Local Gaussian Process Extrapolation for BART Models with Applications to Causal Inference

Meijia Wang\textsuperscript{a} \& Jingyu He\textsuperscript{b}, and P. Richard Hahn\textsuperscript{a}

\textsuperscript{a}Arizona State University, Tempe, AZ; \textsuperscript{b}City University of Hong Kong, Kowloon, Hong Kong

ABSTRACT
Bayesian additive regression trees (BART) is a semi-parametric regression model offering state-of-the-art performance on out-of-sample prediction. Despite this success, standard implementations of BART typically suffer from inaccurate prediction and overly narrow prediction intervals at points outside the range of the training data. This article proposes a novel extrapolation strategy that grafts Gaussian processes to the leaf nodes in BART for predicting points outside the range of the observed data. The new method is compared to standard BART implementations and recent frequentist resampling-based methods for predictive inference. We apply the new approach to a challenging problem from causal inference, wherein for some regions of predictor space, only treated or untreated units are observed (but not both). In simulation studies, the new approach boasts superior performance compared to popular alternatives, such as Jackknife+. Supplementary materials for this article are available online.

1. Introduction

Tree-based supervised learning algorithms, including Classification and Regression Tree (CART) (Breiman et al. 1984), Random Forests (Breiman 2001), and XGBoost (Chen and Guestrin 2016), are widely used in practice for their ability to learn complex nonlinear functions efficiently. Among model-based regression tree methods, Bayesian Additive Regression Trees (BART) (Chipman et al. 2010) stands out as the most popular choice. Empirical evidence has shown that BART can deliver accurate out-of-sample predictions (without covariate shift), and its Bayesian uncertainty intervals often outperform alternative methods in terms of frequentist coverage (see Chipman et al. 2010; Kapelner and Bleich 2013). XBART (He and Hahn 2021) is a stochastic tree ensemble method that can approximate BART models in a fraction of the run-time. Throughout the article, we will refer to BART models but will use the XBART fitting algorithm.

Although tree-based methods are known for providing accurate out-of-sample predictions, their ability to extrapolate is inherently limited by their piece-wise constant structure. Trees divide the covariate space into disjoint rectangular regions (leaf nodes) based on observed covariates, and then estimate a simple model (typically a constant) using training data within each region. As a result, when it comes to predicting testing data that lie outside the observed range of covariates in training, a naive tree regression model would simply extrapolate the constant value from the leaf node where the point falls. Figure 1 illustrates the potential problem of extrapolation for the traditional tree models with constant leaf parameters in one-dimensional covariate space. In such cases, if the testing data falls outside the range of the training data, tree models must rely on extrapolating using the constant leaf parameter from the corresponding leaf node, regardless of how far away the testing observations are from the nearest training point.

In addition to the issue of poor point predictions, a flawed extrapolation model also undermines the accuracy of posterior prediction intervals. Ideally, in regions far from the observed data, a BART model would rely on the prior distribution for predictions. However, standard implementations of BART never sample cutpoints beyond the training range, which can significantly underestimate the uncertainty associated with the unknown function in extrapolation regions. Consequently, the predictive intervals generated by standard tree-based methods often have poor coverage. While regression tree methods can be used in conjunction with resampling approaches to construct improved prediction intervals (Efron and Stein 1981; Papadopoulos 2008; Vovk 2015; Barber et al. 2021), such strategies do not solve the intrinsically poor extrapolation of regression trees.

This article presents a novel approach to help tree methods extrapolate the exterior points and construct prediction intervals. The new method (XBART-GP) fits a Gaussian process (GP) at each leaf node of a BART tree and predicts exterior test points in the leaf by the associated GP model. Our approach differs from a treed Gaussian process (Gramacy and Lee 2008) in that the local GP extrapolation does not affect the initial BART model fit. In other words, the model of Gramacy and Lee (2008) is a “treed” GP where the tree grows based on the likelihood of Gaussian process in the leaf nodes, our approach can be described as a “GPed” tree. The local GP extrapolation defines a novel posterior predictive distribution in terms of BART posterior samples and a GP model; it is not a new likelihood for training or
inferring the trees themselves. Instead, the BART-inferred trees are used as inputs to the novel posterior predictive specification (see Section 3). Our proposed method shares a similar structure with the targeted smoothing BART (tsBART) model (Starling et al. 2018, 2021), which fits a Gaussian process on a single targeted covariate in each tree leaf to introduce smoothness. In contrast, our approach fits the Gaussian process on a selection of covariates to extrapolate prediction.

In simulation studies, we compare our approach with recently proposed alternatives such as Jackknife+ (Barber et al. 2021). Our results illustrate that XBART-GP achieves nearly the desired coverage on out-of-range data without substantially inflating the interval length.

Furthermore, we demonstrate the application of our new extrapolation method in the context of causal inference. In scenarios where only treated (or untreated) units are observed within a specific region of covariate space, it violates the positivity condition required for identifying causal effects from observational data. Inferring the counterfactual potential outcomes requires extrapolating the untreated (treated) response surface from overlap regions where both groups are observed. Previous work on positivity violation includes D’Amour et al. (2021), which focuses on high-dimensional covariate spaces; Nethery, Mealli, and Dominici (2019), which proposes a spline function approximated by a sum of regression trees, that is

$$f(x) = \sum_{l=1}^{L} g(x, T_l, \mu_l),$$

where $T_l$ represents a tree structure. Tree $T_l$ is essentially a set of split rules partitioning the covariate space into a set of subspaces (say $\{A_1, \ldots, A_B\}$), and $\mu_l = (\mu_{l1}, \ldots, \mu_{lB})$ is a vector of leaf parameters associated with the leaf nodes in tree $T_l$.

Figure 2 illustrates a binary regression tree. The tree consists of a set of internal decision nodes and a set of terminal nodes (also called leaf nodes) partitioning the covariate space to $B$ leaf nodes in total. The tree function $g(x, T_l, \mu_l)$ is a step function due to the constant leaf parameter. It assigns point $x$ with the leaf parameter of the leaf node that it falls into in tree $T_l$.

Trees are known to be prone to overfitting due to their inherent high flexibility. Therefore, proper regularization is necessary to achieve good out-of-the-sample performance. BART addresses this issue by imposing a stringent prior on the tree structure that strongly favors weak or small trees. The tree prior $p(T_l)$ specifies the probability for a node to split into two child nodes at depth $d$ to be

$$\alpha(1 + d)^{-\beta}, \quad \alpha \in (0, 1), \quad \beta \in [0, \infty),$$

2.2. BART Model and XBART Algorithm

The BART model (Chipman et al. 2010) assumes that the unknown function $f(x)$ in the regression model (1) can be approximated by a sum of regression trees, that is

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

where we assume the residual term $\epsilon$ is a Gaussian noise term with mean zero and variance $\sigma^2$.

Let $C_{n,\alpha}(x)$ denote the prediction interval for a test point $x$ with confidence level $1 - \alpha$ given $n$ training data. Notations $q_{\alpha/2}[y_i]$ and $d_{\alpha/2}[y_i]$ are the quantile functions for the $[\alpha/2(n + 1)]$th smallest value and the $[(1 - \alpha/2)(n + 1)]$th smallest value among a set of values $\{y_1, \ldots, y_n\}$, respectively.

2.2. BART Model and XBART Algorithm

The BART model (Chipman et al. 2010) assumes that the unknown function $f(x)$ in the regression model (1) can be approximated by a sum of regression trees, that is

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

where we assume the residual term $\epsilon$ is a Gaussian noise term with mean zero and variance $\sigma^2$.

Let $C_{n,\alpha}(x)$ denote the prediction interval for a test point $x$ with confidence level $1 - \alpha$ given $n$ training data. Notations $q_{\alpha/2}[y_i]$ and $d_{\alpha/2}[y_i]$ are the quantile functions for the $[\alpha/2(n + 1)]$th smallest value and the $[(1 - \alpha/2)(n + 1)]$th smallest value among a set of values $\{y_1, \ldots, y_n\}$, respectively.

2.2. BART Model and XBART Algorithm

The BART model (Chipman et al. 2010) assumes that the unknown function $f(x)$ in the regression model (1) can be approximated by a sum of regression trees, that is

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

where we assume the residual term $\epsilon$ is a Gaussian noise term with mean zero and variance $\sigma^2$.

Let $C_{n,\alpha}(x)$ denote the prediction interval for a test point $x$ with confidence level $1 - \alpha$ given $n$ training data. Notations $q_{\alpha/2}[y_i]$ and $d_{\alpha/2}[y_i]$ are the quantile functions for the $[\alpha/2(n + 1)]$th smallest value and the $[(1 - \alpha/2)(n + 1)]$th smallest value among a set of values $\{y_1, \ldots, y_n\}$, respectively.
which decreases exponentially as the tree grows deeper, implying strong regularization on the size of the tree. Chipman et al. (2010) suggest choosing \( \alpha = 0.95 \) and \( \beta = 2 \). The prior of each leaf parameter \( p(\mu_b) \) is assumed to be independent normal with variance \( \tau \), that is, \( \mu_b \sim N(0, \tau) \). The prior of the residual variance \( \sigma^2 \) is set to be inverse-Gamma \((a, b)\).

The ensemble of trees is fitted by Bayesian backfitting and MCMC sampling schemes. Let \( T_{-l} \) denotes the set of all trees except \( T_l \) and similarly define \( \mu_{-l} \). Note that the conditional posterior \( p(T_l | \mu_l, T_{-l}, \mu_{-l}, \sigma^2, y) \) depends on other trees and parameters only through the residuals

\[
\rho_l := y - \sum_{h \neq l} g(x_h, \hat{T}_h, \hat{\mu}_h) = g(x, T_l, \mu_l). \tag{4}
\]

Furthermore, one can integrate out the conjugate prior of \( \mu_l \) and derive the posterior of a tree \( T_l \) in closed form, that is,

\[
p(T_l | \rho_l, \sigma^2) \propto p(T_l) \int p(\rho_l | \mu_l, T_l, \sigma^2) p(\mu_l | T_l, \sigma^2) d\mu_l. \tag{5}
\]

This allows the conditional posterior \( T_l | \rho_l, \sigma^2 \) and \( \mu_l | T_l, \rho_l, \sigma^2 \) to be drawn separately and sequentially for \( l = 1, \ldots, L \). After all the trees are grown (we call finishing a sweep), the residual variance \( \sigma^2 \) is then sampled from the conditional posterior \( \sigma^2 | T_1, \ldots, T_L, \mu_1, \ldots, \mu_L, y \).

The original BART model (Chipman et al. 2010) draws trees from the posterior using a random walk Metropolis-Hastings (MH-MCMC) algorithm. In each iteration, the algorithm randomly proposes a single growing or pruning procedure to each tree and accepts or rejects it according to the MH ratios. However, due to the small modification made by the proposal in each iteration, the MH-MCMC algorithm requires a large number of iterations to explore the posterior model space. The experiments in (Chipman et al. 2010) typically use 200 burn-in steps and 1000 iterations with 200 trees. Furthermore, the algorithm does not scale well with large sizes of data observations or covariates.

XBART (He, Yalov, and Hahn 2019; He and Hahn 2021), short for “accelerated Bayesian additive regression trees,” is a stochastic tree ensemble method inspired by BART. The fundamental idea of XBART is to avoid the expensive MH-MCMC computation by growing completely new trees per iteration. At each node, the model considers a similar set of cutoff points as BART and sample split criteria proportional to the marginal likelihood. This approach is able to discover good-fitting deep trees more efficiently than BART MH-MCMC and scales better to larger datasets. Our approach is designed for both the framework of XBART and BART, while we will demonstrate the performance using XBART for computational efficiency.

To obtain the posterior prediction interval from XBART, we take the sampled trees as draws from a standard Bayesian Monte Carlo algorithm. Suppose the algorithm draws \( S \) sweeps with \( S_0 \) burn-in iterations and \( L \) trees per forest. The predicted posterior mean and variance at iteration \( s \) are defined as \( \hat{f}_s(x) = \sum_{l=1}^L g(x, \hat{T}_l^s, \hat{\mu}_l^s) \) and \( \hat{\sigma}^2_s \), respectively. The final prediction of target value \( \hat{y}_i \) is sampled from \( N(\hat{f}(x_i), \hat{\sigma}^2) \). Furthermore, the prediction interval of target level \( 1 - \alpha \) is given by the quantiles of the predictions as follows

\[
\begin{align*}
\tilde{y}_{SA/2}^{XBART} &= \left[ \frac{\hat{y}_i - \alpha/2}{\hat{\sigma}^2}, \frac{\hat{y}_i + \alpha/2}{\hat{\sigma}^2} \right].
\end{align*}
\tag{6}
\]

Note that this definition of posterior prediction interval relies on the prediction of every single tree with a constant leaf parameter. Consequently, it may still encounter challenges when attempting to make predictions outside the range of the training data, or extrapolation.

2.3. Gaussian Process Regression

This section reviews Gaussian process regression, a critical ingredient of the proposed XBART-GP extrapolation method. Gaussian process regression is a nonparametric kernel-based Bayesian probabilistic model. It can interpolate and extrapolate by defining the model’s behavior across its entire domain according to covariance. Essentially, the specified covariance function characterizes the linear dependence of the response at pairs of points as a function of some measure of distance between those points in covariate space. For a highly readable textbook treatment of the Gaussian process regression, see Rasmussen (2003).

In Gaussian process regression, we assume that \( f(x) \in \mathbb{R}^p \) represent a set of random variables that follows the Gaussian process, characterized by a mean function \( \mu(x) \) and a covariance function \( k(x, x') \). Let \((X, y)\) denote a set of training data, and \( X^* \) denote the input of testing data. Following the standard regression setting in (1), the joint distribution of function values \( f(X^*) \) and \( y \) can be expressed as follows:

\[
(f(X^*), y) \sim N\left(\left(\mu(X^*), (\Sigma_{XX}, \Sigma_{X^*}) \right), (\Sigma_{XX} + \sigma^2 I)^{-1}\right) \tag{7}
\]

where \( \mu(\cdot) \) is a regression function, and the covariance matrices are calculated by pre-specified kernel function \( k(x, x^*) \).

The prediction of the outcome \( f(X^*) \) can be drawn from the conditional distribution \( f(X^*) \sim N(\mu_{X^*}(X), \Sigma_{XX}) \) with conditional mean and covariance:

\[
\mu_{X^*} = \mu(X) + \Sigma_{X^*X} (\Sigma_{XX} + \sigma^2 I)^{-1} (y - \mu(X)) \tag{8}
\]

\[
\Sigma_{X^*X} = \Sigma_{X^*X} - \Sigma_{X^*} (\Sigma_{XX} + \sigma^2 I)^{-1} \Sigma_{XX}. \tag{9}
\]

3. Local GP Extrapolation for BART

Traditional BART prediction intervals in (6) provide good coverage with tight intervals on many data generating processes (He and Hahn 2021) when the training data and the test data are exchangeable. However, predictions are less reliable when the training data and test datasets have substantially different supports.

The intuition behind our method is to use the tried-and-true BART predictions (and intervals) for prediction points within the range of the training data but to incorporate Gaussian process extrapolation for points well outside this support. Therefore, the first step in describing the new method is to formally define the concepts of exterior points in the context of regression trees.

Specifically, we will define extrapolation points locally in terms of the regression trees: a test point \( x' \) is an exterior point if and only if it is outside the range of the training data of the leaf node it falls in. The basic strategy of the new method is to use the leaf-specific training data to extrapolate such exterior points
using a Gaussian process model, which we call it local Gaussian process. Combining these local GP predictions across trees (and across posterior samples) constitutes the main technical content of the new method.

Before getting into those details, a big-picture explanation may be helpful. Typically, a Bayesian posterior predictive distribution is

\[
\tilde{y}(\theta | y_{1:n}, X_{1:n}) = \int \tilde{y}(\theta | \theta) \pi(\theta | y_{1:n}, X_{1:n}) d\theta,
\]

which conveys the idea that future and past data are conditionally independent given the model parameters. Here, we explicitly deviate from this formulation, but only for those future points that are exterior (defined as a function of the model parameters). For such a point, our posterior predictive is defined as

\[
\tilde{y}(\theta | y_{1:n}, X_{1:n}) = \int \tilde{y}(\theta | y_{1:n}, X_{1:n}) \pi(\theta | y_{1:n}, X_{1:n}) d\theta
\]

where the predictive distribution explicitly involves both the training data and model parameters (trees and leaves means in the case of BART). In this sense, the proposed approach does "use the data twice" but not in the construction of the posterior. Rather, our procedure is a way of combining the orthodox BART posterior with an intentionally and explicitly distinct posterior predictive model for extrapolation points. Although this is an uncommon approach that violates so-called "diachronic coherence" (Skyrms 2006), it is perfectly valid as it merely amounts to specifying a user-defined predictive distribution that takes BART posterior samples as inputs. From this perspective, describing the method amounts to providing a definition of the predictive kernel, \(\tilde{y}(\theta | y_{1:n}, X_{1:n})\). Specifically, this will be a Gaussian process with a covariance kernel defined in terms of the trees of the BART model.

3.1. Notation

Consider a fitted BART forest \(\hat{T}_l | 1 \leq l \leq L\) with \(L\) trees. Let \(\{A_l, \ldots, A_{l+B_l}\}\) denote the covariate space partitioned by the \(l\)th tree \(\hat{T}_l\) with \(B_l\) leaf nodes, and \(\hat{\sigma}^2\) for the estimated residual variance. For the \(l\)th tree, suppose a test point \(x\) belongs to covariate space \(A_{lB}\) and falls in the corresponding leaf node \(b\). We denote the training data that falls in this leaf node and its residuals at the current tree (defined by \(4\)) as \(X^e\) and \(r^e\).

Similarly, the exterior testing data falling in leaf node \(b\) and its to-be-predicted residuals can be denoted as \(X^{te}\) and \(r^{te}\), respectively.

It is important to note that the tree partitioned covariate space \(A_{lB}\) can extend to infinity if they are at the boundary. Specifically, the range of subset of training data \(X^e\) falling in the \(b\)th leaf node of the \(l\)-tree forms a \(p\)-dimensional hypercube \(B_{lB}\), which is a subset of \(A_{lB}\). If the testing point \(x\) falls within the range of training data, that is, \(x \in B_{lB}\), it is considered an interior point, and its prediction follows the standard XBART model. However, if \(x \notin A_{lB} \setminus B_{lB}\), it is identified as an exterior point, requiring extrapolation for its prediction. Importantly, this definition is made locally on a per-tree and per-iteration basis. It is possible for a testing data point to be classified as exterior for some of the trees and interior for others within the XBART forest.

3.2. Defining the GP Model

To extrapolate the prediction for exterior points, we assume that the exterior testing data \(X^{te}\) and training data \(X^e\) in the same leaf node follow a Gaussian process. The joint distribution of the training and testing data in a leaf node can be described as follows

\[
\begin{pmatrix}
\tilde{y}^{te}
\end{pmatrix}
= N
\begin{pmatrix}
\mu, \\
\Sigma_X^{te}X^e + \Sigma_X^{te}X^{te} + \frac{\Sigma_X^{te}X^{te}}{\tau^2}I
\end{pmatrix}
\]

where \(\mu\) is the leaf parameter, \(J\) is a column vector of ones, and \(I\) is an identity matrix. By definition, the variance of the response \(y\) is assumed to be \(\sigma^2\). Here we assume that the residuals in each tree contribute equally to the total variance.

To reflect the relative distance among covariates, we define the covariance function of the Gaussian process as the squared exponential kernel

\[
k(x, x^e) = \tau \exp \left( -\sum_{i=1}^p \frac{(x_i - x_i^e)^2}{2\hat{\sigma}^2} \right)
\]

where \(\hat{\sigma}_i\) is the range of the \(i\)th variable \(x_i\) in a leaf node, \(\theta\) controls the smoothness and \(\tau\) determines the scale of the function. \(x = (x_1, \ldots, x_p)\) and \(x^e = (x_1^e, \ldots, x_p^e)\) are the vectors of two data points. In our experiments, we use \(\theta = \sqrt{10}\) and set \(\tau\) to be the variance of observed responses divided by the number of trees.

The conditional distribution of the residuals \(r^{te}\) is given by

\[
\begin{align*}
\mu_{X^{te}} &= \mu + \Sigma_{X^{te}X^e} \left( \Sigma_{X^eX^e} + \frac{1}{L} \hat{\sigma}^2 I \right)^{-1} (r^{te} - \mu) \\
\Sigma_{X^{te}X^{te}} &= \Sigma_{X^eX^e} - \Sigma_{X^eX^{te}} \left( \Sigma_{X^eX^e} + \frac{1}{L} \hat{\sigma}^2 I \right)^{-1} \Sigma_{X^eX^{te}}.
\end{align*}
\]

The prediction for the residuals of the exterior testing data associated with this leaf node is drawn from a conditional normal distribution \(r^{te} \sim N(\mu_{X^{te}}, \Sigma_{X^{te}X^{te}})\) while the prediction for the interior points uses the fixed leaf parameter of the tree.

3.3. Algorithms

It can be computationally intensive to identify the corresponding training data and calculate the covariance matrix for each testing point. To address this, we use the fast algorithm GrowFromRoot, as introduced in He, Yalow, and Hahn (2019). We propose an efficient method that pinpoints the exterior testing data and its neighborhood training data in the same leaf node by recursively partitioning the training and testing data at the same time. The procedures are summarized in Algorithms 1 and 2.

Certain algorithm design choices have been made to enhance performance and efficiency, and they are worth emphasizing. First, we partition the covariates into two categories: split variables \(v\) and active variables \(v_a\). Split variables are those that appear along the decision path from the root to a leaf node. Active variables \(v_a\) are a subset of split variables that satisfies the following condition: if there exists a testing point \(x \in X^e\) that falls outside the range of \(B_{lB}\) on that variable in leaf node \(b\), then it is considered an active variable. By using only active variables in defining the GP covariance matrix, the extrapolation
is limited to those variables only, which is intuitive and improves time efficiency. Second, when dealing with large sample size of training data, we subsample training data when performing the GP extrapolation. It helps manage computational resources. Third, to avoid the identification of exterior points being misled by any outliers, we define the hypercube $B_{lb}$ in each leaf node by the 95% quantile of the training data. These algorithmic choices collectively contribute to the effectiveness and efficiency of our method in extrapolation.

### 3.4. Simulation Studies

This section illustrates the performance of XBART-GP and compares it to various prediction inference methods in regression modeling. We carefully construct data-generating processes that incorporate covariate shifts and evaluate the model’s predictive ability on interior and exterior data points separately.

Specifically, we compare our proposed method with the latest frequentist approach Jackknife+ (Barber et al. 2021) and
Table 1. Four true f functions.

| Name           | Function                                                                 |
|----------------|--------------------------------------------------------------------------|
| Linear         | $x^2 y_1; \gamma_j = -2 + \frac{4j-1}{\delta-1}$                       |
| Single index   | $10 \sqrt{\sum_{j=1}^{10} \sin((5a) x_j)}; a = \sum_{j=1}^{10} (x_j - \gamma_j)^2; \gamma_j = -1.5 + \frac{j-1}{3}$ |
| Trig + poly    | $5 \sin(3x_1) + 2x_2^2 + 3x_3 x_4$                                      |
| Max            | $\max(x_1, x_2, x_3)$                                                   |

For both XBART and XBART-GP, we set hyperparameters as follows, num_trees = 20, num_sweeps = 120 (number of iterations), num_burnin = 20, Nmin = 20, $\tau = \text{var}(y)/\text{num}\_\text{trees}$ (where var(y) is the variance of the response variable in the training set). We choose $\theta_{\text{gp}} = \sqrt{10}$ and $\tau_{\text{gp}} = \tau$ to be the smoothness and scale parameters of the kernel function for XBART-GP. Regarding simulations of Jackknife+ and CV+ on Random Forest, referred to as Jackknife+ RF and CV+ RF, respectively, we adopt the default settings provided in the scikit-learn (Pedregosa et al. 2011) package in Python.

The real signal is drawn from four challenging functions, $f$, listed in Table 1. The response variable $y$ is generated independently from $y = f(x) + \epsilon$ with Gaussian noise $\epsilon$. In all cases, we generate 200 data points with covariates $x_j \overset{iid}{\sim} N(0,1)$ for $j = 1, \ldots, d = 10$ as the training set and another 200 data points with $x_j \overset{iid}{\sim} N(0,1.5^2)$ as the testing set.

To evaluate the performance on both in- and out-of-range data, here we simplify the concept of interior and exterior points. In our simulation studies, we classify any testing data outside the range of the training data as exterior points. Note that the exterior points reported here differ from the exterior points defined in XBART-GP, as the latter depend on each tree’s structure and vary across trees. With the aforementioned setup, approximately half of the testing data is interior, while the other half is exterior. We gauge the model’s performance separately on these interior and exterior points. We repeat the experiments on each data-generating process 10 times and calculate the average empirical probability of coverage and the average interval length with coverage level $1 - \alpha = 0.9$.

Simulation studies and time efficiency comparison on XBART-GP and its baselines were conducted in Python 3.8.10 on a Linux machine with Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz processor and 64GB RAM; eight cores were used for parallelization whenever it was applicable.

Figure 3 visualizes the simulation results for linear and single index functions. This figure compares the root mean square error (RMSE) of point prediction, coverage rate, and interval length of various approaches. The full results of four functions are summarized in Appendix B.1. Furthermore, we include an additional set of experiments to evaluate the performance of combining Jackknife+ and CV+ with XBART-GP, which is presented in Appendix B.3.

XBART-GP has the smallest RMSE of point prediction across all four functions on both interior and exterior points. Notably,
for interior data, our proposed method is the only one that delivers nominal coverage with reasonable interval lengths. Furthermore, XBART-GP demonstrates substantially greater prediction coverage than the baseline methods for exterior points. It is important to highlight that XBART-GP achieves better coverage without inflating the prediction interval too much compared to XBART.

When combining Jackknife+ and CV+ with XBART, both methods produce competitive RMSE, coverage, and interval length compared to XBART itself on interior points. However, the leave-one-out frequentist approach fails to extrapolate the predictive intervals on exterior points.

3.4.1. Time Efficiency Comparison

Next, we assess the performance and time efficiency of XBART-GP and other baselines as the sample size increases.

The training and testing covariates are generated from the same distributions as in the previous section, with testing covariates having a slightly larger variance. We generate the signal with the linear function in Table 1. We estimate the average coverage, interval width, and computational time of all six methods with sample size \( n = n_t = 50, 100, 150, 200, 300, 500 \) over 10 independent trials for each sample size. The target coverage level is \( 1 - \alpha = 0.9 \).

Figure 4 illustrates the performance comparison of XBART-GP and other baseline methods on exterior points as the sample size increases. The performances on interior points are similar, which are summarized in Appendix B.2. Note that the proportion of the exterior points varies according to the training size. Empirically, the percentage of exterior points decreases from 80% to 40% when the sample size increases from 50 to 500 on 10-dimensional covariates in this particular data-generating process. The x-axes in Figure 4 only reflect the sizes of the training sets. The running time is reported in the logarithm scale evaluating all testing data, including interior and exterior points.

On exterior points, XBART-GP achieves the smallest RMSE of point prediction and maintains a consistently high coverage level close to 90% as the sample size grows. In contrast, the point-wise prediction of the baseline approaches reduces slightly. Furthermore, the coverage of the baseline methods significantly decreases as the models become more confident and generate narrower predictive intervals on larger datasets.

Regarding running time, XBART and CV+ XBART are the most efficient methods. The run-time of XBART-GP increases almost linearly with the amount of training and testing sample size, but it is worth the time when the distribution of the test set shifts from the training set.

4. Application of Local GP Extrapolation on Causal Inference

4.1. Background

An illustrative example of an applied extrapolation problem arises in treatment effect estimation, specifically when one of the necessary assumptions is violated within the potential outcome framework (Imbens and Rubin 2015) for estimating the treatment effect.

Let \( Y \) denote the response variable and \( Z \) represent the binary treatment variable, where \( Z_i = 1 \) indicates that the \( i \)th observation being treated, and \( Z_i = 0 \) indicates that the \( i \)th observation is in the control group. The potential outcomes under treatment and control are denoted as \( Y_i(1) \) and \( Y_i(0) \) for the \( i \)th unit, respectively. The observable outcome can be expressed as \( Y_i = Z_i Y_i(1) + (1 - Z_i) Y_i(0) \). The treatment effect on an individual \( x_i \) is defined as \( \tau(x_i) = Y_i(1) - Y_i(0) \).

Most causal inference methods using the potential outcome framework also rely on the following assumptions:

1. Stable Unit Treatment Value Assumption (SUTVA);
2. Ignorability assumption: \( Y_i(1), Y_i(0) \perp \perp Z_i \mid X_i \);
3. Positivity assumption: \( 0 < P(Z_i = 1 \mid x_i) < 1 \).

The positivity assumption, also known as the overlap assumption, plays a central role in causal inference. In this study, we divide the covariate space into two distinct regions: the overlap region where the positivity assumption holds,
and the non-overlap region where it is violated. Following the classification proposed by Zhu et al. (2021), violations of positivity assumptions can be categorized into two categories, structural and practical violation.

Structural violations occur when it is impossible for individuals with specific covariate values to receive treatment, leading to the treatment effect for these individuals or covariate values being irrelevant. One potential solution to address this issue is to “trim” the population, as described in (Ho et al. 2007; Crump et al. 2009; Petersen et al. 2012). This approach allows for the estimation of the treatment effect only in the overlap region (where the positivity assumption holds).

Practical violation refers to situations where a subset of covariate values is observed exclusively in either the treatment or control group, typically due to selection bias or limited sample size. Nevertheless, both the average treatment effect (ATE) for the entire population and the conditional average treatment effect (CATE) for specific covariate values remain of interest (Yang and Ding 2018). In these cases, the strategy is to perform model-based extrapolation, extending an estimated response surface into regions where no data was observed.

Recently, several extrapolation methods have been developed to estimate the ATE of the entire population with poor overlap in the covariate space. Li, Morgan, and Zaslavsky (2018) and Li, Thomas, and Li (2018) proposed using overlap weights to balance the covariates for estimating the ATE. (Nethery, Mealli, and Dominici 2019) introduced a method that defines overlap and non-overlap regions based on the propensity score. They then use BART to estimate the treatment effect in the overlap region and use a spline model to extrapolate the treatment effect in the non-overlap region. Zhu, Mitra, and Roy (2023) addressed this problem by modeling the treatment effect with the Gaussian process model, which by its nature can extrapolate based on the covariate kernel.

### 4.2. Local GP Extrapolation with Bayesian Causal Forest

In this section, we propose a new method XBCF-GP to extrapolate CATE in the non-overlap region by combining the local GP extrapolation method with the Accelerated Bayesian Causal Forest (XBCF) model (Krantsevich, He, and Hahn 2023). First, we briefly review the structure of the XBCF model. Next, we formally define the overlap and non-overlap area in the ensemble trees framework. Finally, we describe the modifications we have made to integrate the local GP algorithm for extrapolating the treatment effect estimated by XBCF.

XBCF is a recent approach for estimating heterogeneous treatment effects based on Bayesian Causal Forest (Hahn et al. 2020). The model expresses the observable as

\[ y_i = \mu(x_i, \hat{\pi}_i) + b_i \tau(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2_{\epsilon_i}), \]

\[ a \sim N(0, 1), \quad b_i, b_i \sim N(0, 1/2) \]  

(13)

where \( \hat{\pi}_i \) is the estimated propensity score. Function \( \mu(x_i, \hat{\pi}_i) \) aims to capture the prognostic effect and function \( \tau(x_i) \) captures the treatment effect. Both functions are modeled by a sum of XBART trees, respectively.

Using the derived split criterion of XBCF, the marginal likelihood of treatment trees in \( \tau(x_i) \) always prefers to stop splitting when the leaf nodes only consist of either the treatment group or the control group. On the other hand, the prognostic forest \( \mu(x_i, \hat{\pi}_i) \) is not affected by the positivity violation as it does not depend on the treatment variable \( z \). Consequently, both estimators will be biased in the presence of positivity violation. Nevertheless, XBCF has demonstrated its ability to effectively capture the heterogeneity in the overlap region, as demonstrated in prior work (Krantsevich, He, and Hahn 2023). This makes it possible to provide inferences on the treatment effect in the non-overlap region by combining it with the local Gaussian process method for BART trees.

While XBCF relies on the three basic assumptions in the potential outcome framework described in the previous section, the violation of the positivity assumption in practical scenarios prevents firm conclusions about ATE nor CATE in the non-overlap region. To address this limitation, we make the assumption that there is no abrupt change in the treatment effect function within both overlap and non-overlap regions. This assumption allows us to make inferences about CATE using the extrapolated predictive interval obtained from local Gaussian processes.

To extrapolate the treatment effect in non-overlap regions using the XBCF model, we adapt the local Gaussian process technique as follows. Consider a leaf node \( b \) in a regression tree, with covariate subspace \( A_b \). Let matrix \( X^b \) denote the training data that falls into the leaf node, which can be separated into a treatment group \( X^b_{tr} \) and a control group \( X^b_{te} \) based on the corresponding treatment variable \( z \). The ranges of covariates in the treated and the control form two hypercubes denoted as \( B^T \) and \( B^C \), respectively. The overlap region in the leaf node is the intersection of \( B^T \) and \( B^C \), or \( B^O = B^T \cap B^C \). Consequently, the non-overlap area is \( B^NO = A_X \setminus B^O \).

Note that in each treatment tree, the residuals in the non-overlap region are biased in a direction relative to the true treatment effect. This is because of the treatment trees’ no-splitting behavior in the non-overlap region. As a result, if residuals from the non-overlap region are included in the Gaussian process, the prediction will be extrapolated in the opposite direction. Therefore, unlike XBART-GP, we exclusively use the training data from the overlap region, denote as \( X^O_{tr} \), to extrapolate the prediction for testing data in the non-overlap area, denoted as \( X^O_{te} \), namely

\[ \left( r^O_{te} \right) \sim N \left( \mu_j, \left( \hat{\Sigma}_{X^O_{te}X^O_{te}}, \frac{\sum X^O_{te}X^O_{te} + \frac{1}{L} \hat{\sigma}^2_z}{\hat{\sigma}^2_z} \right) \right). \]

(14)

Here \( r^O_{te} \) represents a vector of partial residuals for the current treatment tree, while \( r^O_{te} \) is a vector of residuals we wish to predict for the non-overlap testing data. \( \hat{\sigma}^2_z \) is a diagonal matrix with diagonal elements equal to estimated variance \( \hat{\sigma}^2_i \) or \( \hat{\sigma}^2_z \) depending on the treatment group of the training data and \( L \) is the total number of prognostic and treatment trees.

The prediction algorithm for the treatment trees in XBCF-GP follows the same procedures as in Algorithm 2 except for the difference we describe above. Specifically, in line 4, we replace the out-of-range hypercube \( B^O \) by the overlap area \( B^O \). In line 6, the subset of training data \( X^O_{tr} \) is sampled from the training data \( X^O_{tr} \) from the overlap area instead of the entire training set. Then in line 7, we choose to extrapolate the testing data \( X^O_{te} \) in the
non-overlap area. Similar to the algorithm design choices made in XBART-GP, the overlap area is defined by the 95% quantiles of the treated and control data for robustness concerns.

In addition to the modification we make to the XBART-GP algorithm, we also introduce a tweak to the original XBCF model to ensure the extrapolation works smoothly. The accuracy of the GP in providing accurate estimates relies on having a minimum amount of overlap data in the leaf nodes. Although it is uncommon for the treatment trees in XBCF to split in the non-overlap area, it can still occur in rare cases and result in suboptimal extrapolation. To address this issue, we implemented a strong prior on the treatment trees to discourage splits that do not have sufficient treatment and control data in their children nodes. The amount of overlap data required in a leaf node can be controlled through the hyperparameter \( \text{Nmin} \). This hyper-parameter is critical as it determines the balance between the quality of the method's estimates near the overlap area boundary and its extrapolation performance in the non-overlap region. In our experiments using 500 data points, we set \( \text{Nmin} \) to 20.

The XBCF-GP model combines the strengths of the Bayesian Causal Forest and the Gaussian Process model. XBCF excels at accurately and efficiently estimating homogeneous and heterogeneous treatment effects on large datasets with many covariates, provided the positivity assumption holds. By applying the GP model on a per-leaf node basis, XBCF-GP enables the extrapolation of treatment effects in regions where the positivity assumption is violated by leveraging the most relevant covariates in the surrounding area.

Our method provides a more effective way to assess the uncertainty of the estimated treatment effect in cases where the positivity assumption is violated. By using the local Gaussian process, our uncertainty estimation takes into account the distance between the testing data and the overlap region. This means that if the testing data is significantly distant from the overlap region, XBCF-GP will reflect this uncertainty by generating wider prediction intervals.

### 4.3. Simulation Study

This section compares prediction interval coverage and length under the setting of causal inference when the positivity assumption is violated. To demonstrate the performance of XBCF-GP, we compare it with a set of alternative extrapolation approaches proposed by Nethery, Mealli, and Dominici (2019), originally designed for estimating the ATE. The baseline methods we use are summarized as follows:

- **SBART** Also known as BART-Stratified. Fit separate BART models for the treated and the control group, then estimate the treatment effect as the difference between the expected values of the treated and the control models.

- **UBART** Untrimmed BART is another baseline method introduced in Nethery, Mealli, and Dominici (2019), in which a single BART model is fitted with covariates, treatment variable, and estimated propensity score. The treatment effect is estimated by predicting the potential outcomes with the posterior predictive distributions.

- **BART+SPL** Nethery, Mealli, and Dominici (2019) proposed this method to estimate the average treatment effect by using BART in the overlap area and a spline model in the non-overlap area. The region of overlap is defined by propensity score with recommended parameters \( a = 0.1 \) and \( b = 7 \).

Figure 5 illustrates the performance of XBCF-GP and other baseline methods on a one-dimensional toy dataset. The independent variable \( x \) is uniformly drawn from the range \([-10, 10]\). The prognostic function is the sine function, and the treatment effect is 0.25\( x \). The probability for a sample being treated is \( \pi(x) = \max(0, \min(1, 0.08x + 0.5)) \). The signal is generated from \( f(x) = \sin(x) + (0.25x)z \) with treatment variable \( z \sim \text{Bern}(\pi(x)) \). The response values \( y \) are generated as \( y = f(x) + \epsilon \), where \( \epsilon \sim N(0, 0.2 \times \text{sd}(f)) \) with the standard deviation \( \text{sd}(f) \) taken over the training data. The fitted models in this example use the same parameter setup as in the simulation study, which will be specified shortly.

The figure demonstrates how different models estimate the CATE. The first three methods (XBCF, XBCF-GP, and SBART) show robustness to noise. However, the last two methods (UBART and BART+SPL), which directly estimate CATE using the response variable, introduce significant noise in the estimates. All three BART-based methods are susceptible to the influence of the prognostic structure, especially in the non-overlap area. XBCF dodges the bullet by setting up a separate BART forest to control for the prognostic effect, but it cannot extrapolate properly in the non-overlap region. On the other hand, XBCF-GP inherits the well-regularized structure from XBCF and provides smooth and accurate extrapolations of treatment effect estimations.

For a comprehensive comparison of these methods, we adopt the same data-generating process used in Hahn et al. (2020) and Krantsevich, He, and Hahn (2023), but we modify the propensity function to create the positivity violation scenario. The covariate vector \( x \) consists of five variables. The first three (denoted as \( x_1, x_2, x_3 \)) are generated from the standard normal distribution. \( x_4 \) is a binary variable, and \( x_5 \) is an unordered categorical variable with levels denoted as 1, 2, 3. The prognostic function can be either linear or nonlinear:

\[
\mu(x) = \begin{cases} 
-6 + g(x_4) + 6 |x_3 - 1| & \text{linear} \\
1 + g(x_4) + x_1 x_3 & \text{nonlinear},
\end{cases}
\]

where \( g(1) = 2, g(2) = -1 \) and \( g(3) = -4 \). The treatment effect can be either homogeneous or heterogeneous:

\[
\tau(x) = \begin{cases} 
3 & \text{homogeneous} \\
1 + 2x_2 x_5 & \text{heterogeneous}.
\end{cases}
\]

In the simulation studies, we adjust the propensity function such that roughly 20% of the data has a propensity score of 1, 20% of the data has a propensity score of 0, and the rest of the data has a propensity score that is between 0 and 1. The propensity function \( \pi(x) \) is given by

\[
\pi(x) = \max(0, \min(1, h(x))),
\]

\[
h(x) = 1.1 \Phi(3\mu(x)/s - 0.5x_1 - c) - 0.15 + u_i/10,
\]

where \( s \) is the standard deviation of \( \mu(x) \) taken over the observed samples and \( u_i \sim \text{Uniform}(0,1) \). For the linear prognostic function, we set \( c = 0 \), while for the nonlinear function, we
Figure 5. An example of treatment effect estimation with positivity violation on all baseline methods. The control group is heavily distributed on the left side and the treatment group is distributed on the right side. The two vertical lines indicate the boundaries of the overlap area. The line in the middle represents the true treatment effect, while the curved lines/dots in the center denote the estimated treatment effect for each method. The lower and upper curved lines/dots show their respective 95% predictive intervals.

use $c = 3$. Any propensity score exceeding 1 or less than 0 is bounded to 1 or 0, respectively. In addition, we add a Gaussian noise $\epsilon \sim N(0, 0.5 \times sd(\mu(x) + \tau(x)z))$ with the standard deviation taken over the samples to generate the response values.

To gauge the performance of XBCF-GP across overlap and non-overlap regions, we evaluate the average treatment effect (ATE) and conditional average treatment effect (CATE) using three key metrics: average root mean square error (RMSE), coverage, and average interval length. In addition, we also compare the time efficiency of the proposed method to baseline methods. The reported results are averaged over 10 independent replications for each scenario, with a sample size of $n = 500$.

For XBCF-GP, we set hyper-parameter $\theta = \sqrt{10}$ and $\tau = \hat{\sigma}^2/L_\tau$, where $\hat{\sigma}^2$ is the estimated variance and $L_\tau = 20$ is the default number of treatment trees in the XBCF model. We choose a different scale parameter $\tau_{gp}$ from XBART-GP as the variance of the treatment effect tends to be smaller than the total variance of the observed data in most cases. As for the other baseline methods, including XBCF, we use their default parameters for all scenarios. The experiments were conducted in R 4.2.2 on the same machine described in the previous sections.

Table 2 summarizes the simulation results of various methods. Our focus is on CATE estimation, where XBCF-GP consistently outperforms the other methods. It achieves more accurate CATE estimation, with competitive CATE coverage and narrower intervals compared to the alternatives. XBCF-GP demonstrates its capability to enhance the accuracy and coverage of both ATE and CATE estimation based on its base model.

5. Discussion

The local Gaussian process extrapolation technique developed in this study offers several advantages over the standard BART extrapolation methods implemented in widely available software. First, unlike BART, which extrapolates a constant function beyond the range of the training data range, our XBART-GP uses a Gaussian process at each leaf that ensures “local” Gaussian process extrapolation. This approach incorporates variable relevance by considering only the variables selected as splitting rules for covariance kernel calculation. Additionally, the extrapolation takes place at each tree, resulting in improved prediction. Even at points within the convex hull of the training data, GP prediction at local exterior points enforces additional smoothness.

Second, the prediction intervals obtained from BART are primarily dominated by the estimated residual variances, resulting in nearly constant interval widths at any exterior point. In contrast, our local GP extrapolation method yields interval widths that rapidly expand as the point moves further away from
Third, the local GP approach does not require additional training time compared to a regular BART model fit. Instead, the extrapolation component is a direct modification of the posterior predictive distribution defined in terms of BART posterior samples and a specified covariance function. Our approach addresses the question of how to use a well-fitting BART model at interpolation points to define an extrapolation model. By grafting a Gaussian process to each leaf node for the extrapolation purpose only; posterior uncertainty in the BART model parameters is then propagated by averaging over posterior samples. The philosophy behind this approach is to retain the BART likelihood function unchanged when it fits the training data well, while also allowing for a departure from the pure BART model for points far from the training data. Extrapolation always requires a leap of faith, and there is no guarantee that any extrapolation method will succeed, but we argue that there are conceptual and computational advantages to disentangling the extrapolation model from the interpolation model.

Finally, we demonstrate the application of local GP extrapolation of BART models to address the problem of imperfect overlap in treatment effect estimation, a common challenge in applied causal inference. By incorporating the local GP extrapolation technique into the Bayesian causal forest model, violations of the positivity assumption in practice can be effectively handled. Specifically, we can extrapolate the treatment effect surface directly rather than composing it as the difference between two separate extrapolations (one for each treatment arm). Our simulation studies demonstrate that XBCF-GP produces treatment effect estimates with good coverage on a variety of data-generating processes. These tools will help other researchers to draw more reliable inferences from applied data which frequently exhibit imperfect overlap.

Supplementary Materials

Appendix to Local Gaussian process extrapolation for BART: Additional introduction to baseline method Jackknife+ (Barber et al. 2021) in Appendix A; extensive simulation results for XBART-GP and its baseline in Appendix B. (Appendix.pdf)

R-package for XBART-GP: R-package “XBART” is adapted from the original XBART model (https://github.com/JingyuHe/XBART) (He, Yalov, and Hahn 2019; He and Hahn 2021) and contains code to perform local Gaussian process extrapolation on trained XBART model described in this article. Please read the file README contained in the zip file for installation instructions. (XBART.zip, zip archive)

Python-package for XBART-GP: Python-package “XBART” is adapted from the original XBART model (https://github.com/JingyuHe/XBART) (He, Yalov, and Hahn 2019; He and Hahn 2021) and contains code to perform local Gaussian process extrapolation on trained XBART model described in this article. Please read the file README contained in the zip file for installation instructions. (XBART-py.zip, zip archive)

R-package for XBCF-GP: R-package “XBCF” is adapted from the original XBCF model (https://github.com/socket778/XBCF) (Krantsevich, He, and Hahn 2023) and contains code to perform local Gaussian process extrapolation on trained XBCF model described in the article. Please read the file README contained in the zip file for installation instructions. (XBCF.zip, zip archive)

Python code for XBART-GP simulations: The supplemental files contain code to perform simulation experiments described in Section 3.4. (XBART-GP simulation.zip, zip archive)

R code for XBCF-GP simulations: The supplemental files contain code to perform simulation experiments described in Section 4.3. (XBCF-GP simulation.zip, zip archive)

Disclosure Statement

No potential conflict of interest was reported by the authors.
Funding

Jingyu He gratefully acknowledges funding for this project fully supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 2154921).

ORCID

Meijia Wang © http://orcid.org/0000-0002-8633-3521
Jingyu He © http://orcid.org/0000-0003-0370-8867

References

Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. (2021), "Prediction Inference with the Jackknife+," *The Annals of Statistics*, 49, 486–507. [724,725,726,734]
Breiman, L. (2001), "Random Forests," *Machine Learning*, 45, 5–32. [724]
Breiman, L., Friedman, J., Olshen, R., and Stone, C. J. (1984), *Classification and Regression Trees*, Boca Raton, FL: Chapman and Hall/CRC. [724]
Chen, T., and Guestrin, C. (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. [724]
Chipman, H. A., George, E. I., McCulloch, R. E., et al. (2010), "BART: Bayesian Additive Regression Trees," *The Annals of Applied Statistics*, 4, 266–298. [724,725,726]
Crump, R. K., Hotz, V. J., Imbens, G. W., and Mitnik, O. A. (2009), "Dealing with Limited Overlap in Estimation of Average Treatment Effects," *Biometrika*, 96, 187–199. [731]
D’Amour, A., Ding, P., Feller, A., Lei, L., and Sekhon, J. (2021), "Overlap in Observational Studies with High-Dimensional Covariates," *Journal of Econometrics*, 221, 644–654. [725]
Efron, B., and Stein, C. (1981), "The Jackknife Estimate of variance," *The Annals of Statistics*, 9, 586–596. [724]
Gramacy, R. B., and Lee, H. K. H. (2008), "Bayesian Treed Gaussian Process Models with an Application to Computer Modeling," *Journal of the American Statistical Association*, 103, 1119–1130. [724]
Hahn, P. R., Murray, J. S., Carvalho, C. M., et al. (2020), "Bayesian Tree Models for Causal Inference: Regularization, Confounding, and Heterogeneous Effects," *Baysian Analysis*, 15, 965–1056. [725,731,732]
He, J., and Hahn, P. R. (2021), "Stochastic Tree Ensembles for Regularized Nonlinear Regression," *Journal of the American Statistical Association*, 1–61 (just-accepted). [724,726,727]
He, J., Yalov, S., and Hahn, P. R. (2019), "XBART: Accelerated Bayesian Additive Regression Trees," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1130–1138. [726,727,734]
Ho, D. E., Imai, K., King, G., and Stuart, E. A. (2007), "Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference," *Political Analysis*, 15, 199–236. [731]
Imbens, G. W. and Rubin, D. B. (2015), *Causal Inference in Statistics, Social, and Biomedical Sciences*, Cambridge: Cambridge University Press. [730]
Kapelner, A., and Bleich, J. (2013), "Prediction with Missing Data via Bayesian Additive Regression Trees," *Canadian Journal of Statistics*, 43, 224–239. [724]
Krantsevich, N., He, J., and Hahn, P. R. (2023), "Stochastic Tree Ensembles for Estimating Heterogeneous Effects," in *International Conference on Artificial Intelligence and Statistics*, pp. 6120–6131. PMLR. [725,731,732,734]
Li, F., Morgan, K. L., and Zaslavsky, A. M. (2018), "Balancing Covariates via Propensity Score Weighting," *Journal of the American Statistical Association*, 113, 390–400. [731]
Li, F., Thomas, L. E., and Li, F. (2018), "Addressing Extreme Propensity Scores via the Overlap Weights," *American Journal of Epidemiology*, 188, 250–257. [731]
Nethery, R. C., Mealli, F., and Dominici, F. (2019), "Estimating Population Average Causal Effects in the Presence of Non-overlap: The Effect of Natural Gas Compressor Station Exposure on Cancer Mortality," *The Annals of Applied Statistics*, 13, 1242–1267. [725,731,732]
Papadopoulos, H. (2008), *Inductive Conformal Prediction: Theory and Application to Neural Networks*, Rijeka: INTECH Open Access Publisher. [724]
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011), "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12, 2825–2830. [729]
Petersen, M. L., Porter, K. E., Gruber, S., Wang, Y., and van der Laan, M. J. (2012), "Diagnosing and Responding to Violations in the Positivity Assumption," *Statistical Methods in Medical Research*, 21, 31–54. [731]
Rasmussen, C. E. (2003), "Gaussian Processes in Machine Learning," in *Advanced Lectures on Machine Learning. Summer School on Machine Learning*, eds. O. Bousquet, U. von Luxburg, and G. Rätsch, pp. 63–71, Berlin: Springer. [726]
Skyrms, B. (2006), "Diachronic Coherence and Radical Probabilism," *Philosophy of Science*, 73, 959–968. [727]
Starling, J. E., Murray, J. S., Carvalho, C. M., Bukowski, R. K., and Scott, J. G. (2018), "Bart with Targeted Smoothing: An Analysis of Patient-Specific Stillbirth Risk," *The Annals of Applied Statistics*, 14, 28–50. [725]
Starling, J. E., Murray, J. S., Lohr, P. A., Aiken, A. R. A., Carvalho, C. M., and Scott, J. G. (2021), "Targeted Smooth Bayesian Causal Forests: An Analysis of Heterogeneous Treatment Effects for Simultaneous vs. Interval Medical Abortion Regimens Over Gestation," *The Annals of Applied Statistics*, 15, 1194–1219. [725]
Vovk, V. (2015), "Cross-Conformal Predictors," *Annals of Mathematics and Artificial Intelligence*, 74, 9–28. [724]
Yang, S., and Ding, P. (2018), "Asymptotic Inference of Causal Effects with Observational Studies Trimmed by the Estimated Propensity Scores," *Biometrika*, 105, 487–493. [731]
Zhu, A. Y., Mitra, N., and Roy, J. (2023), "Addressing Positivity Violations in Causal Effect Estimation Using Gaussian Process Priors," *Statistics in Medicine*, 42, 33–51. [725,731]
Zhu, Y., Hubbard, R. A., Chubak, J., Roy, J., and Mitra, N. (2021), "Core Concepts in Pharmacoepidemiology: Violations of the Positivity Assumption in the Causal Analysis of Observational Data: Consequences and Statistical Approaches," *Pharmacoepidemiology and Drug Safety*, 30, 1471–1485. [731]