in parenthesis is the running time of the entire warm-start BART fitting process, including XBART fit and assuming all 25 indepedent warm-start BART MCMC were fitted *sequentially* rather than in parallel. In other words, the

number in parentheses is the most conservative estimation of the total running time, because the 25 independent BART chains can be trivially parallelized to achieve much lower total running time. Indeed, with 25 processors, the run time would be the XBART run time plus the warm-start run time (not in parentheses).

The warm-start BART boasts a substantial advantage in terms of credible interval coverage and root mean squared error. In all cases, warm-start BART has the best coverage and RMSE among all three approaches and is still faster than BART under the most conservative estimation of running time. Especially when the true function is linear, warm-start initialization helps BART get a considerable improvement in estimation, which may indicated inadequate chain length of BART (that is, poor mixing).

6. Discussion. In this paper, we have introduced a novel tree-based ensemble framework, XBART, for supervised learning, that has a wide range of applicability. We demonstrated the advantage of the algorithm in simulation studies: XBART has state-of-the-art prediction accuracy with computational demands that are competitive with alternatives. While this paper has focused on Gaussian nonlinear regression, the proposed algorithm extends to other settings straightforwardly, such as logit multi-class classification, binary probit classification, Poisson regression, or classification by a central-limit approximation; these extensions will be described in separate forthcoming manuscripts.

While we proved the consistency of a single XBART tree in this paper, an open question is whether the forest is consistent or not. One proof strategy would be to make a minor modification of the model wherein an initial tree is fit to the data and then a forest is fit to the residual. By consistency of the initial tree, the residual will eventually converge to pure noise and a proof would need to show that the resulting Markov Chain had expectation of zero in the large data limit.

Another interesting research topic is the connection between XBART and full Bayesian methods. In this paper we show that the warm-start provided by XBART helps the MCMC algorithm converge faster and attain better credible interval coverage. Although we show the forest is a Markov chain with stationary distribution, it is still not clear if the algorithm is a Gibbs sampler corresponding to a valid posterior.

The software package `XBART` is available online at http://www.github.com/jingyuhe/xbart for both `R` and `python`.

References.

Bartlett, P. L. and M. Traskin (2007). Adaboost is consistent. *Journal of Machine Learning Research 8*(Oct), 2347–2368.

Biau, G., L. Devroye, and G. Lugosi (2008). Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research 9*(Sep), 2015–2033.

Breiman, L. (1996). Bagging predictors. *Machine learning 24*(2), 123–140.

Breiman, L. (1997). Arcing the edge. Technical report, Statistics Department, University of California at Berkeley.

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

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

Chen, T. and C. Guestrin (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. ACM.

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

Chipman, H. A., E. I. George, and R. E. McCulloch (2007). Bayesian ensemble learning. In *Advances in neural information processing systems*, pp. 265–272.

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

Chollet, F. et al. (2015). Keras.

Coram, M., S. P. Lalley, et al. (2006). Consistency of Bayes estimators of a binary regression function. *The Annals of Statistics 34*(3), 1233–1269.

Denison, D. G., B. K. Mallick, and A. F. Smith (1998). A Bayesian CART algorithm. *Biometrika 85*(2), 363–377.

Freund, Y. and R. E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences 55*(1), 119–139.

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

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis 38*(4), 367–378.

Gordon, L. and R. A. Olshen (1980). Consistent nonparametric regression from recursive partitioning schemes. *Journal of Multivariate Analysis 10*(4), 611–627.

Gramacy, R. B. and H. K. H. Lee (2008). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association 103*(483), 1119–1130.

Györfi, L., M. Kohler, A. Krzyzak, and H. Walk (2006). *A distribution-free theory of nonparametric regression*. Springer Science & Business Media.

Hahn, P. R., J. S. Murray, C. M. Carvalho, et al. (2020). Bayesian regression tree models for causal inference: regularization, confounding, and heterogeneous effects. *Bayesian Analysis*.

Hazan, T., G. Papandreou, and D. Tarlow (2016). *Perturbations, Optimization, and Statistics*. MIT Press.

He, J., S. Yalov, and P. R. Hahn (2019). XBART: Accelerated Bayesian additive regression trees. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1130–1138.

Hill, J., A. Linero, and J. Murray (2020). Bayesian additive regression trees: A review and look forward. *Annual*

Review of Statistics and Its Application 7.

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

Ishwaran, H., U. B. Kogalur, E. H. Blackstone, M. S. Lauer, et al. (2008). Random survival forests. *The annals of applied statistics 2*(3), 841–860.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pp. 3146–3154.

Kindo, B. P., H. Wang, and E. A. Peña (2016). Multinomial probit Bayesian additive regression trees. *Stat 5*(1), 119–131.

Laurent, B. and P. Massart (2000). Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, 1302–1338.

Lin, Y. and Y. Jeon (2006). Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association 101*(474), 578–590.

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

Linero, A. R. (2018). Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association 113*(522), 626–636.

Linero, A. R., D. Sinha, and S. R. Lipsitz (2019). Semiparametric mixed-scale models using shared bayesian forests. *Biometrics*.

Linero, A. R. and Y. Yang (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.

Logan, B. R., R. Sparapani, R. E. McCulloch, and P. W. Laud (2019). Decision making and uncertainty quantification for individualized treatments using Bayesian additive regression trees. *Statistical methods in medical research 28*(4), 1079–1093.

Mehta, M., R. Agrawal, and J. Rissanen (1996). SLIQ: A fast scalable classifier for data mining. In *International Conference on Extending Database Technology*, pp. 18–32. Springer.

Murray, J. S. (2017). Log-linear Bayesian additive regression trees for multinomial Logistic and count regression models. *arXiv preprint arXiv:1701.01503*.

Pratola, M. (2016). Efficent Metropolis-Hastings proposal mechanism for Bayesian regression tree models. *Bayesian Analysis 11*(3), 885–911.

Pratola, M., H. Chipman, E. George, and R. McCulloch (2017). Heteroscedastic BART using multiplicative regression trees. *arXiv preprint arXiv:1709.07542*.

Pratola, M. T., H. A. Chipman, J. R. Gattiker, D. M. Higdon, R. McCulloch, and W. N. Rust (2014). Parallel Bayesian additive regression trees. *Journal of Computational and Graphical Statistics 23*(3), 830–852.

Rocková, V. (2019). A note on semi-parametric Bernstein-von Mises theorems for BART priors.

Ročková, V. and E. Saha (2019). On theory for BART. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2839–2848.

Rocková, V. and S. van der Pas (2017). Posterior concentration for Bayesian regression trees and forests. *arXiv*

preprint arXiv:1708.08734.

Scornet, E., G. Biau, J.-P. Vert, et al. (2015). Consistency of random forests. *The Annals of Statistics 43*(4), 1716–1741.

Starling, J. E., J. S. Murray, C. M. Carvalho, R. Bukowski, and J. G. Scott (2018). Functional response regression with funBART: an analysis of patient-specific stillbirth risk. *arXiv preprint arXiv:1805.07656*.

Starling, J. E., J. S. Murray, P. A. Lohr, A. R. Aiken, C. M. Carvalho, and J. G. Scott (2019). Targeted smooth bayesian causal forests: An analysis of heterogeneous treatment effects for simultaneous versus interval medical abortion regimens over gestation. *arXiv preprint arXiv:1905.09405*.

van der Pas, S. and V. Ročková (2017). Bayesian dyadic trees and histograms for regression. In *Advances in Neural Information Processing Systems*, pp. 2089–2099.

Wager, S. and S. Athey (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association 113*(523), 1228–1242.

Wright, M. N. and A. Ziegler (2015). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *arXiv preprint arXiv:1508.04409*.

Zhang, T., B. Yu, et al. (2005). Boosting with early stopping: Convergence and consistency. *The Annals of Statistics 33*(4), 1538–1579.

Supplementary Material

APPENDIX A: CATEGORICAL COVARIATES

Section 2.5.1 suggests pre-sorting covariates to compute sufficient statistics efficiently, this strategy is straightforward for continuous covariates. However, because of possible ties in ordered categorical covariates, a more efficient algorithm is needed to calculate sufficient statistics.

We restate notations in section 2.5.1. Without loss of generality, we assume that all covariates are categorical. Let $\mathbf{O}$ denote the V-by-n array such that o_{vh} denotes the index, in the data, of the observation with the h-th smallest value of the v-th predictor variable x_v. Then, taking the cumulative sums gives

$$s(\leq, v, c) = \sum_{h \leq c} \mathrm{r}_{o_{vh}}$$

and

$$s(>, v, c) = \sum_{h=1}^{n} r_{lh} - s(\leq, v, c).$$

Algorithm 4 Pseudocode of calculating sufficient statistics for categorical covariates.

1: Sort categorical covariates, create $\mathbf{O}$ matrix. Count number of unique observations `unique_val` and `val_count` vector (suppose vectors are length K).
2: **for** i from 1 to K **do**
3: Calculate sufficient statistics for cutpoint candidate `unique_val`[i] as

$$s(\leq, v, \texttt{unique\_val}[i]) = \sum_{h \in [\sum_{m=1}^{i-1} \texttt{val\_count[m]}, \sum_{m=1}^{i} \texttt{val\_count[m]}]} r_{o_{vh}}.$$

and

$$s(>, v, c) = \sum_{h=1}^{n} r_{lh} - s(\leq, v, c).$$

4: **end for**
5: Calculate split criterion, determine a cutpoint.
6: **if** no-split is selected or stop conditions are reached **then**
7: Draw leaf parameters and **return**.
8: **else**
9: Sift `unique_val` and `val_count` for left and right child nodes. Repeat step 3 when evaluate split criterion at child nodes.
10: **end if**

The subscript l on the residual indicates that these evaluations pertain to the update of the lth tree. Notice that when covariates are categorical, x_{vh} is not necessarily smaller than $x_{v(h+1)}$ due to potential ties in x. As a result, the number of unique cutpoint candidates is less than n. We propose an extra data structure to bookkeeping unique cutpoint and number of ties as follows. For the v-th categorical predictor variable x_v, let `unique_val` be a vector of unique values (sorted,

from small to large) in x_v and `val_count` be a vector of counts of replication for each unique value. Therefore, the cutpoint candidate is a element in the vector `unique_val`, say the i-th element. Then the cumulative sums is

$$s(\leq, v, \texttt{unique\_val}[i]) = \sum_{h \in [\sum_{m=1}^{i-1} \texttt{val\_count}[m], \sum_{m=1}^{i} \texttt{val\_count}[m]]} r_{o_{vh}}.$$

When sifting data to left and right child after drawing a cutpoint, we create the same `unique_val` and `val_count` vector for all categorical covariates with data in two child nodes respectively. See Algorithm 4 for details.

APPENDIX B: PROOF OF LEMMA 2

First we establish the connection between theoretical split criterion of XBART (equation (19)) and CART. For current node A, the theoretical split criterion of XBART for candidate split at variable i and value x is

$$\begin{aligned} L^*(x) = &\frac{1}{\sigma^2}\mathbb{P}(\mathbf{x}^{(i)} \leq x \mid \mathbf{x} \in A)\left[\mathbb{E}(y \mid \mathbf{x}^{(i)} \leq x, \mathbf{x} \in A)\right]^2 \\ &+ \frac{1}{\sigma^2}\mathbb{P}(\mathbf{x}^{(i)} > x \mid \mathbf{x} \in A)\left[\mathbb{E}(y \mid \mathbf{x}^{(i)} > x, \mathbf{x} \in A)\right]^2. \end{aligned} \tag{22}$$

The CART theoretical split criterion is

$$\begin{aligned} L^*_{\text{CART}}(x) = &\mathbb{V}(y \mid \mathbf{x} \in A) - \mathbb{P}(\mathbf{x}^{(j)} \leq x \mid \mathbf{x} \in A)\mathbb{V}(y \mid \mathbf{x}^{(j)} \leq x, \mathbf{x} \in A) \\ &- \mathbb{P}(\mathbf{x}^{(j)} > x \mid \mathbf{x} \in A)\mathbb{V}(y \mid \mathbf{x}^{(j)} > x, \mathbf{x} \in A). \end{aligned} \tag{23}$$

Remember that the cuts is always parallel to axis,

$$\mathbb{V}(y \mid \mathbf{x}^{(j)} \leq x, \mathbf{x} \in A) = \mathbb{E}(y^2 \mid \mathbf{x}^{(j)} \leq x, \mathbf{x} \in A) - \left[\mathbb{E}(y \mid \mathbf{x}^{(j)} \leq x, \mathbf{x} \in A)\right]^2. \tag{24}$$

We have

$$\mathbb{E}(y^2 \mid \mathbf{x}^{(j)} \leq x, \mathbf{x} \in A) = \frac{1}{\Omega\left(\left\{\mathbf{x}^{(j)} \leq x, \mathbf{x} \in A\right\}\right)} \int_{\mathbf{x} \in \left\{\mathbf{x}^{(j)} \leq x, \mathbf{x} \in A\right\}} m^2(\mathbf{x}) d\mathbf{x}, \tag{25}$$

where $\Omega(A)$ represents volume of a cube A. Observe that

$$\mathbb{P}(\mathbf{x}^{(j)} \leq x \mid \mathbf{x} \in A) = \frac{\Omega\left(\left\{\mathbf{x}^{(j)} \leq x, \mathbf{x} \in A\right\}\right)}{\Omega(A)},$$

by easy calculation, we obtain

$$
\begin{aligned}
&\mathbb{E}(y^2 \mid \mathbf{x} \in A) - \mathbb{P}(\mathbf{x}^{(j)} \le x \mid \mathbf{x} \in A)\mathbb{E}(y^2 \mid \mathbf{x}^{(j)} \le x, \mathbf{x} \in A) \\
&\quad - \mathbb{P}(\mathbf{x}^{(j)} > x \mid \mathbf{x} \in A)\mathbb{E}(y^2 \mid \mathbf{x}^{(j)} > x, \mathbf{x} \in A) \\
&= \frac{1}{\Omega(A)} \int_{\mathbf{x} \in A} m^2(\mathbf{x}) d\mathbf{x} - \frac{1}{\Omega(A)} \int_{\mathbf{x} \in \left\{\mathbf{x}^{(j)} \le x, \mathbf{x} \in A\right\}} m^2(\mathbf{x}) d\mathbf{x} - \frac{1}{\Omega(A)} \int_{\mathbf{x} \in \left\{\mathbf{x}^{(j)} > x, \mathbf{x} \in A\right\}} m^2(\mathbf{x}) d\mathbf{x} \\
&= 0.
\end{aligned} \tag{26}
$$

As a result, the CART theoretical split criterion is equivalent to

$$
\begin{aligned}
L^*_{\mathrm{CART}}(x) &= [\mathbb{E}(y \mid \mathbf{x} \in A)]^2 - \mathbb{P}(\mathbf{x}^{(i)} \le x \mid \mathbf{x} \in A) \left[\mathbb{E}(y \mid \mathbf{x}^{(i)} \le x, \mathbf{x} \in A)\right]^2 \\
&\quad - \mathbb{P}(\mathbf{x}^{(i)} > x \mid \mathbf{x} \in A) \left[\mathbb{E}(y \mid \mathbf{x}^{(i)} > x, \mathbf{x} \in A)\right]^2 \\
&= [\mathbb{E}(y \mid \mathbf{x} \in A)]^2 - \sigma^2 L^*_{\mathrm{XBART}}(x).
\end{aligned} \tag{27}
$$

Since $[\mathbb{E}(y \mid \mathbf{x} \in A)]^2$ and σ^2 are constant, and we maximize $L^*_{\mathrm{XBART}}(x)$ but minimize $L^*_{\mathrm{CART}}(x)$ in practice, we claim that the two *theoretical* split criterions are equivalent. Therefore, the proof of lemma 2 for CART case in Scornet et al. (2015) can be applied directly without modification. We refer readers to Scornet et al. (2015) for details.

APPENDIX C: PROOF OF LEMMA 3

C.1. Proof of Lemma 3 for the case $k = 1$. Preliminary results Let $Z_i = \max_{1 \le i \le n} |\epsilon_i|$, we have

$$
\mathbb{P}(Z_i \ge t) = 1 - \exp\left[n \ln(1 - 2\mathbb{P}(\epsilon_i \ge t))\right].
$$

The tail of Gaussian distribution has a standard bound:

$$
\mathbb{P}(\epsilon_i \ge t) \le \frac{\sigma}{t\sqrt{2\pi}} \left(-\frac{t^2}{2\sigma^2}\right). \tag{28}
$$

As a result, there exist a positive constant C_ρ and $N_1 \in \mathbb{N}^*$ such that with probability $1 - \rho$, for all $n > N_1$,

$$
\max_{1 \le i \le n} |\epsilon_i| \le C_\rho \sqrt{\log(n)}. \tag{29}
$$

In addition, we have

$$
\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n} \epsilon_i\right| \ge \alpha\right] \le \frac{\sigma}{\alpha\sqrt{n}} \exp\left(-\frac{\alpha^2 n}{2\sigma^2}\right). \tag{30}
$$

Let $\mathrm{N}_n(A)$ denotes number of data observations in a set A. Next we derive from the inequality above and union bound inequality that there exists $N_2 \in \mathbb{N}^*$ such that with probability $1 - \rho$, for all $n > N_2$ and all $0 \le a_n \le b_n \le 1$ satisfying $\mathrm{N}_n([a_n, b_n] \times [0,1]^{p-1}) > \sqrt{n}$,

$$
\left|\frac{1}{\mathrm{N}_n([a_n, b_n] \times [0,1]^{p-1})} \sum_{i : X_i \in [a_n, b_n] \times [0,1]^{p-1}} \epsilon_i\right| \le \alpha. \tag{31}
$$

and

$$\frac{1}{\mathrm{N}_n([a_n,b_n]\times[0,1]^{p-1})}\sum_{i:X_i\in[a_n,b_n]\times[0,1]^{p-1}}\epsilon_i^2\leq\tilde{\sigma}^2. \tag{32}$$

Furthermore, it's easy to verify

$$\left|\frac{1}{\mathrm{N}_n([a_n,b_n]\times[0,1]^{p-1})}\sum_{i:X_i\in[a_n,b_n]\times[0,1]^{p-1}}Y_i\right|\leq||m||_\infty+\alpha, \tag{33}$$

and

$$\left|\frac{1}{\mathrm{N}_n([a_n,b_n]\times[0,1]^{p-1})}\sum_{i:X_i\in[a_n,b_n]\times[0,1]^{p-1}}Y_i^2\right|\leq||m||_\infty^2+\tilde{\sigma}^2+2\alpha||m||_\infty. \tag{34}$$

By the Glivenko-Cantelli theorem, there exist $N_3\in\mathbb{N}^*$ such that with probability $1-\rho$, for all $0\leq a\leq b\leq 1$ and all $n>N_3$,

$$(b-a-\delta^2)n\leq\mathrm{N}_n([a_n,b_n]\times[0,1]^{p-1})\leq(b-a+\delta^2)n. \tag{35}$$

In the following proof, we assume to be on the event that all claims above holds with probability $1-3\rho$ for all $n>N=\max\{N_1,N_2,N_3\}$. Take $x_1,x_2\in[0,1]$ such that $|x_1-x_2|<\delta$ and assume that $x_1<x_2$. We partition the space $[0,1]^p$ into several pieces as follows, see Figure 3 for an illustration of notations for $p=2$.

$$\begin{cases}A_{L,\sqrt{\delta}}=[0,\sqrt{\delta}]\times[0,1]^{p-1}\\A_{R,\sqrt{\delta}}=[1-\sqrt{\delta},1]\times[0,1]^{p-1}\\A_{C,\sqrt{\delta}}=[\sqrt{\delta},1-\sqrt{\delta}]\times[0,1]^{p-1}\end{cases}. \tag{36}$$

Similarly,

$$\begin{cases}A_{L,1}=[0,x_1]\times[0,1]^{p-1}\\A_{R,1}=[x_1,1]\times[0,1]^{p-1}\\A_{L,2}=[0,x_2]\times[0,1]^{p-1}\\A_{R,2}=[x_2,1]\times[0,1]^{p-1}\\A_C=[x_1,x_2]\times[0,1]^{p-1}\end{cases}. \tag{37}$$

Figure 3 illustrates notations

For simplicity, we write the split criterion of the first cut as $L_{n,1}(1,x)$ denoting split at the first variable,

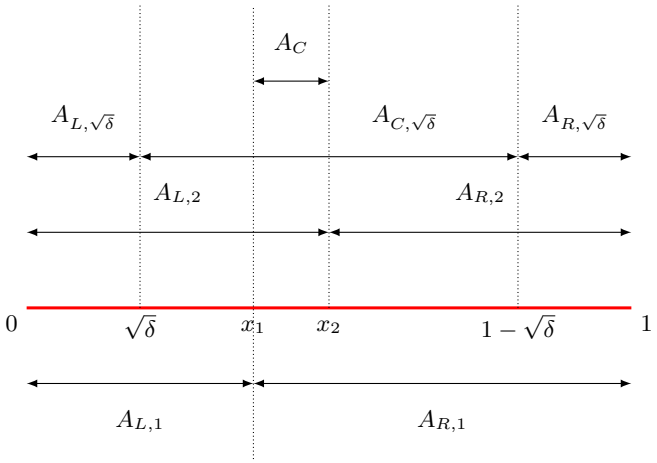


Fig 3: Illustration of notations for $p = 2$.

at value x. Recall that our split criterion is defined as

$$
\begin{aligned}
L_{n,1}(1,x) = &\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_L)\right)}\frac{1}{n}\left(\mathrm{N}_n(A_L)\sum_{i:\mathbf{x}_i^{(1)}\le x} y_i^2 - (\mathrm{N}_n(A_L)-1)\sum_{i:\mathbf{x}_i^{(1)}\le x}(y_i-\bar{y}_l)^2\right)\\
&+\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_R)\right)}\frac{1}{n}\left(\mathrm{N}_n(A_R)\sum_{i:\mathbf{x}_i^{(1)}> x} y_i^2 - (\mathrm{N}_n(A_R)-1)\sum_{i:\mathbf{x}_i^{(1)}> x}(y_i-\bar{y}_r)^2\right)\\
&+\frac{\gamma_x}{n}.
\end{aligned}
\tag{38}
$$

The difference of split criterion on two cutpoints x_1 and x_2 is

$$
\begin{aligned}
&L_{n,1}(1,x_1) - L_{n,1}(1,x_2)\\
=&\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,1})\right)}\frac{1}{n}\left(\mathrm{N}_n(A_{L,1})\sum_{i:\mathbf{x}_i^{(1)}\le x_1} y_i^2 - (\mathrm{N}_n(A_{L,1})-1)\sum_{i:\mathbf{x}_i^{(1)}\le x_1}(y_i-\bar{y}_{A_{L,1}})^2\right)\\
&+\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{R,1})\right)}\frac{1}{n}\left(\mathrm{N}_n(A_{R,1})\sum_{i:\mathbf{x}_i^{(1)}> x_1} y_i^2 - (\mathrm{N}_n(A_{R,1})-1)\sum_{i:\mathbf{x}_i^{(1)}> x_1}(y_i-\bar{y}_{A_{R,1}})^2\right)\\
&-\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}\frac{1}{n}\left(\mathrm{N}_n(A_{L,2})\sum_{i:\mathbf{x}_i^{(1)}\le x_2} y_i^2 - (\mathrm{N}_n(A_{L,2})-1)\sum_{i:\mathbf{x}_i^{(1)}\le x_2}(y_i-\bar{y}_{A_{L,2}})^2\right)\\
&-\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{R,2})\right)}\frac{1}{n}\left(\mathrm{N}_n(A_{R,2})\sum_{i:\mathbf{x}_i^{(1)}> x_2} y_i^2 - (\mathrm{N}_n(A_{R,2})-1)\sum_{i:\mathbf{x}_i^{(1)}> x_2}(y_i-\bar{y}_{A_{R,2}})^2\right)\\
&+\frac{\gamma_{x_1}}{n}-\frac{\gamma_{x_2}}{n}.
\end{aligned}
\tag{39}
$$

We need to prove lemma 3 for all possible cases depending on location of x_1 and x_2. For notation simplicity, note that after collecting terms, the difference of split criterion can be represented as summation of points

for the range of index $\{i : \mathbf{x}_i^{(1)} < x_1\}$, $\{i : \mathbf{x}_i^{(1)} \in [x_1, x_2]\}$ and $\{i : \mathbf{x}_i^{(1)} > x_2\}$. We will use the same decomposition throughout the proof.

First case

Assume that $x_1, x_2 \in A_{C,\sqrt{\delta}}$, two cutpoint candidates are not close to the edge. Consider the split criterion

$$
\begin{aligned}
& L_{n,1}(1, x_1) - L_{n,1}(1, x_2) \\
=& \frac{\tau}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,1})\right)} \frac{1}{n} \left(\mathrm{N}_n(A_{L,1}) \sum_{i:\mathbf{x}_i^{(1)} \le x_1} y_i^2 - (\mathrm{N}_n(A_{L,1}) - 1) \sum_{i:\mathbf{x}_i^{(1)} \le x_1} (y_i - \bar{y}_{A_{L,1}})^2 \right) \\
& + \frac{\tau}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{R,1})\right)} \frac{1}{n} \left(\mathrm{N}_n(A_{R,1}) \sum_{i:\mathbf{x}_i^{(1)} > x_1} y_i^2 - (\mathrm{N}_n(A_{R,1}) - 1) \sum_{i:\mathbf{x}_i^{(1)} > x_1} (y_i - \bar{y}_{A_{R,1}})^2 \right) \\
& - \frac{\tau}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,2})\right)} \frac{1}{n} \left(\mathrm{N}_n(A_{L,2}) \sum_{i:\mathbf{x}_i^{(1)} \le x_2} y_i^2 - (\mathrm{N}_n(A_{L,2}) - 1) \sum_{i:\mathbf{x}_i^{(1)} \le x_2} (y_i - \bar{y}_{A_{L,2}})^2 \right) \\
& - \frac{\tau}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{R,2})\right)} \frac{1}{n} \left(\mathrm{N}_n(A_{R,2}) \sum_{i:\mathbf{x}_i^{(1)} > x_2} y_i^2 - (\mathrm{N}_n(A_{R,2}) - 1) \sum_{i:\mathbf{x}_i^{(1)} > x_2} (y_i - \bar{y}_{A_{R,2}})^2 \right) \\
& + \frac{\gamma_{x_1}}{n} - \frac{\gamma_{x_2}}{n} \\
=& J_1 + J_2 + J_3 + \frac{\gamma_{x_1}}{n} - \frac{\gamma_{x_2}}{n}.
\end{aligned} \tag{40}
$$

First, take n large enough, we have

$$
\left| \frac{\gamma_{x_1}}{n} - \frac{\gamma_{x_2}}{n} \right| \le \alpha. \tag{41}
$$

Let J_2 corresponding to $\{i \mid \mathbf{x}_i^{(1)} \in [x_1, x_2]\}$

$$
\begin{aligned}
J_2 = & \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{R,1})\right)} \frac{1}{n}\left(\mathrm{N}_n(A_{R,1}) \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]} y_i^2-\left(\mathrm{N}_n(A_{R,1})-1\right) \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]}\left(y_i-\bar{y}_{A_{R,1}}\right)^2\right) \\
& -\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)} \frac{1}{n}\left(\mathrm{N}_n(A_{L,2}) \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]} y_i^2\right. \\
& \left.-\left(\mathrm{N}_n(A_{L,2})-1\right) \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]}\left(y_i-\bar{y}_{A_{L,2}}\right)^2\right) \\
= & \frac{\tau \mathrm{N}_n(A_{R,1})}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{R,1})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]} y_i^2\right)-\frac{\tau \mathrm{N}_n(A_{L,2})}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]} y_i^2\right) \\
& +\frac{\tau\left(\mathrm{N}_n(A_{L,2})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]}\left(y_i-\bar{y}_{A_{L,2}}\right)^2\right) \\
& -\frac{\tau\left(\mathrm{N}_n(A_{R,1})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{R,1})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]}\left(y_i-\bar{y}_{A_{R,1}}\right)^2\right) \\
= & J_{21}+J_{22}.
\end{aligned} \tag{42}
$$

Note that $|ax - by| \leq |a||x-y| + |a-b||y|$, we have

$$
\begin{aligned}
|J_{22}| = & \left|\frac{\tau\left(\mathrm{N}_n(A_{L,2})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]}\left(y_i-\bar{y}_{A_{L,2}}\right)^2\right)\right. \\
& \left.-\frac{\tau\left(\mathrm{N}_n(A_{R,1})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{R,1})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]}\left(y_i-\bar{y}_{A_{R,1}}\right)^2\right)\right| \\
\leq & \left|\frac{\tau\left(\mathrm{N}_n(A_{L,2})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}\right| \times\left|\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]}\left(y_i-\bar{y}_{A_{L,2}}\right)^2-\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]}\left(y_i-\bar{y}_{A_{R,1}}\right)^2\right| \\
& +\left|\frac{\tau\left(\mathrm{N}_n(A_{L,2})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}-\frac{\tau\left(\mathrm{N}_n(A_{R,1})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{R,1})\right)}\right| \times\left|\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \in [x_1,x_2]}\left(y_i-\bar{y}_{A_{R,1}}\right)^2\right| .
\end{aligned} \tag{43}
$$

Since we assume that $x_1, x_2 \in A_{C,\sqrt{\delta}}$, by equation (35)

$$
\begin{aligned}
\left|\frac{\tau\left(\mathrm{N}_n(A_{L,2})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}\right| & \leq\left|\frac{\tau(\delta^2-\sqrt{\delta}) n}{\sigma^2(\sigma^2+\tau(1-\delta^2-\sqrt{\delta}) n)}\right| \\
& \leq\left|\frac{\tau(\delta^2-\sqrt{\delta})}{\sigma^2(\tau(1-\delta^2-\sqrt{\delta}))}\right| \\
& =C(\delta) \rightarrow 0 \text{ as } \delta \rightarrow 0.
\end{aligned} \tag{44}
$$

Note that this bound is valid for $\mathrm{N}_n(A_{L,1}), \mathrm{N}_n(A_{L,2}), \mathrm{N}_n(A_{R,1})$ and $\mathrm{N}_n(A_{R,2})$. By inequality (33) and (34), it is obvious that

$$\left|\frac{1}{n}\sum_{i:\mathbf{x}_i^{(1)}\in[x_1,x_2]}(y_i-\bar{y}_{A_{R,1}})^2\right| \leq \left|\frac{1}{N(A_C)}\sum_{i:\mathbf{x}_i^{(1)}\in[x_1,x_2]}(y_i-\bar{y}_{A_{R,1}})^2\right| \leq M \tag{45}$$

by a constant M. Furthermore

$$\left|\frac{\tau(\mathrm{N}_n(A_{L,2})-1)}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,2})\right)} - \frac{\tau(\mathrm{N}_n(A_{R,1})-1)}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{R,1})\right)}\right| \leq 2C(\delta). \tag{46}$$

The bound of second term follows equation (8) in supplementary materials of Scornet et al. (2015) directly,

$$|J_{22}| \leq C(\delta)\times 4(||m||_\infty+\alpha)\left((\delta+\delta^2)(2||m||_\infty+\alpha)+\alpha\right)+2C(\delta)M. \tag{47}$$

The other term J_{21} is

$$\begin{aligned} |J_{21}| &= \left|\frac{\tau\mathrm{N}_n(A_{R,1})}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{R,1})\right)}\left(\frac{1}{n}\sum_{i:\mathbf{x}_i^{(1)}\in[x_1,x_2]}y_i^2\right) - \frac{\tau\mathrm{N}_n(A_{L,2})}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,2})\right)}\left(\frac{1}{n}\sum_{i:\mathbf{x}_i^{(1)}\in[x_1,x_2]}y_i^2\right)\right| \\ &\leq \left|\frac{\tau\mathrm{N}_n(A_{R,1})}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{R,1})\right)} - \frac{\tau\mathrm{N}_n(A_{L,2})}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,2})\right)}\right| \times \left|\frac{1}{n}\sum_{i:\mathbf{x}_i^{(1)}\in[x_1,x_2]}y_i^2\right|. \end{aligned} \tag{48}$$

The bound of coefficient here is slightly different from equation (44) and (46)

$$\begin{aligned} &\left|\frac{\tau\mathrm{N}_n(A_{R,1})}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{R,1})\right)} - \frac{\tau\mathrm{N}_n(A_{L,2})}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,2})\right)}\right| \\ =&\left|\frac{\tau(\mathrm{N}_n(A_{R,1})-\mathrm{N}_n(A_{L,2}))}{(\sigma^2+\tau\mathrm{N}_n(A_{R,1}))(\sigma^2+\tau\mathrm{N}_n(A_{L,2}))}\right| \\ \leq&\left|\frac{\tau}{(\sigma^2+\tau\mathrm{N}_n(A_{R,1}))(\sigma^2+\tau\mathrm{N}_n(A_{L,2}))}\right|(|\mathrm{N}_n(A_{R,1})|+|\mathrm{N}_n(A_{L,2})|) \\ \leq&\frac{2\tau(1-\sqrt{\delta}+\delta^2)n}{\left(\sigma^2+\tau(1-\sqrt{\delta}-\delta^2)n\right)^2} = g(\delta,n)\to 0 \text{ when } n \text{ is large.} \end{aligned} \tag{49}$$

Note that the upper bounds in equation (44) and (46) can be arbitrarily small if $\delta\to 0$, but the upper bound in equation (49) relies on making n large. Use the tail bound of non-central χ^2 distribution, result of supplementary materials of Scornet et al. (2015), and similar to J_{22}

$$|J_{21}| \leq g(\delta,n)M, \tag{50}$$

which can be arbitrarily small when n is large.

Now we switch to J_1, corresponding to $i \mid X_i^{(1)} \in [0, x_1]$, we proceed with similar decomposition.

$$
\begin{aligned}
J_1 =& \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,1})\right)} \frac{1}{n}\left(\mathrm{N}_n(A_{L,1}) \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} y_i^2-\left(\mathrm{N}_n(A_{L,1})-1\right) \sum_{i:\mathbf{x}_i^{(1)} \leq x_1}\left(y_i-\bar{y}_{A_{L,1}}\right)^2\right) \\
& -\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)} \frac{1}{n}\left(\mathrm{N}_n(A_{L,2}) \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} y_i^2-\left(\mathrm{N}_n(A_{L,2})-1\right) \sum_{i:\mathbf{x}_i^{(1)} \leq x_1}\left(y_i-\bar{y}_{A_{L,2}}\right)^2\right) \\
=& \frac{\tau \mathrm{N}_n(A_{L,1})}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,1})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} y_i^2\right)-\frac{\tau \mathrm{N}_n(A_{L,2})}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} y_i^2\right) \\
& +\frac{\tau\left(\mathrm{N}_n(A_{L,2})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1}\left(y_i-\bar{y}_{A_{L,2}}\right)^2\right) \\
& -\frac{\tau\left(\mathrm{N}_n(A_{L,1})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,1})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1}\left(y_i-\bar{y}_{A_{L,1}}\right)^2\right) \\
=& J_{11}+J_{12}.
\end{aligned} \tag{51}
$$

$$
\begin{aligned}
|J_{12}| =& \left|\frac{\tau\left(\mathrm{N}_n(A_{L,2})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1}\left(y_i-\bar{y}_{A_{L,2}}\right)^2\right)\right. \\
& \left.-\frac{\tau\left(\mathrm{N}_n(A_{L,1})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,1})\right)}\left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1}\left(y_i-\bar{y}_{A_{L,1}}\right)^2\right)\right| \\
\leq & \left|\frac{\tau\left(\mathrm{N}_n(A_{L,2})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}\right| \times\left|\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1}\left(y_i-\bar{y}_{A_{L,2}}\right)^2-\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1}\left(y_i-\bar{y}_{A_{L,1}}\right)^2\right| \\
& +\left|\frac{\tau\left(\mathrm{N}_n(A_{L,2})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,2})\right)}-\frac{\tau\left(\mathrm{N}_n(A_{L,1})-1\right)}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{L,1})\right)}\right| \times\left|\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1}\left(y_i-\bar{y}_{A_{L,1}}\right)^2\right| .
\end{aligned} \tag{52}
$$

Same as J_{22},

$$
\begin{aligned}
|J_{12}| & \leq C(\delta) \times\left|\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \in [x_1, x_2]}\left(y_i-\bar{y}_{A_{R,1}}\right)^2\right|+2 C(\delta) M \\
& \leq C(\delta) \times 5(||m||_{\infty} \sqrt{\delta}+\alpha)+2 C(\delta) M .
\end{aligned} \tag{53}
$$

The second equation above use result of equation (9) of supplementary material of Scornet et al. (2015).

Similar to J_{21}, we have

$$
\begin{aligned}
|J_{11}| &= \left| \frac{\tau \mathrm{N}_n(A_{L,1})}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,1})\right)} \left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \le x_1} y_i^2 \right) - \frac{\tau \mathrm{N}_n(A_{L,2})}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,2})\right)} \left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \le x_1} y_i^2 \right) \right| \\
&\le \left| \frac{\tau \mathrm{N}_n(A_{L,1})}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,1})\right)} - \frac{\tau \mathrm{N}_n(A_{L,2})}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,2})\right)} \right| \times \left| \frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \le x_1} y_i^2 \right| \\
&\le g(\delta, n) M.
\end{aligned} \tag{54}
$$

J_3 have the same bound as J_1. Collect all terms, we have

$$
\begin{aligned}
&|J_1| \le g(\delta, n) M + C(\delta) \times 25(||m||_\infty \sqrt{\delta} + \alpha) + 2C(\delta) M \\
&|J_2| \le g(\delta, n) M + C(\delta) \times 4(||m||_\infty + \alpha)\left((\delta + \delta^2)(2||m||_\infty + \alpha) + \alpha\right) + 2C(\delta) M \\
&|J_3| \le g(\delta, n) M + C(\delta) \times 25(||m||_\infty \sqrt{\delta} + \alpha) + 2C(\delta) M \\
&|L_{n,1}(1, x_1) - L_{n,1}(1, x_2)| \le |J_1| + |J_2| + |J_3|
\end{aligned} \tag{55}
$$

Consequently, for all n large enough and δ small enough, we have

$$
|L_{n,1}(1, x_1) - L_{n,1}(1, x_2)| \le 3\alpha. \tag{56}
$$

Second case

Assume that $x_1, x_2 \in A_{L,\sqrt{\delta}}$, take same arguments as above, we have

$$\mathrm{N}_n(A_{L,1}), \mathrm{N}_n(A_{L,2}) \leq (\sqrt{\delta} + \delta^2)n. \tag{57}$$

Different from the first case, now both x_1 and x_2 are close to the left edge, which is corresponding to term J_1. Note that $|J_2|$ and $|J_3|$ are the same as the first case since the control over region A_C and $A_{R,1} \times A_{R,2}$ and not changed.

$$\begin{aligned}
|J_{12}| = & \left| \frac{\tau(\mathrm{N}_n(A_{L,2}) - 1)}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,2})\right)} \left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} (y_i - \bar{y}_{A_{L,2}})^2 \right) \right. \\
& \left. - \frac{\tau(\mathrm{N}_n(A_{L,1}) - 1)}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,1})\right)} \left(\frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} (y_i - \bar{y}_{A_{L,1}})^2 \right) \right| \\
\leq & \left| \frac{\tau(\mathrm{N}_n(A_{L,2}) - 1)}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,2})\right)} \right| \times \left| \frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} (y_i - \bar{y}_{A_{L,2}})^2 - \frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} (y_i - \bar{y}_{A_{L,1}})^2 \right| \\
& + \left| \frac{\tau(\mathrm{N}_n(A_{L,2}) - 1)}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,2})\right)} - \frac{\tau(\mathrm{N}_n(A_{L,1}) - 1)}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,1})\right)} \right| \times \left| \frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} (y_i - \bar{y}_{A_{L,1}})^2 \right|.
\end{aligned} \tag{58}$$

We have

$$\left| \frac{\tau(\mathrm{N}_n(A_{L,2}) - 1)}{\sigma^2\left(\sigma^2 + \tau \mathrm{N}_n(A_{L,2})\right)} \right| \leq \left| \frac{\tau \mathrm{N}_n(A_{L,2})}{\sigma^2 \tau \mathrm{N}_n(A_{L,2})} \right| = \frac{1}{\sigma^2} \tag{59}$$

$$\begin{aligned}
& \left| \frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} (y_i - \bar{y}_{A_{L,2}})^2 - \frac{1}{n} \sum_{i:\mathbf{x}_i^{(1)} \leq x_1} (y_i - \bar{y}_{A_{L,1}})^2 \right| \\
= & 2|\bar{y}_{A_{L,1}} - \bar{y}_{A_{L,2}}| \times \frac{1}{n} \left| \sum_{i:\mathbf{x}_i^{(1)} < x_1} \left(y_i - \frac{\bar{y}_{A_{L,1}} + \bar{y}_{A_{L,2}}}{2} \right) \right| \\
\leq & 4(||m||_\infty + \alpha) \left(\frac{(||m||_\infty + \alpha)\mathrm{N}_n(A_{L,1})}{n} + \frac{1}{n} \left| \sum_{i:\mathbf{x}_i^{(1)} < x_1} m(\mathbf{x}_i) + \epsilon_i \right| \right) \\
\leq & 4(||m||_\infty + \alpha) \left((||m||_\infty + \alpha)(\sqrt{\delta} + \delta^2) + \frac{\mathrm{N}_n(A_{L,1})}{n} (||m||_\infty + \alpha) \right) \\
\leq & 4(||m||_\infty + \alpha) \left((||m||_\infty + \alpha + 1)(\sqrt{\delta} + \delta^2) \right)
\end{aligned} \tag{60}$$

$$
\begin{aligned}
&\left|\frac{\tau(\mathrm{N}_n(A_{L,2})-1)}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,2})\right)}-\frac{\tau(\mathrm{N}_n(A_{L,1})-1)}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,1})\right)}\right|\times\left|\frac{1}{n}\sum_{i:\mathbf{x}_i^{(1)}\le x_1}(y_i-\bar{y}_{A_{L,1}})^2\right| \\
(61)\qquad &=\left|\frac{\tau(\mathrm{N}_n(A_{L,2})-1)}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,2})\right)}-\frac{\tau(\mathrm{N}_n(A_{L,1})-1)}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,1})\right)}\right|\times\frac{\mathrm{N}_n(A_{L,1})}{n}\left|\frac{1}{\mathrm{N}_n(A_{L,1})}\sum_{i:\mathbf{x}_i^{(1)}\le x_1}(y_i-\bar{y}_{A_{L,1}})^2\right| \\
&\le\frac{2}{\sigma^2}(\sqrt{\delta}+\delta^2)M.
\end{aligned}
$$

As a result

$$
|J_{12}|\le\frac{1}{\sigma^2}4(||m||_\infty+\alpha)\left((||m||_\infty+\alpha+1)(\sqrt{\delta}+\delta^2)\right)+\frac{2}{\sigma^2}(\sqrt{\delta}+\delta^2)M\to 0. \tag{62}
$$

$$
\begin{aligned}
|J_{11}|&=\left|\frac{\tau\mathrm{N}_n(A_{L,1})}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,1})\right)}\left(\frac{1}{n}\sum_{i:\mathbf{x}_i^{(1)}\le x_1}y_i^2\right)-\frac{\tau\mathrm{N}_n(A_{L,2})}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,2})\right)}\left(\frac{1}{n}\sum_{i:\mathbf{x}_i^{(1)}\le x_1}y_i^2\right)\right| \\
(63)\qquad &\le\left|\frac{\tau\mathrm{N}_n(A_{L,1})}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,1})\right)}-\frac{\tau\mathrm{N}_n(A_{L,2})}{\sigma^2\left(\sigma^2+\tau\mathrm{N}_n(A_{L,2})\right)}\right|\times\left|\frac{1}{n}\sum_{i:\mathbf{x}_i^{(1)}\le x_1}y_i^2\right| \\
&\le\frac{2}{\sigma^2}(\sqrt{\delta}+\delta^2)M\to 0.
\end{aligned}
$$

Consequently we conclude that for all $n>N$ and all δ small enough,

$$
|L_{n,1}(1,x_1)-L_{n,1}(1,x_2)|\le 3\alpha. \tag{64}
$$

The other cases $\{x_1,x_2\in A_{R,\sqrt{\delta}}\}$, $\{x_1\in A_{L,\sqrt{\delta}},x_2\in A_{C,\sqrt{\delta}}\}$ and $\{x_1\in A_{C,\sqrt{\delta}},x_2\in A_{R,\sqrt{\delta}}\}$ can be proved in the same way. Details are omitted.

C.2. Proof of Lemma 3 for the case $k = 2$. Preliminary results

Similarly, Laurent and Massart (2000) gives tail bound of χ^2 distribution,

$$\mathbb{P}[\chi_n^2 \geq 5n] \leq \exp(-n).$$

By the tail bound above, it's straightforward to show that

Suppose $\mathbf{x}$ follows χ^2 distribution with degrees of freedom k and non-central parameter λ

$$P(\mathbf{x} \geq x) \leq \frac{\sqrt{\pi}}{2e}\Phi(\sqrt{x})I_{\frac{k}{2}}(1)M_{k-1}, \tag{65}$$

where I_v is a modified Bessel function of the first kind, $M_{k-1} = E(y^{k-1})$ and y is a Gaussian $(\mu, 1)$ random variable truncated on $(\sqrt{x}, \infty)$. So we can claim that with probability $1-\rho$, the term $\frac{1}{n}\sum_{i=1}^n y_i^2$ is bounded.

Follow the notation of Scornet et al. (2015), let $d_1' = (1, x_1')$ and $d_2' = (2, x_2'$ be such that $|x_1 - x_1'| \leq \delta$ and $|x_2 - x_2'| \leq \delta$.

There exist a constant $C_\rho > 0$ and N_1 such that, with probability $1-\rho$, for all $n > N_1$,

$$\max_{1\leq i\leq n} |\epsilon_i| \leq C_\rho\sqrt{\log(n)} \tag{66}$$

and

$$\max_{1\leq i\leq n} |\epsilon_i^2| \leq C_\rho^2 \log(n). \tag{67}$$

Fix $\rho > 0$, there exist N_2 such that, with probability $1-\rho$, for all $n > N_2$ and all $A_n = [a_n^{(1)}, b_n^{(1)}]\times[a_n^{(2)}, b_n^{(2)}] \subset [0,1]^2$ satisfying $\mathrm{N}_n(A_n) > \sqrt{n}$,

$$\left|\frac{1}{\mathrm{N}_n(A_n)}\sum_{i:\mathbf{x}_i\in A_n}\epsilon_i\right| \leq \alpha \tag{68}$$

and

$$\frac{1}{\mathrm{N}_n(A_n)}\sum_{i:\mathbf{x}_i\in A_n}\epsilon_i^2 \leq \tilde{\sigma}^2. \tag{69}$$

Furthermore, it's easy to verify

$$\left|\frac{1}{\mathrm{N}_n(A_n)}\sum_{i:\mathbf{x}_i\in A_n} y_i\right| \leq ||m||_\infty + \alpha \tag{70}$$

and

$$\left|\frac{1}{\mathrm{N}_n(A_n)}\sum_{i:\mathbf{x}_i\in A_n} y_i^2\right| \leq ||m||_\infty^2 + \tilde{\sigma}^2 + 2\alpha||m||_\infty. \tag{71}$$

Similar to the $k = 1$ case, we denote partition of space as

$$\begin{cases} A_{R,1} = [x_1, 1] \times [0,1]^{p-1} \\ A_{B,2} = [x_1, 1] \times [0, x_2] \times [0,1]^{p-1} \\ A_{H,2} = [x_1, 1] \times [x_2, 1] \times [0,1]^{p-1} \\ A'_{B,2} = [x'_1, 1] \times [0, x'_2] \times [0,1]^{p-1} \\ A'_{H,2} = [x'_1, 1] \times [x'_2, 1] \times [0,1]^{p-1}. \end{cases} \tag{72}$$

See Figure 4 for an illustration of notations.

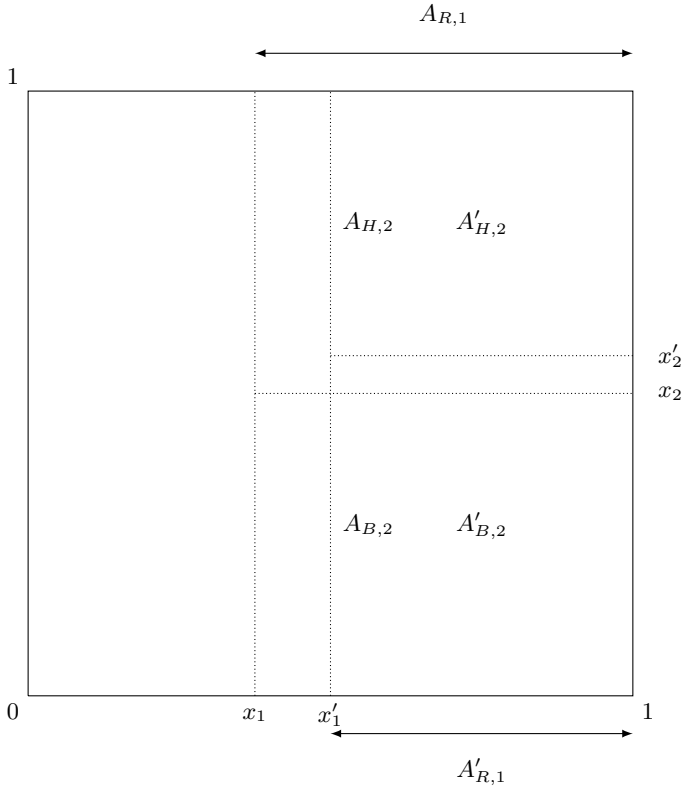


Fig 4: Illustration of notations for $k = 2$.

Let $d_1 = (1, x_1)$, $d_2 = (2, x_2)$, $d'_1 = (1, x'_1)$ and $d'_2 = (2, x'_2)$ be four cutpoints and $|x_1 - x'_1| < \delta$, $|x_2 - x'_2| < \delta$, then

$$L_n(d_1, d_2) - L_n(d'_1, d'_2) = L_n(d_1, d_2) - L_n(d'_1, d_2) + L_n(d'_1, d_2) - L_n(d'_1, d'_2). \tag{73}$$

$$
\begin{aligned}
&L_n(d_1,d_2)-L_n(d_1',d_2)\\
=&\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{B,2})\right)}\frac{1}{\mathrm{N}_n(A_{R,1})}\left(\mathrm{N}_n(A_{B,2})\sum_{i:\mathbf{x}_i^{(2)}\le x_2} y_i^2\mathbb{1}_{\mathbf{x}_i^{(1)}>x_1}\right.\\
&\qquad\left.-(\mathrm{N}_n(A_{B,2})-1)\sum_{i:\mathbf{x}_i^{(2)}\le x_2}(y_i-\bar{y}_{A_{B,2}})^2\mathbb{1}_{\mathbf{x}_i^{(1)}>x_1}\right)\\
&+\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{H,2})\right)}\frac{1}{\mathrm{N}_n(A_{R,1})}\left(\mathrm{N}_n(A_{H,2})\sum_{i:\mathbf{x}_i^{(2)}> x_2} y_i^2\mathbb{1}_{\mathbf{x}_i^{(1)}>x_1}\right.\\
&\qquad\left.-(\mathrm{N}_n(A_{H,2})-1)\sum_{i:\mathbf{x}_i^{(2)}> x_2}(y_i-\bar{y}_{A_{H,2}})^2\mathbb{1}_{\mathbf{x}_i^{(1)}>x_1}\right)\\
&-\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{B,2})\right)}\frac{1}{\mathrm{N}_n(A'_{R,1})}\left(\mathrm{N}_n(A'_{B,2})\sum_{i:\mathbf{x}_i^{(2)}\le x_2} y_i^2\mathbb{1}_{\mathbf{x}_i^{(1)}>x_1'}\right.\\
&\qquad\left.-(\mathrm{N}_n(A'_{B,2})-1)\sum_{i:\mathbf{x}_i^{(2)}\le x_2}(y_i-\bar{y}_{A'_{B,2}})^2\mathbb{1}_{\mathbf{x}_i^{(1)}>x_1'}\right)\\
&-\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)}\frac{1}{\mathrm{N}_n(A'_{R,1})}\left(\mathrm{N}_n(A'_{H,2})\sum_{i:\mathbf{x}_i^{(2)}> x_2} y_i^2\mathbb{1}_{\mathbf{x}_i^{(1)}>x_1'}\right.\\
&\qquad\left.-(\mathrm{N}_n(A'_{H,2})-1)\sum_{i:\mathbf{x}_i^{(2)}> x_2'}(y_i-\bar{y}_{A'_{H,2}})^2\mathbb{1}_{\mathbf{x}_i^{(1)}>x_1'}\right)\\
&+\frac{\gamma_{x_1,x_2}}{\mathrm{N}_n(A_{R,1})}-\frac{\gamma_{x_1',x_2}}{\mathrm{N}_n(A'_{R,1})}\\
&=A_1+B_1
\end{aligned}
\tag{74}
$$

$$
\begin{aligned}
A_1 =& \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{H,2})\right)} \frac{1}{\mathrm{N}_n(A_{R,1})} \Bigg(\mathrm{N}_n(A_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x_1} \\
& \qquad -(\mathrm{N}_n(A_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i-\bar{y}_{A_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x_1} \Bigg) \\
& - \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{1}{\mathrm{N}_n(A'_{R,1})} \Bigg(\mathrm{N}_n(A'_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \\
& \qquad -(\mathrm{N}_n(A'_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x'_2} (y_i-\bar{y}_{A'_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \Bigg) \\
=& A_{1,1}+A_{1,2}+A_{1,3}
\end{aligned} \tag{75}
$$

$$
\begin{aligned}
A_{1,1} =& \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{H,2})\right)} \frac{1}{\mathrm{N}_n(A_{R,1})} \Bigg(\mathrm{N}_n(A_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \\
& \qquad -(\mathrm{N}_n(A_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i-\bar{y}_{A_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \Bigg) \\
& - \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{1}{\mathrm{N}_n(A_{R,1})} \Bigg(\mathrm{N}_n(A'_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \\
& \qquad -(\mathrm{N}_n(A'_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i-\bar{y}_{A'_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \Bigg)
\end{aligned} \tag{76}
$$

$$
\begin{aligned}
A_{1,2} =& \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{1}{\mathrm{N}_n(A_{R,1})} \Bigg(\mathrm{N}_n(A'_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \\
& \qquad -(\mathrm{N}_n(A'_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i-\bar{y}_{A'_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \Bigg) \\
& - \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{1}{\mathrm{N}_n(A'_{R,1})} \Bigg(\mathrm{N}_n(A'_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \\
& \qquad -(\mathrm{N}_n(A'_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x'_2} (y_i-\bar{y}_{A'_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \Bigg)
\end{aligned} \tag{77}
$$

$$A_{1,3} = \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{H,2})\right)} \frac{1}{\mathrm{N}_n(A_{R,1})} \left(\mathrm{N}_n(A_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)} \in [x_1, x_1']} \right. \tag{78}$$
$$\left. -(\mathrm{N}_n(A_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i - \bar{y}_{A_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)} \in [x_1, x_1']} \right)$$

$$\begin{aligned} A_{1,1} =& \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{H,2})\right)} \frac{1}{\mathrm{N}_n(A_{R,1})} \left(\mathrm{N}_n(A_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)} > x_1'} \right. \\ & \left. -(\mathrm{N}_n(A_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i - \bar{y}_{A_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)} > x_1'} \right) \\ & - \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{1}{\mathrm{N}_n(A_{R,1})} \left(\mathrm{N}_n(A'_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)} > x_1'} \right. \\ & \left. -(\mathrm{N}_n(A'_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i - \bar{y}_{A'_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)} > x_1'} \right) \end{aligned} \tag{79}$$

$$\begin{aligned} =& \left(\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{H,2})\right)} \frac{\mathrm{N}_n(A_{H,2})}{\mathrm{N}_n(A_{R,1})} - \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{\mathrm{N}_n(A'_{H,2})}{\mathrm{N}_n(A_{R,1})} \right) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)} > x_1'} \\ & + \left(\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{(\mathrm{N}_n(A'_{H,2})-1)}{\mathrm{N}_n(A_{R,1})} \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i - \bar{y}_{A'_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)} > x_1'} \right. \\ & \left. - \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{H,2})\right)} \frac{(\mathrm{N}_n(A_{H,2})-1)}{\mathrm{N}_n(A_{R,1})} \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i - \bar{y}_{A_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)} > x_1'} \right) \end{aligned}$$

which goes to zero using the same argument as $k = 1$ case.

$$
\begin{aligned}
A_{1,2} =& \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{1}{\mathrm{N}_n(A_{R,1})} \left(\mathrm{N}_n(A'_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \right. \\
& \left. -(\mathrm{N}_n(A'_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i - \bar{y}_{A'_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \right) \\
& - \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{1}{\mathrm{N}_n(A'_{R,1})} \left(\mathrm{N}_n(A'_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \right. \\
& \left. -(\mathrm{N}_n(A'_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x'_2} (y_i - \bar{y}_{A'_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \right) \\
=& \left(\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{1}{\mathrm{N}_n(A_{R,1})} - \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{1}{\mathrm{N}_n(A'_{R,1})} \right) \\
& \times \left(\mathrm{N}_n(A'_{H,2}) \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} - (\mathrm{N}_n(A'_{H,2})-1) \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i - \bar{y}_{A'_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \right)
\end{aligned} \tag{80}
$$

$$
\begin{aligned}
|A_{1,2}| \leq & \left| \frac{\tau \mathrm{N}_n(A'_{H,2})}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{\mathrm{N}_n(A'_{H,2})}{\mathrm{N}_n(A_{R,1})} - \frac{\tau \mathrm{N}_n(A'_{H,2})}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A'_{H,2})\right)} \frac{\mathrm{N}_n(A'_{H,2})}{\mathrm{N}_n(A'_{R,1})} \right| \\
& \times \left(\left| \frac{1}{\mathrm{N}_n(A'_{H,2})} \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \right| + \left| \frac{1}{\mathrm{N}_n(A'_{H,2})} \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i - \bar{y}_{A'_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)}>x'_1} \right| \right)
\end{aligned} \tag{81}
$$

Same as before, the second term is bounded and

$$
|A_{1,2}| \leq M \left| \frac{\mathrm{N}_n(A'_{H,2})}{\mathrm{N}_n(A_{R,1})} - \frac{\mathrm{N}_n(A'_{H,2})}{\mathrm{N}_n(A'_{R,1})} \right| \to 0 \tag{82}
$$

$$
\begin{aligned}
|A_{1,3}| \leq & \left| \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{H,2})\right)} \frac{\mathrm{N}_n(A_{H,2})}{\mathrm{N}_n(A_{R,1})} \mathrm{N}_n\left(\left\{ \mathbf{x}_i^{(1)} \in [x_1, x'_1] \right\} \times \left\{ \mathbf{x}_i^{(2)} > x_2 \right\} \right) \right| \\
& \times \left| \frac{1}{\mathrm{N}_n\left(\left\{ \mathbf{x}_i^{(1)} \in [x_1, x'_1] \right\} \times \left\{ \mathbf{x}_i^{(2)} > x_2 \right\} \right)} \sum_{i:\mathbf{x}_i^{(2)}>x_2} y_i^2 \mathbb{1}_{\mathbf{x}_i^{(1)} \in [x_1, x'_1]} \right| \\
& + \left| \frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{H,2})\right)} \frac{\mathrm{N}_n(A_{H,2})-1}{\mathrm{N}_n(A_{R,1})} \mathrm{N}_n\left(\left\{ \mathbf{x}_i^{(1)} \in [x_1, x'_1] \right\} \times \left\{ \mathbf{x}_i^{(2)} > x_2 \right\} \right) \right| \\
& \times \left| \frac{1}{\mathrm{N}_n\left(\left\{ \mathbf{x}_i^{(1)} \in [x_1, x'_1] \right\} \times \left\{ \mathbf{x}_i^{(2)} > x_2 \right\} \right)} \sum_{i:\mathbf{x}_i^{(2)}>x_2} (y_i - \bar{y}_{A_{H,2}})^2 \mathbb{1}_{\mathbf{x}_i^{(1)} \in [x_1, x'_1]} \right| \\
= & A_{1,3,1} + A_{1,3,2}.
\end{aligned} \tag{83}
$$

Note that $\frac{\tau \mathrm{N}_n(A_{H,2})}{\sigma^2(\sigma^2+\tau \mathrm{N}_n(A_{H,2}))}$ is bounded by a constant M as n is large,

$$\left|\frac{\tau}{\sigma^2\left(\sigma^2+\tau \mathrm{N}_n(A_{H,2})\right)}\frac{\mathrm{N}_n(A_{H,2})}{\mathrm{N}_n(A_{R,1})}\mathrm{N}_n\left(\left\{\mathbf{x}_i^{(1)}\in[x_1,x_1']\right\}\times\left\{\mathbf{x}_i^{(2)}>x_2\right\}\right)\right| \leq M\frac{\delta^2+\delta}{\delta^2-\sqrt{\delta}}\to 0. \tag{84}$$

So we have $A_{1,3,1}\to 0$ if n is large and δ is small.

If $\mathrm{N}_n\left(\left\{\mathbf{x}_i^{(1)}\in[x_1,x_1']\right\}\times\left\{\mathbf{x}_i^{(2)}>x_2\right\}\right)<\sqrt{n}$,

$$\left|\frac{1}{\mathrm{N}_n\left(\left\{\mathbf{x}_i^{(1)}\in[x_1,x_1']\right\}\times\left\{\mathbf{x}_i^{(2)}>x_2\right\}\right)}\sum_{l:\mathbf{x}_i^{(2)}>x_2}(y_i-\bar{y}_{A_{H,2}})^2\mathbb{1}_{\mathbf{x}_i^{(1)}\in[x_1,x_1']}\right| \leq \frac{C_\rho^2\log(n)}{\sqrt{n}}. \tag{85}$$

If $\mathrm{N}_n\left(\left\{\mathbf{x}_i^{(1)}\in[x_1,x_1']\right\}\times\left\{\mathbf{x}_i^{(2)}>x_2\right\}\right)>\sqrt{n}$, note that $|1-x_1|\geq\xi$, $\mathrm{N}_n(A_{R,1})>\mathrm{N}_n(\xi)>(\xi-\delta^2)n$, $\mathrm{N}_n\left(\left\{\mathbf{x}_i^{(1)}\in[x_1,x_1']\right\}\times\left\{\mathbf{x}_i^{(2)}>x_2\right\}\right)\leq\mathrm{N}_n\left(\left\{\mathbf{x}_i^{(1)}\in[x_1,x_1']\right\}\right)\leq(\delta+\delta^2)n$. As a result

$$\left|\frac{\mathrm{N}_n\left(\left\{\mathbf{x}_i^{(1)}\in[x_1,x_1']\right\}\times\left\{\mathbf{x}_i^{(2)}>x_2\right\}\right)}{\mathrm{N}_n(A_{R,1})}\right| \leq \frac{\delta-\delta^2}{\xi+\delta^2}\leq\frac{\delta}{\xi} \tag{86}$$

$$\begin{aligned}&\left|\frac{\mathrm{N}_n\left(\left\{\mathbf{x}_i^{(1)}\in[x_1,x_1']\right\}\times\left\{\mathbf{x}_i^{(2)}>x_2\right\}\right)}{\mathrm{N}_n(A_{R,1})}\right| \\ &\quad\times\left|\frac{1}{\mathrm{N}_n\left(\left\{\mathbf{x}_i^{(1)}\in[x_1,x_1']\right\}\times\left\{\mathbf{x}_i^{(2)}>x_2\right\}\right)}\sum_{l:\mathbf{x}_i^{(2)}>x_2}(y_i-\bar{y}_{A_{H,2}})^2\mathbb{1}_{\mathbf{x}_i^{(1)}\in[x_1,x_1']}\right| \\ &\leq\frac{\delta}{\xi}\left(3(||m||_\infty+\alpha)^2+||m||_\infty^2+\tilde{\sigma}^2+2||m||_\infty^2\alpha\right).\end{aligned} \tag{87}$$

Therefore $A_{1,3,2}\to 0$. Collecting all bounds, we conclude that $A_1\to 0$. It is straightforward to prove with similar arguments that $B_1\to 0$ and $L_n(d_1,d_2)-L_n(d_1',d_2')\to 0$.