Robust Estimation of Heterogeneous Treatment Effects: An Algorithm-based Approach

Ruohong Li[1,2], Honglang Wang[3], Yi Zhao[1,2], Jing Su[1], and Wanzhu Tu[1,2] (wtu1@iu.edu)
[1] Department of Biostatistics and Health Data Science, Indiana University School of Medicine
[2] Fairbanks School of Public Health
[3] Department of Mathematical Sciences, Indiana University-Purdue University Indianapolis

August 26, 2021

Abstract

Heterogeneous treatment effect estimation is an essential element in the practice of tailoring treatment to suit the characteristics of individual patients. Most existing methods are not sufficiently robust against data irregularities. To enhance the robustness of the existing methods, we recently put forward a general estimating equation that unifies many existing learners. But the performance of model-based learners depends heavily on the correctness of the underlying treatment effect model. This paper addresses this vulnerability by converting the treatment effect estimation to a weighted supervised learning problem. We combine the general estimating equation with supervised learning algorithms, such as the gradient boosting machine, random forest, and artificial neural network, with appropriate modifications. This extension retains the estimators' robustness while enhancing their flexibility and scalability. Simulation shows that the algorithm-based estimation methods outperform their model-based counterparts in the presence of nonlinearity and non-additivity. We developed an **R** package, **RCATE**, for public access to the proposed methods. To illustrate the methods, we present a real data example to compare the blood pressure-lowering effects of two classes of antihypertensive agents.

Keywords: Causal inference, machine learning, robust estimation, heterogeneous treatment effect, least absolute deviation

---

This is the author's manuscript of the article published in final edited form as:

Li, R., Wang, H., Zhao, Y., Su, J., & Tu, W. (2021). Robust estimation of heterogeneous treatment effects: An algorithm-based approach. Communications in Statistics - Simulation and Computation, 1–18. https://doi.org/10.1080/03610918.2021.1974883

1 Introduction

The practice of precision medicine relies on a sound causal understanding of treatment effects varying with patient characteristics. Estimating such effects, known as the heterogeneous treatment effects, from observational data is typically done within the Neyman-Rubin causal framework with appropriate assumptions (Sekhon, 2008). Popular approaches include the *Quality* or Q-learning that directly regresses the outcomes on patient characteristics (Watkins and Dayan, 1992; Watkins, 1989) and the *Advantage* or A-learning that models the contrasts among treatments (Murphy, 2003; Robins, 2004).

Despite the general applicability of these estimation methods, practical challenges abound: (1) Few existing estimators are designed to deal with data irregularities and high dimensionality. (2) Model-based methods remain vulnerable to model misspecification. (3) Few software packages are available for practical use in an off-the-shelf fashion and can handle the above issues. The lack of ready-made robust analytical tools has hindered the practical use of these methods because practitioners are rarely in a position to implement and test sophisticated causal inference methods.

Efforts have been made to alleviate the impact of data irregularities. For example, Xiao et al. (2019) extended the L_2-based R-learner (Nie and Wager, 2017), a method under the general A-learning umbrella, to the pinball loss function. More recently, our research team has put forward a general estimating equation for robust estimation of heterogeneous treatment effects, supported by strong theoretical and empirical evidence (Li et al., 2021). This estimating equation unifies many of the existing methods, including the R-learner (Nie and Wager, 2017), inverse propensity weighting (Hirano et al., 2003; Horvitz and Thompson, 1952), various modified outcome and covariate methods (with and without efficiency augmentation) (Chen et al., 2017; Tian et al., 2014), and the augmented inverse propensity weighting method (Robins and Rotnitzky, 1995). We showed that under fairly general regularity conditions, the robust estimators ascertained from the general estimating equation are asymptotic normal to allow for valid inference. Despite its broad coverage and good theoretical properties, the general estimating equation estimators are not robust against model misspecifications, nor are they easy to implement in practical data analyses.

This paper extends our previous work by combining the A-learner from the general estimating equation with supervised learning algorithms to further enhance its robustness again model misspecifications. This modification also frees analysts from the tedious and error-prone work of model building. We implement the causal inferences tools in the form of an **R** package - **RCATE**, short for Robust Estimation of the Conditional Average Treatment Effects, for a scalable solution to heterogeneous treatment estimation.

2 Methods

2.1 Notation and assumptions

Let T be a binary variable for treatment assignment: $T = 1$ if a patient is in the treatment group, and $T = -1$ otherwise. We define $Y^{(1)}$ and $Y^{(-1)}$ as the potential outcomes under treatments $T = 1$ and $T = -1$, respectively. Here, $Y^{(1)}$ and $Y^{(-1)}$ are assumed to be univariate and continuous. Let $\mathbf{X}$ be the p-dimensional pre-treatment covariates. In an observational study, one observes T, $\mathbf{X}$, and $Y = I(T = 1)Y^{(1)} + I(T = -1)Y^{(-1)}$, where

$I(\cdot)$ is an indicator function. We assume that the data $\{(Y_i, T_i, \mathbf{X}_i)\}_{i=1}^n$ are independent and identically distributed (i.i.d.). The estimation target is the treatment effect $\tau_0(\mathbf{x})$, commonly known as the conditional average treatment effect (CATE)

$$\tau_0(\mathbf{x}) = E[Y^{(1)} - Y^{(-1)}|\mathbf{X} = \mathbf{x}] = E[Y|\mathbf{X} = \mathbf{x}, T = 1] - E[Y|\mathbf{X} = \mathbf{x}, T = -1],$$

where the last part follows from the ignorability assumption below. With a binary treatment indicator, one can always express the conditional mean outcome as $E(Y|\mathbf{X}, T) = b_0(\mathbf{X}) + \frac{T}{2}\tau_0(\mathbf{X})$, with $b_0(\mathbf{x}) = \frac{1}{2}(E[Y^{(1)}|\mathbf{X} = \mathbf{x}] + E[Y^{(-1)}|\mathbf{X} = \mathbf{x}])$. This leads to a general interaction model

$$Y_i = b_0(\mathbf{X}_i) + \frac{T_i}{2}\tau_0(\mathbf{X}_i) + \varepsilon_i. \tag{1}$$

We further define $\mu(\mathbf{x}) = E[Y|\mathbf{X} = \mathbf{x}]$, $\mu^{(1)}(\mathbf{x}) = E[Y|\mathbf{X} = \mathbf{x}, T = 1]$, and $\mu^{(-1)}(\mathbf{x}) = E[Y|\mathbf{X} = \mathbf{x}, T = -1]$.

To estimate $\tau_0(\mathbf{X}_i)$, we operate under the following assumptions: (1) *Ignorability* — Treatment assignment T_i is independent of the potential outcomes $(Y_i^{(1)}, Y_i^{(-1)})$ given the covariates $\mathbf{X}_i$, i.e., $\{Y_i^{(1)}, Y_i^{(-1)} \perp\!\!\!\perp T_i|\mathbf{X}_i\}$; (2) *Positivity* — The propensity score is strictly between 0 and 1, i.e., $p(\mathbf{x}) := P(T = 1|\mathbf{X} = \mathbf{x}) \in (0, 1)$; (3) *Stable Unit Treatment Values Assumption (SUTVA)* – the potential outcome in one individual is only affected by the treatment he receives; (4) *Conditional Independence Error* – The error is independent of the treatment assignment, conditional on the covariates, i.e., $\{\varepsilon_i \perp\!\!\!\perp T_i|\mathbf{X}_i\}$. We further assume that the conditional expectation of the error exists. The commonly seen assumption of $E[\varepsilon] = 0$ is sufficient but not necessary.

2.2 The existing methods

There is a sizable literature on the estimation of CATE using observational data. Caron et al. (2020) and Zhang et al. (2020) provided state-of-the-art reviews of the methods for CATE estimation. We summarize the existing methods in Table 1, along with the available analytical software. Importantly, most of these methods are based on the L_2-loss function, whose performance deteriorates with data irregularity.

(Table 1 goes here)

The estimating equation that we proposed (Li et al., 2021), while not covering all methods in Table 1, does accommodate many loss functions, including the L_1-loss, Huber loss, and Bi-square loss, and thus greatly enhancing the estimators' robustness against data irregularities. In the next section, we briefly review this formulation and the methods it covers.

2.3 A unified estimating equation for CATE

We previously described the general estimating equation that covers many of the existing methods for CATE estimation. An important feature of the estimating equation is that it readily accommodates the L_1 loss function so that robust estimation can be derived; see Li et al. (2021) for detailed derivation and theoretical development. Briefly, we consider the following estimating equation

$$\min_{\tau(\cdot)\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} w(\mathbf{X}_i, T_i) M(Y_i - g(\mathbf{X}_i) - c(\mathbf{X}_i, T_i)\tau(\mathbf{X}_i)), \tag{2}$$

where $\mathcal{F}$ is the treatment effect function space subject to predefined assumptions such as smoothness, $M(\cdot)$ is a user-specified loss function, and the two weight functions $w(\mathbf{x}, t)$ and

$c(\mathbf{x}, t)$ are subject to the following constraints:

C1. $p(\mathbf{x})w(\mathbf{x}, 1)c(\mathbf{x}, 1) + (1 - p(\mathbf{x}))w(\mathbf{x}, -1)c(\mathbf{x}, -1) = 0$;

C2. $c(\mathbf{x}, 1) - c(\mathbf{x}, -1) = 1$;

C3. $w(\mathbf{x}, t)c(\mathbf{x}, t) \neq 0$.

Equation (2) covers many existing popular methods for heterogeneous treatment effect estimation, including the modified covariate methods (MCM) (Chen et al., 2017; Tian et al., 2014), MCM with efficiency augmentation (MCM-EA) (Chen et al., 2017; Tian et al., 2014), inverse propensity score weighting (IPW) (Hirano et al., 2003; Horvitz and Thompson, 1952), augmented inverse propensity score weighting (AIPW) (Robins and Rotnitzky, 1995), and the R-learner (RL) (Nie and Wager, 2017). In Table 2, we list the functions c, w, and g that meet the constraints for popular A-learning methods.

(Table 2 goes here)

An important appeal of the general formulation is its flexibility in specifying M, a feature that enhances the robustness against various forms of data irregularities through the use of L_1 and Huber loss functions. Here, we used the L_1-loss for illustration purpose. With the L_1-loss and under the above conditions, we have

$$\tau_0(\cdot) = \arg\min_{\tau(\cdot)} E\big[w(\mathbf{X}_i, T_i) \cdot |Y_i - g(\mathbf{X}_i) - c(\mathbf{X}_i, T_i)\tau(\mathbf{X}_i)|\big|\mathbf{X}_i\big]. \tag{3}$$

In the present research, we estimate $\tau(\mathbf{X})$ using modified supervised learning algorithms, which side-step the need to specify τ, and thus enhancing the method's flexibility and scalability without sacrificing its robustness against data irregularities.

2.4 Supervised learning algorithms for robust CATE Estimation

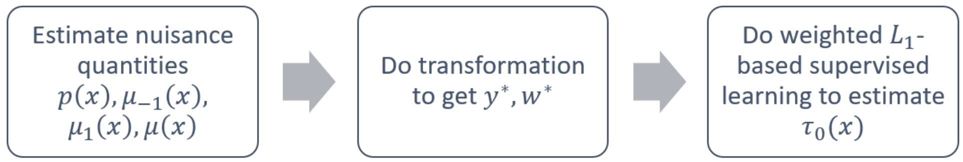
Through a transformation, CATE estimation in (3) under the L_1-loss function can be seen as an optimization problem of ordinary least absolute deviation (LAD),

$$\hat{\tau}(\cdot) = \arg\min_{\tau(\cdot)\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} w_i^*(\mathbf{X}_i, T_i)|Y_i^* - \tau(\mathbf{X}_i))|, \tag{4}$$

where $Y_i^* = \frac{Y_i - g(\mathbf{X}_i)}{c(\mathbf{X}_i, T_i)}$ and $w_i^*(\mathbf{X}_i, T_i) = w_i(\mathbf{X}_i, T_i)|c(\mathbf{X}_i, T_i)|$. We now show how to adapt three supervised learning algorithms for this purpose.

Depending on the structured assumptions one chooses for $\mathcal{F}$, one can select an appropriate learning algorithm for estimation, while taking care of the high dimensionality in $\mathbf{X}$. In Section 3, we compare the L_1 and L_2-based algorithms. For the L_2-based methods, the transformed weight is $w_i^*(\mathbf{X}_i, T_i) = w_i(\mathbf{X}_i, T_i)c(\mathbf{X}_i, T_i)^2$.

With the objective function in (4), different supervised learning algorithms can be used to estimate CATE - the optimization becomes a weighted supervised learning problem, where Y_i^* and w_i^* are the new outcome and new weight of each sample. The nuisance quantities in Y_i^* and w_i^* need to be pre-estimated and plugged in. Here we use L_1-based gradient boosting machine (GBM) with $Y|T = -1$, $Y|T = 1$, Y as outcomes to estimate $\mu^{(-1)}(\mathbf{x})$, $\mu^{(1)}(\mathbf{x})$, and $\mu(\mathbf{x})$. Note that $\mu^{(1)}(\mathbf{x})$ and $\mu^{(-1)}(\mathbf{x})$ are only needed for AIPW. And we use L_2-based GBM with $D = (T + 1)/2$ to estimate $p(\mathbf{x})$. Any supervised learning algorithm with a weighted L_1 loss can be used to optimize (4) for robust CATE estimation. In this section, we describe three different algorithms for this purpose. The algorithms we describe are based on Random Forest (RF), GBM, and artificial neural network (ANN). The common process underlying these algorithms is graphically depicted in the following figure.



To achieve robust estimates of τ, we modified the existing supervised learning algorithms by incorporating the L_2-loss function. For example in RF, we used a weighted LAD splitting rule and the mean-of-medians to aggregate the trees, as opposed to the L_2-loss function and mean-of-means in the standard RF. Similarly in GBM, we used the L_1-loss to compute the working response and we calculated the weighted medians for prediction of the terminal nodes. In ANN, we used weighted LAD in back-propagation, and an L_1 regularization in high-dimensional situations to ascertain the sparse weights; here we used the adaptive moment estimation (Adam) to avoid being stuck at a local optimum (Kingma and Ba, 2014). We describe the algorithmic details in the following subsections.

2.4.1 A Robust Random Forest Learner

We first use RF for robust estimation of CATE. The building blocks of random forests are regression trees (Breiman et al., 1984). The tree structure comes from the recursively partitioning of the sample by covariates to minimize heterogeneity in the outcomes. The partition that minimizes the heterogeneity in child nodes is chosen, so that variables reducing heterogeneity most have the best chance of being selected than the background noise variables (Biau, 2012). Binary splits lead to trees, and then aggregated results within the terminal nodes are used for prediction. The random forest creates a more stable structure and reduces the variance by combining a large number of de-correlated regression trees (Breiman, 2001).

The standard regression trees minimize the mean squared error (MSE) in child nodes (i.e., $MSE = \sum_{i \in L_l}(y_i - \bar{y_l})^2 + \sum_{i \in L_r}(y_i - \bar{y_r})^2$, where $\bar{y_l}$ and $\bar{y_r}$ are the average values within the left and right child nodes) (Hastie et al., 2009). And robust random forests for regression have been studied to gain robustness against outliers, including using mean-of-medians (Meinshausen and Ridgeway, 2006) or median-of-means as estimators, and the LAD-based splitting rule (Roy and Larocque, 2012). Empirical studies have demonstrated that these modifications offer more protection against outliers than the standard RF.

The robust RF-based CATE estimation splits the samples by using the weighted LAD (WLAD) rule, a variant of the LAD rule. The WLAD rule is

$$WLAD = \sum_{i \in L_l} w_i^* |y_i^* - \tilde{y_l^*}'| + \sum_{i \in L_r} w_i^* |y_i^* - \tilde{y_r^*}'|, \tag{5}$$

where $\tilde{y_l^*}'$ and $\tilde{y_r^*}'$ are the leaf node medians to increase robustness and w_i^* is the transformed weight of each observation. For prediction, we use the mean-of-medians that is consistant with the WLAD rule (Meinshausen and Ridgeway, 2006) instead of the median of means as advocated by Roy and Larocque (2012).

Algorithm 1: Robust RF-based CATE estimating algorithm

Input: Data $\{(Y_i, T_i, \mathbf{X}_i)\}_{i=1}^n$, number of trees T, fraction of features used in splitting $p_{fraction} \in (0, 1)$, minimum node size k, and bootstrap sample size N.

Estimate nuisance quantities $p(\mathbf{x}), \mu(\mathbf{x}), \mu^{(1)}(\mathbf{x}), \mu^{(-1)}(\mathbf{x})$ using (robust) GBM;

Calculate w_i^* and y_i^* according to Table 2 and Formulation (4);

for *t in 1,...,T* **do**

a. Randomly select N observations with replacement from the dataset as the bootstrap sample and randomly select a subset of variables with size $p_{fraction} \times p$;

b. Fit a regression tree by repeating following steps until we reach the minimum node size k:

b.1 Find the variable and the cutoff value that best split the data into two child nodes based on (5);

b.2 Split the current node into two child nodes;

c. Calculate the median of the transformed outcomes in each terminal node as CATE estimator;

end

Output: Mean-of-medians as the final CATE estimation $\hat{\tau}(\mathbf{x})$ and splitting criterion of trees.

The tuning parameters T, $p_{fraction}$, k, and N can be selected by cross validation.

2.4.2 The robust gradient boosting machine learner

Gradient boosting machine is a supervised learning technique that produces a prediction model $\hat{f}(\mathbf{x})$ in the form of sequential weak-learners, typically regression trees, so that it performs better in high-dimensional settings (Friedman et al., 2000; Friedman, 2001, 2002). GBM builds the model in a step-wise fashion by allowing optimization of a differentiable loss function $\Psi(y, f)$. The principle idea behind this algorithm is to construct weak-learners that are maximally correlated with the negative gradient of the loss function, associated with the whole ensemble.

Friedman's GBM algorithm initializes $\hat{f}(\mathbf{x})$ to be a constant. Then, in each iteration, it computes the negative gradient as the working response

$$z_i = -\frac{\partial}{\partial f(\mathbf{x}_i)} \Psi(y_i, f(\mathbf{x}_i)) \Big|_{f(\mathbf{x}_i)=\hat{f}(\mathbf{x}_i)}.$$

A regression model $g(\mathbf{x})$ is fitted to predict z from the covariates $\mathbf{x}$. Finally, it updates the estimate of $f(\mathbf{x})$ as $\hat{f}(\mathbf{x}) \leftarrow \hat{f}(\mathbf{x}) + \lambda g(\mathbf{x})$, where λ is the step size. Friedman also proposed the LAD-TreeBoost algorithm (Friedman, 2001), a variation of GBM, which is highly robust against outliers. Ridgeway (2007) later extended the LAD-TreeBoost algorithm to a weighted version.

In the proposed robust GBM for CATE estimation, we further extended Ridgeway's algorithm by combining it with the unified CATE estimation formulation as follows:

Algorithm 2: Robust GBM-based CATE estimating algorithm

Input: Data $\{(Y_i, T_i, \mathbf{X}_i)\}_{i=1}^n$, number of trees T, fraction of observations used in splitting $p_{sample} \in (0, 1)$, interaction depth c, and step size λ.

Estimate nuisance quantities $p(\mathbf{x}), \mu(\mathbf{x}), \mu^{(1)}(\mathbf{x}), \mu^{(-1)}(\mathbf{x})$ using (robust) GBM;

Calculate w_i^* and y_i^* according to Table 2 and Formulation (4);

Initialize $\hat{\tau}(\mathbf{x})$ to be a constant, $\hat{\tau}(\mathbf{x}) = median_{w^*}(y^*)$;

for *t in 1,...,T* **do**

a. Compute the negative gradient as the working response $z_i = -sign(y_i^* - \hat{\tau}(\mathbf{x}_i))$;

b. Randomly select $p_{sample} \times n$ observations without replacement from the dataset;

c. Fit a regression tree to predict z_i using covariates $\mathbf{x}_i$ with interaction depth c and the number of leaf nodes K;

d. Compute the optimal predictions for feature $\mathbf{x}$ as $\rho_k(\mathbf{x}) = argmin_\rho \sum_{\mathbf{x}_i \in S_k} \Psi(y_i^*, \hat{\tau}(\mathbf{x}_i) + \rho, w_i^*)$, where $\Psi(y, x, w) = w|y - x|$ and k indicates the index of the terminal node S_k into which an observation with feature x would fall;

e. Update $\hat{\tau}(\mathbf{x})$ as $\hat{\tau}(\mathbf{x}) \leftarrow \hat{\tau}(\mathbf{x}) + \lambda\rho_k(\mathbf{x})$, where λ is step size.

end

Output: Splitting criterion and CATE estimates as the resulted $\hat{\tau}(\mathbf{x})$ from the above iteration.

For robust estimation, the terminal node estimate is the weighted median $median_{w^*}(z)$, defined as the solution ρ to the equation $\frac{\sum w_i^* I(y_i^* \leq \rho)}{\sum w_i^*} = \frac{1}{2}$. Tuning parameters T, λ, c, and K can be selected via cross validation.

2.4.3 A robust artificial neural network learner

Artificial neural network (ANN) is a computer program designed to simulate the way the human brain processes information (Goodfellow et al., 2016). A no-hidden-layer ANN with identity activation function is similar to linear regression in its modeling structure. But an ANN with multiple hidden layers offers more enhanced modeling flexibility. A feed-forward neural network with two hidden layers can be written as $g(\mathbf{x}) := f^3(W^3 f^2(W^2 f^1(W^1\mathbf{x})))$, where $W^l = (w^l_{jk})$ are the weights between layer $l-1$ and l, and w^l_{jk} is the weight between the k-th node in layer $l-1$ and the j-th node in layer l, and f^l is the activation function at layer l.

Multi-layer networks use a variety of techniques to learn the weights. The most popular approach is backpropagation (Rumelhart et al., 1986). In training, the loss of the model is defined based on the difference between the outcome y and the predicted output $\hat{y}$. The most popular loss function is the Root Mean Squared Error (RMSE) (i.e., $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$). However, numerous studies have shown that the presence of outliers poses a serious threat to the standard least squares analysis (Liano, 1996). The L_1-loss provides an effective remedy that can be applied to ANN (i.e., $\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i|$). An empirical study shows that L_1-based estimator had an improved performance than that of the L_2-based algorithm when outliers exist (El-Melegy et al., 2009).

As typical for CATE estimation, the activation functions of the hidden layers are rectified linear activation unis (ReLUs) and the last activation function is the identity function (Nair and Hinton, 2010). ReLU is a piecewise linear function that outputs the input directly if it is positive; otherwise, it outputs zero. Models that uses ReLUs are easier to train and often have better performance.

To ensure robustness, we propose to use the weighted Mean Absolute Error (MAE) $\frac{1}{n}\sum_{i=1}^{n} w_i^* |y_i^* - \hat{y}_i^*|$ as the loss function, where w^* and y^* are the transformed weight and outcome in the unified formulation (4). We use the adaptive moment estimation (Adam), a gradient-based optimization algorithm, which runs averages of both the gradients and the second moments of the gradients (Kingma and Ba, 2014), to train the ANN. We add an L_1 regularization term $\lambda\|W\|_1$ to the loss function in high-dimensional settings in the first layer to achieve sparsity by driving some weights to zero (Feng and Simon, 2017; Girosi et al., 1995), where λ is the tuning parameter.

The algorithm is as follows:

Algorithm 3: Robust ANN-based CATE estimating algorithm

Input: Data $\{(Y_i, T_i, \mathbf{X}_i)\}_{i=1}^n$, number of iterations T, batch size $\mathcal{B}$, Adam parameters β_1, β_2, η, and ε, and L_1 regularization parameter λ in high-dimensional case.

Estimate nuisance quantities $p(\mathbf{x}), \mu(\mathbf{x}), \mu^{(1)}(\mathbf{x}), \mu^{(-1)}(\mathbf{x})$ using (robust) GBM;

Calculate w_i^* and y_i^* according to Table 2 and Formulation (4);

Initialize an ANN with weights W, the decaying average of past gradients m to a zero vector, and the decaying average of past squared gradients v to a zero vector;

for *t in 1,...,T* **do**

a. Sample a mini-batch of data $\{y_i^*, \mathbf{x}_i, w_i^*\}$ without replacement with size $\mathcal{B}$;

b. Compute the negative gradients $g^{(t)}$ based on weighted MAE;

c. Update m and v by $m^{(t)} = \beta_1 m^{(t-1)} + (1-\beta_1)g^{(t)}, v^{(t)} = \beta_2 v^{(t-1)} + (1-\beta_2){g^{(t)}}^2$;

d. Compute bias correction terms $\hat{m}^{(t)} = \frac{m^{(t)}}{1-\beta_1^t}, \hat{v}^{(t)} = \frac{v^{(t)}}{1-\beta_2^t}$;

e. Update the weights by $W^{(t)} = W^{(t-1)} - \eta\frac{\hat{m}^{(t)}}{\sqrt{\hat{v}^{(t)}}+\varepsilon}$.

end

Output: Weights W in the ANN and the resulted $\hat{\tau}(\mathbf{x})$ represented by the network.

Key advantages of the algorithm-based CATE estimators, in comparison with their model-based counterparts, are their automated implementation and scalability, as well as their accommodation of the non-additive effects and the high-dimensionality of X. For different algorithm-based CATE learners, we summarize the advantages and disadvantages in Table 3. Generally speaking, RF is easier to tune and it performs well in low dimensional cases. But a well-tuned GBM tends to outperforms RF in a high-dimensional data situation. ANN usually outperforms GBM and RF for image and text data because ANN is more flexible. For CATE estimation, however, when we have structured non-image or non-text data, the representation problem is easier to solve, and ANN might not offer added advantages.

(Table 3 goes here)

2.4.4 An R package for implementation

To make the proposed algorithms more accessible, we implemented the three CATE-learning algorithms in an **R** package **RCATE**. Each of the algorithms can be combined with MCM-EA, RL, and AIPW to achieve robust CATE estimation. For input data, we only require specification of the outcome, treatment assignment, and pre-treatment covariates. There is no need for users to estimate the nuisance quantities. A more detailed description of the **R** package **RCATE** and example code are provided in Appendix A.

3 Simulation Studies

3.1 Design and implementation

We conducted three sets of simulations to evaluate the performance of the proposed methods. **Simulation Study 1**: We compared the additive-model-based and algorithm-based learners under both L_1 and L_2 loss functions when the true treatment effect model involved interactions, i.e., non-additive.

Simulation Study 2: We compared the proposed L_1-based algorithms with other machine learning algorithms in high-dimensional settings.

Supplemental Simulation Study (S): We compared the algorithm-based robust estimators against model-based ones when the true treatment effect models were indeed additive; see details in Appendix B.

The methods considered in each of the three simulation studies are described in Table 4, where the numbers in the parentheses indicate the specific simulation studies.

(Table 4 goes here)

We designed the simulation settings followed the structure of the real data in Section 4. The binary treatment levels (i.e., $T \in \{-1, 1\}$) and continuous outcome were used throughout. And we set the number of replications to $R = 1,000$ and the size of the validation set to $n_\nu = 1,000$.

We assessed the performance of these methods using mean squared error (MSE), mean absolute error (MAE), and coverage probability (CP). The MSE and MAE were defined as follows:

$$MAE_v = \frac{1}{R}\sum_{r=1}^{R} |\hat{\tau}^{(r)}(\mathbf{x}_v) - \tau_0(\mathbf{x}_v)|, \quad MSE_v = \frac{1}{R}\sum_{r=1}^{R} [\hat{\tau}^{(r)}(\mathbf{x}_v) - \tau_0(\mathbf{x}_v)]^2,$$

where $\mathbf{x}_v$ is the v-th observation from the validation set, $\hat{\tau}^{(r)}(\mathbf{x})$ is the estimator of $\tau(\mathbf{x})$ based on the r-th data replicate. We summarized the performance over the whole validation set by taking the average (i.e.,$\overline{MSE} = \frac{1}{n_v}\sum_{v=1}^{n_v} MSE_v$). For simplicity, we reported MSE and MAE.

We calculated the CP as the proportion of the times that 95% bootstrap percentile intervals contained the true value of interest, out of the total number of simulating iterations ($R = 1,000$), i.e.,

$$CP = \frac{1}{R}\sum_{i=1}^{R} I(\text{C.I. covers the true value}),$$

The tuning parameters were summarized in Appendix B.

3.1.1 Simulation 1: ML vs. model-based methods when τ_0 is not additive

We generated the outcome from the following model

$$Y_i = b_0(\mathbf{X}_i) + \frac{T_i}{2}\tau_0(\mathbf{X}_i) + \varepsilon_i, \quad \varepsilon_i \sim (1-p_o)N(0,1) + p_o P.$$

We used two different error distributions $P = N(0,100)$ and $P = Laplace(0, \sqrt{50})$. The covariates were continuous variables ($\mathbf{X}_i \sim N_{10}(0,1)$). The treatment assignment followed a logistic model

$$D_i|\mathbf{X}_i \sim Bernoulli(p(\mathbf{X}_i)), \quad T_i = 2D_i - 1, \quad logit(p(\mathbf{X}_i)) = X_{i1} - X_{i2}.$$

Functions $b_0(\mathbf{X}_i)$ and $\tau_0(\mathbf{X}_i)$ in the response surface were

$$b_0(\mathbf{X}_i) = 100 + 4X_{i1} + X_{i2} - 3X_{i3},$$

$$\tau_0(\mathbf{X}_i) = 6sin(2X_{i1}) + 3(X_{i2}+3)X_{i3} + 9tanh(0.5X_{i4}) + 3X_{i5}(2I(X_{i4}) - 1),$$

where the true treatment effect function included an interaction term, and thus violating the additive model assumption.

We compared all methods indicated by "(1)" in Table 4 while altering two design factors: The proportions of outliers p_o and the outlier generating mechanisms: (1) $p_o \in \{0, 0.05, 0.1, 0.15, 0.2, 0.3, 0.5\}$, $n = 1000$, and $P = N(0,100)$, and (2) $p_o \in \{0, 0.05, 0.1, 0.15, 0.2, 0.3, 0.5\}$, $n = 1000$, and $P = Laplace(0, \sqrt{50})$.

(Figure 1 goes here)

We reported the MSE and MAE of the CATE estimators graphically in Figure 1. The figure showed that all L_1-based algorithms outperformed the L_2-based ones. Advantage of the robust algorithms, as measured by MSE and MAE, increased with the proportion of outliers. Because the true treatment effect function was non-additive, when $p_o < 0.2$, the proposed machine learning algorithms outperformed additive models in MSE and CP; CPs were summarized in tabular form in Appendix Table B.3. The performance of robust GAMs was better than robust QL when the proportion of outliers was close to the breakdown point of LAD regression, i.e., $p_o = 0.5$.

There were little practical differences among the robust GBM, robust ANN and robust RF when combined with MCM-EA and R-learning. But the robust GBM didn't work well together with AIPW transformation because AIPW tended to generate transformed weights with a large variability, and GBM was more likely to overfit when the data were noisy (Park and Ho, 2019).

3.1.2 Simulation 2: Performance in high-dimensional settings

Here, we only considered the methods that performed well in Simulation Study 1, and we focused on the methods' performance in high-dimensional settings and when outliers existed.

We generated data sets with the same outlier distributions P, baseline function, and propensity score function as in Simulation Study 1. And we fixed the proportion of outliers at 0.15, sample size at $n = 1,000$, and the data dimension at $p \in \{100, 2000\}$.

The true treatment effect functions when $p = 100$ and $p = 2000$ were

$$\begin{aligned}\tau_0(\mathbf{X}_i) =& 6sin(2X_{i1}) + 3(X_{i2} + 3)X_{i3} + 9tanh(0.5X_{i4}) + 3X_{i5}(2I(X_{i4}) - 1)+ \\ & 3X_{i6} + 2X_{i7} + X_{i8} - 2X_{i9} - 4X_{i10},\end{aligned}$$

and

$$\begin{aligned}\tau_0(\mathbf{X}_i) =& 6sin(2X_{i1}) + 3(X_{i2} + 3)X_{i3} + 9tanh(0.5X_{i4}) + 3X_{i5}(2I(X_{i4}) - 1)+ \\ & \sum_{j=6}^{50} \beta_j X_{ij}, \beta_j \sim Unif(-2, 2),\end{aligned}.$$

Figure 2 (A) and (C) showed that when $p = 100$, the robust GBM and robust ANN combined with AIPW and MCM-EA outperformed all other methods when outliers exist. Among the existing algorithms, causal MARS had the best performance. The performance of robust RF and robust ANN combined with RL tied with that of the causal MARS. The boosting algorithms generally performed better than RFs, because a single deep tree tended to struggle to reduce bias on high dimensional data, so did the forests. When we increased the dimension to $p = 2000$ Figure 2 (B) and (D) showed that the robust GBMs had the best performance when the data dimension was much larger than the sample size.

(Figure 2 goes here)

We additionally compared the computational speed of the proposed algorithms and additive models under difference sample sizes and dimensions of data. The robust RF was implemented in **R**, so that the speed was relatively slow and was not included in the comparisons here. The CPU time was collected on a personal computer with Intel Core i7-7700 CPU @3.60Ghz and 32 GB RAM. Table 5 showed that the robust GBM was the most efficient algorithms among all those considered in the comparison. Its advantage was most prominent when the sample size or dimension was high.

(Table 5 goes here)

4 Real data application

To illustrate the use of the proposed algorithms, we assessed the treatment effects of two different antihypertensive therapies by analyzing recorded clinical data set from the "All of Us" research program. Sponsored by NIH, the program collected research data from multiple sources, including health surveys, health records, and digital health technologies (All of Us Research Program Investigators, 2019). Research data are publicly accessible at `https://workbench.researchallofus.org/` through web-based Jupyter Notebook.

In this analysis, we compared the monotherapeutic effects of angiotensin-converting-enzyme inhibitors (ACEI) and thiazide diuretics on systolic blood pressure (SBP). We considered those receiving thiazide as in treatment group A ($n = 504$), and those receiving ACEI as in group B ($n = 1040$). The primary outcome of interest is the clinically recorded SBP in response to these therapies. Covariates of interest included the demographic and clinical characteristics of the participants; see Table 6.

We expressed the treatment effect as a function of the patient characteristics $\mathbf{x}$

$$\tau_0(\mathbf{x}) = E[Y^{(B)} - Y^{(A)}|\mathbf{X} = \mathbf{x}],$$

where $Y^{(A)}$ and $Y^{(B)}$ represented the potential outcome of the two treatment groups. Since the treatment effect of a therapy is measured by its ability to lower SBP, a positive $\hat{\tau}(\mathbf{x})$ indicates a superior effect of the thiazide diuretics, for a given $\mathbf{x}$. An important covariate is the baseline SBP.

In this analysis, we included individuals that were only on thiazide diuretic or ACEI for at least a month. Their first SBP within three months after the initiation of thiazide or ACEI was used as the outcome. The pre-treatment characteristics were measured within three months before the initiation of thiazide or ACEI, and they were presented in Table 6. Missing lab values were imputed by multiple imputation (Rubin, 2004).

(Table 6 goes here)

Preliminary data examination showed that the observed outcome was right-skewed. See Figure 3. The Shapiro–Wilk's test confirmed that the SBP was not normally distributed (thiazide diuretic: $W = 0.9739$, $p = 8.011e - 08$; ACEI: $W = 0.9763$, $p = 5.422e - 12$). We, therefore, used the L_1-based algorithms to analyze the data. Here the weighted supervised learning algorithms were used to accommodate the possible complex treatment effect function.

(Figure 3 goes here)

A closer examination of the patient characteristics revealed that patients on thiazide had higher sodium and high density lipid (HDL) levels, lower albumin level and glomerular filtration rate (GFR), and more likely to be female. Using GBM, we examined the mean function of SBP $\hat{\mu}(\mathbf{x})$, $\hat{\mu}^{(1)}(\mathbf{x})$, $\hat{\mu}^{(-1)}(\mathbf{x})$ and the propensity of patient receiving ACEI $\hat{p}(\mathbf{x})$. The estimated propensity score distributions were clearly different for the two treatment groups, whereas the mean functions were similar. See Figure 4. The different propensity score distributions of the two groups clearly showed the non-random nature of treatment assignment, and that a naive comparison should not be trusted.

(Figure 4 goes here)

We then analyzed the data with the proposed algorithms: the robust RF and robust GBM combined with MCM-EA and R-learning. We use these four methods to estimate the CATE. Estimated treatment effects conditioning on pre-treatment SBP were shown graphically in Figure 5. To plot these marginal effects, we fixed the continuous covariates at their mean values, and categorical covariates at their mode levels.

Results showed that the SBP lowering effects of thiazide diuretics and ACEI were similar when the pre-treatment SBP were below 160 mmHg. But for individuals with baseline SBP greater than 160 mmHg, diuretics tended to have a stronger SBP-lowering effect. This observation was largely consistent with the findings of the Antihypertensive and Lipid-Lowering Treatment to Prevent Heart Attack Trial (ALLHAT), which showed a comparable effect of thiazide-like diuretic chlorthalidone and ACEI lisinopril (The ALLHAT Officers and Coordinators for the ALLHAT Collaborative Research Group, 2002). Diuretics reduce blood pressure through their natriuretic actions – increase urinary excretion of sodium and reduce extracellular fluid volume (ECFV). It works particularly well in patients with greatly expanded ECFV, and thus explaining the greater SBP reduction in patients with higher pre-treatment SBP (Duarte and Cooper-DeHoff, 2010).

(Figure 5 goes here)

To verify the conditional independence error assumption, we performed the invariant residual distribution test (IRD-test), invariant environment prediction test (IEP-test), invariant conditional quantile prediction test (ICQP-test), invariant targeted prediction test (ITP-test) (Heinze-Deml et al., 2018). The conditional independence error assumption held for both proposed methods at the significant level of 0.05.

(Table 7 goes here)

5 Discussion

The practice of precision medicine relies on a sound understanding of the causal effects of specific treatments in patients with different characteristics. By expressing the treatment effect as a function of patient characteristics, the heterogeneous treatment effect provides a useful quantification of the unknown causal effect. Among the existing methods for estimating heterogeneous treatment effects, few have considered the conditions of the data from which the estimates are derived - outliers and other forms of data irregularities could severely undermine the validity of the causal estimation. We described a general estimating equation that produces robust estimates against such data irregularities in recent work. However, the method requires the correct specification of the treatment effect function. From a practical perspective, such a requirement represents a significant constraint. Even when flexible additive models are used to accommodate the potential nonlinear effects, there is no assurance that such an additive structure would be adequate. To address this issue, we introduced a set of modified machine learning algorithms for treatment effect estimation. We also presented the necessary computational tools for practical data analysis.

When implemented within the framework of the previously proposed estimating equation for heterogeneous causal effects, we show that supervised learning algorithms could significantly reduce the risk of model misspecification without losing the method's robustness. In a sense, the work presents a data-driven analytical approach that reduces the users' burden of model specification while retaining good theoretical properties of the general estimating equation. A critical ingredient of this approach is the use of machine learning techniques to optimize the objective function. Simulation results confirmed that the new procedures' good performance. As a result of this development, we improved the general estimating equation's scalability in real data applications, making the methods more readily usable in practical data analysis.

6 Acknowledgement

WT was partially supported by grants RO1 HL095086, RO1 AA025208, U24 AA026969 from the National Institutes of Health. JS was partially supported by the Indiana University Precision Health Initiative. The All of Us Research Program is supported by the National Institutes of Health, Office of the Director: Regional Medical Centers: 1 OT2 OD026549; 1 OT2 OD026554; 1 OT2 OD026557; 1 OT2 OD026556; 1 OT2 OD026550; 1 OT2 OD 026552; 1 OT2 OD026553; 1 OT2 OD026548; 1 OT2 OD026551; 1 OT2 OD026555; IAA: AOD 16037; Federally Qualified Health Centers: HHSN 263201600085U; Data and Research Center: 5 U2C OD023196; Biobank: 1 U24 OD023121; The Participant Center: U24 OD023176; Participant Technology Systems Center: 1 U24 OD023163; Communications and Engagement: 3 OT2 OD023205; 3 OT2 OD023206; and Community Partners: 1 OT2 OD025277; 3 OT2 OD025315; 1 OT2 OD025337; 1 OT2 OD025276. In addition, the All of Us Research Program would not be possible without the partnership of its participants.

7 Disclosure

None.

References

All of Us Research Program Investigators (2019). The "all of us" research program. *New England Journal of Medicine 381*(7), 668–676.

Athey, S. and G. Imbens (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences 113*(27), 7353–7360.

Athey, S., J. Tibshirani, S. Wager, et al. (2019). Generalized random forests. *The Annals of Statistics 47*(2), 1148–1178.

Biau, G. (2012). Analysis of a random forests model. *The Journal of Machine Learning Research 13*(1), 1063–1095.

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

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984). *Classification and regression trees*. CRC press.

Caron, A., I. Manolopoulou, and G. Baio (2020). Estimating individual treatment effects using non-parametric regression models: a review. *arXiv preprint arXiv:2009.06472*.

Chen, S., L. Tian, T. Cai, and M. Yu (2017). A general statistical framework for subgroup identification and comparative treatment scoring. *Biometrics 73*(4), 1199–1209.

Duarte, J. D. and R. M. Cooper-DeHoff (2010). Mechanisms for blood pressure lowering and metabolic effects of thiazide and thiazide-like diuretics. *Expert review of cardiovascular therapy 8*(6), 793–802.

El-Melegy, M. T., M. H. Essai, and A. A. Ali (2009). Robust training of artificial feedforward neural networks. In *Foundations of Computational, Intelligence Volume 1*, pp. 217–242. Springer.

Feng, J. and N. Simon (2017). Sparse-input neural networks for high-dimensional nonparametric regression and classification. *arXiv preprint arXiv:1711.07592*.

Friedman, J., T. Hastie, R. Tibshirani, et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics 28*(2), 337–407.

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.

Girosi, F., M. Jones, and T. Poggio (1995). Regularization theory and neural networks architectures. *Neural computation 7*(2), 219–269.

Goodfellow, I., Y. Bengio, A. Courville, and Y. Bengio (2016). *Deep learning*, Volume 1. MIT press Cambridge.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media.

Heinze-Deml, C., J. Peters, and N. Meinshausen (2018). Invariant causal prediction for nonlinear models. *Journal of Causal Inference 6*(2).

Hirano, K., G. W. Imbens, and G. Ridder (2003). Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica 71*(4), 1161–1189.

Horvitz, D. G. and D. J. Thompson (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association 47*(260), 663–685.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Künzel, S. R., J. S. Sekhon, P. J. Bickel, and B. Yu (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences 116*(10), 4156–4165.

Li, R., H. Wang, and W. Tu (2021). Robust estimation of heterogeneous treatment effects using electronic health record data. *Statistics in Medicine 40*(11), 2713–2752. Also accessible at https://arxiv.org/abs/2105.03325.

Liano, K. (1996). Robust error measure for supervised neural network learning with outliers. *IEEE Transactions on Neural Networks 7*(1), 246–250.

Meinshausen, N. and G. Ridgeway (2006). Quantile regression forests. *Journal of Machine Learning Research 7*(6).

Murphy, S. A. (2003). Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 65*(2), 331–355.

Nair, V. and G. E. Hinton (2010). Rectified linear units improve restricted boltzmann machines. In *Icml.*

Nie, X. and S. Wager (2017). Quasi-oracle estimation of heterogeneous treatment effects. *arXiv preprint arXiv:1712.04912*.

Park, Y. and J. Ho (2019). Tackling overfitting in boosting for noisy healthcare data. *IEEE Transactions on Knowledge and Data Engineering*, 1–1.

Powers, S., J. Qian, K. Jung, A. Schuler, N. H. Shah, T. Hastie, and R. Tibshirani (2018). Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in medicine 37*(11), 1767–1787.

Ridgeway, G. (2007). Generalized boosted models: A guide to the gbm package. *Update 1*(1), 2007.

Robins, J. M. (2004). Optimal structural nested models for optimal sequential decisions. In *Proceedings of the second seattle Symposium in Biostatistics*, pp. 189–326. Springer.

Robins, J. M. and A. Rotnitzky (1995). Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association 90*(429), 122–129.

Roy, M.-H. and D. Larocque (2012). Robustness of random forests for regression. *Journal of Nonparametric Statistics 24*(4), 993–1006.

Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys*, Volume 81. John Wiley & Sons.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *nature 323*(6088), 533–536.

Sekhon, J. S. (2008). The neyman-rubin model of causal inference and estimation via matching methods. *The Oxford handbook of political methodology 2*, 1–32.

The ALLHAT Officers and Coordinators for the ALLHAT Collaborative Research Group (2002, 12). Major Outcomes in High-Risk Hypertensive Patients Randomized to Angiotensin-Converting Enzyme Inhibitor or Calcium Channel Blocker vs DiureticThe Antihypertensive and Lipid-Lowering Treatment to Prevent Heart Attack Trial (ALLHAT). *JAMA 288*(23), 2981–2997.

Tian, L., A. A. Alizadeh, A. J. Gentles, and R. Tibshirani (2014). A simple method for estimating interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association 109*(508), 1517–1532.

Watkins, C. J. and P. Dayan (1992). Q-learning. *Machine learning 8*(3-4), 279–292.

Watkins, C. J. C. H. (1989). Learning from delayed rewards.

Xiao, W., H. H. Zhang, and W. Lu (2019). Robust regression for optimal individualized treatment rules. *Statistics in medicine 38*(11), 2059–2073.

Zhang, W., J. Li, and L. Liu (2020). A unified survey on treatment effect heterogeneity modeling and uplift modeling. *arXiv preprint arXiv:2007.12769*.

Table 1: Summary of existing popular CATE estimation algorithms

Base-learner/ Algorithm	Description	Pros(+) and Cons(-)	Available **R** packages
The single-learner (or S-learner)	Fits a single-model for the outcome with the covariates and treatment assignment indicator.	(+) If the treatment effect is simple, then pooling the data together will be beneficial. (−) Performs bad if the treatment effect is strongly heterogeneous and the response surfaces of two groups are very different.	**rlearner** **causalToolbox**
The two-learner (or T-learner)	Fits two models for the outcome of two treatment groups separately with the covariates.	(+) Performs well if the treatment effect is strongly heterogeneous and the response surfaces of two groups are very different. (−) Uses the data inefficiently.	**rlearner** **causalToolbox**
The X-learner (Künzel et al., 2019)	A three step approach to crossover the information in the control and treated subjects.	(+) Has the advantages of both S and T-learner. (−) The three-step estimator increases the risk of over-fitting and the difficulty in tuning parameter.	**rlearner** **causalToolbox**
Inverse propensity score weighting (IPW)	Transforms the outcome by inverse propensity score weighting, then the conditional expectation of the transformed outcome is the treatment effect.	(+) After transformation, the IPW provides the flexibility in choosing off-the-shelf supervised learning algorithms. (−) Relies on the accurate estimation of the propensity score.	
Augmented inverse propensity score weighting (AIPW)	Augmented IPW is robust to mis-specified mean or propensity score model.	(+) In addition to the advantage of IPW, AIPW has the property of double robustness.	**RCATE**
The R-learner (RL)	Decomposes the outcome by subtracting the mean model and gets an estimating equation.	(+) In addition to the advantage of IPW, R-learner has quasi-oracle property.	**rlearner** **RCATE**
The modified covariate method with efficiency augmentation (MCM-EA)	Transforms the covariates to get an estimating equation.	(+) Same as IPW. (−) Relies on the accurate estimation of mean and propensity score.	**RCATE**
The Q-learner	Fits the interaction model and the slope is the treatment effect function.	(+) No nuisance parameter need to be estimated. (−) Lacks of flexibility in algorithm choosing and sensitive to model mis-specification.	
Causal tree (Athey and Imbens, 2016)	Uses regression tree that splits by maximizing the difference between treatment effects in child nodes to fit the outcome.	(+) Easy to interpret and provides the grouping of subjects. (−) Suffers from the problem of high variance.	**causalTree**

Causal forest (Athey et al., 2019)	Uses randomly selected subsample and covariates to build causal trees, then aggregate the results.	(+) Addresses the high variance problem. (−) Lose the interpretability.	**grf**
Causal boosting (Powers et al., 2018)	An adaption of gradient boosting algorithm with causal trees as weak-learner.	(+) Well-tuned causal boosting outperforms the causal forest. (−) Takes longer to train than causal forest and could overfit the training data.	**causalLearning**
Causal MARS (Powers et al., 2018)	Fits two multivariate adaptive regression spline models in parallel in two arms of the data. In each step, it chooses the same basis function to add to each model.	(+) Alleviates the bias problem of tree-based algorithms because they use the average treatment effect within each leaf as the prediction for that leaf.	**causalLearning**

Table 2: Parameters of some popular methods in the framework

Method	$w(\mathbf{X}_i, T_i)$	$g(\mathbf{X}_i)$	$c(\mathbf{X}_i, T_i)$
MCM	$\{T_i p(\mathbf{X}_i) + (1-T_i)/2\}^{-1}$	0	$\frac{T_i}{2}$
MCM-EA	$\{T_i p(\mathbf{X}_i) + (1-T_i)/2\}^{-1}$	$\mu(\mathbf{X}_i)$	$\frac{T_i}{2}$
RL	1	$\mu(\mathbf{X}_i)$	$\{T_i - 2p(\mathbf{X}_i) + 1\}/2$
IPW	$\left\{\frac{T_i - 2p(\mathbf{X}_i)+1}{2p(\mathbf{X}_i)(1-p(\mathbf{X}_i))}\right\}^2$	0	$\frac{2p(\mathbf{X}_i)(1-p(\mathbf{X}_i))}{T_i - 2p(\mathbf{X}_i)+1}$
AIPW	$\left\{\frac{T_i - 2p(\mathbf{X}_i)+1}{2p(\mathbf{X}_i)(1-p(\mathbf{X}_i))}\right\}^2$	$(1-p(\mathbf{X}_i))\mu_1(\mathbf{X}_i) + p(\mathbf{X}_i)\mu_{-1}(\mathbf{X}_i)$	$\frac{2p(\mathbf{X}_i)(1-p(\mathbf{X}_i))}{T_i - 2p(\mathbf{X}_i)+1}$

Table 3: Supervised learning algorithms for CATE estimation

Algorithm	Advantages	Disadvantages	Main Hyperparameters
Random Forests	Hard to overfit, easy to tune, good for parallel computing	Model can get large	Number of trees, number of features used in splitting
GBM	High-performing in high-dimensional case	Harder to tune than RF, take longer to train than RF	Number of trees, depth of trees, learning rate
Neural Network	Can handle extremely complex task	Hard and slow to train	Number of neurons in the hidden layer, number of epochs, learning rate

Table 4: Methods considered in the simulation studies. Numbers in the parentheses indicate the specific simulation studies in which the methods were assessed.

Methods under the Unified Formulation				Other Candidate Methods	
	MCM-EA	RL	AIPW	Method	
Robust RF	(1)(2)(S)	(1)(2)(S)	(1)(2)(S)	Robust QL	(1)
Robust GBM	(1)(2)(S)	(1)(2)(S)	(1)(2)(S)	Causal BART	(2)
Robust ANN	(1)(2)(S)	(1)(2)(S)	(1)(2)(S)	Causal Boosting	(2)
RF	(1)	(1)	(1)	Causal Forest	(2)
GBM	(1)	(1)	(1)	Causal MARS	(2)
ANN	(1)	(1)	(1)	X-learner+RF	(2)
Robust GAM	(1)(S)	(1)(S)	(1)(S)		

Table 5: Comparison of the CPU time (s) of RF/GBM/ANN and additive model

Dimension	Algorithm	$n = 1000$	$n = 3000$	$n = 5000$	$n = 8000$
$p = 10$	Random Forests	0.30	1.67	3.34	7.41
	GBM	0.28	0.79	1.29	2.13
	Robust GBM	0.29	0.99	1.63	2.58
	ANN	4.72	12.87	21.43	35.89
	Robust ANN	4.51	12.63	20.90	35.25
	Robust GAM	1.65	18.94	38.23	86.18
$p = 100$	Random Forests	2.54	12.99	28.71	60.51
	GBM	2.27	6.64	11.33	18.75
	Robust GBM	2.29	7.13	12.13	19.02
	ANN	5.24	14.29	25.05	39.13
	Robust ANN	5.24	14.22	24.63	42.04
	Robust GAM	33.65	243.24	N/A	N/A

Table 6: Demographic and Clinical Characteristics of Study Subjects

Variable	Thiazide diuretic (n=504)	ACEI (n=1040)	p-value
	mean (sd)		
Systolic BP (mmHg)	134.19 (17.22)	133.97 (21.61)	0.838
Pre-treatment Systolic BP (mmHg)	140.17 (18.46)	138.46 (21.96)	0.131
Age (year)	54.10 (12.19)	54.08 (11.94)	0.975
BMI	38.97 (9.26)	37.57 (33.09)	0.350
Potassium (mmol/L)	4.06 (0.45)	4.03 (0.47)	0.375
Sodium (mmol/L)	139.06 (2.78)	138.60 (3.08)	0.005*
Cholesterol in LDL (mg/dL)	111.44 (42.24)	111.15 (53.06)	0.914
Cholesterol in HDL (mg/dL)	47.51 (13.89)	45.32 (16.68)	0.011*
Albumim (g/dL)	11.21 (14.09)	20.00 (17.24)	<0.001*
Triglyceride (mg/dL)	171.03 (114.82)	181.31 (188.53)	0.260
Hemoglobin A1c (%)	7.25 (2.03)	7.25 (1.99)	0.993
Glomerular filtration rate (ml/min/1.73m^2)	58.49 (18.56)	63.12 (18.04)	<0.001*
	n (percentage)		
Female	324 (64.3)	589 (56.6)	
Male	174 (34.5)	425 (40.9)	0.008*
Not answered	6 (1.2)	26 (2.5)	
Black	113 (22.4)	366 (35.2)	
White	279 (55.4)	415 (39.9)	<0.001*
More than one race or not answered	112 (22.2)	259 (24.9)	
Hispanic	91 (18.1)	215 (20.7)	0.254

Table 7: Conditional independence test results (p-value)

Method	IRD-test	IEP-test	ICQP-test	ITP-test
Robust RF + MCM-EA	0.17	0.50	1.00	0.38
Robust RF + RL	0.22	0.54	0.95	0.49
Robust GBM + MCM-EA	0.06	0.57	0.69	0.29
Robust GBM + RL	0.29	0.50	0.32	0.55

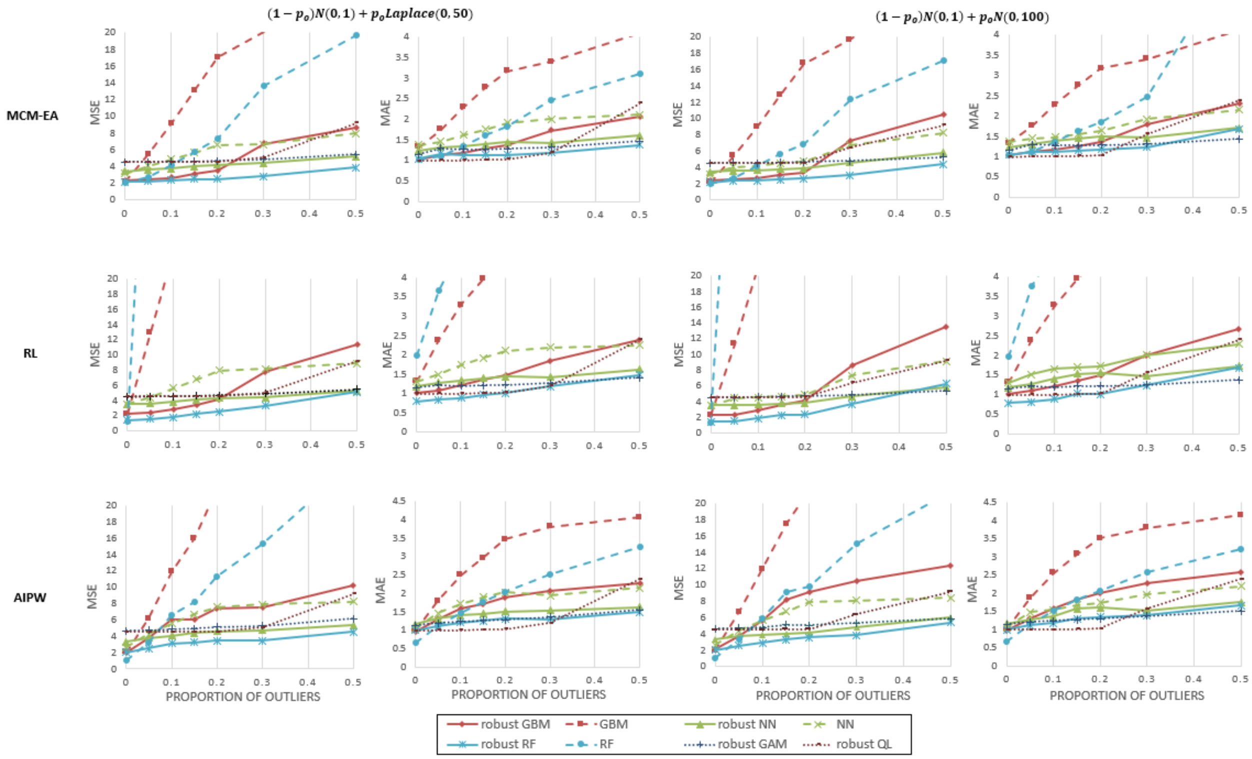


Figure 1: Results of Simulation Study 1 - MSE and MAE of different methods under various proportions of outliers and error generating mechanisms. The robust GBMs were indicated by red solid lines, the robust RFs were indicated by blue solid lines, the robust ANNs were indicated by green solid lines. The GBMs, RFs, and ANNs were indicated by dashed red, blue, and green lines. The robust GAMs were indicated by blue dotted line, and robust QL was indicated by brown dotted line.

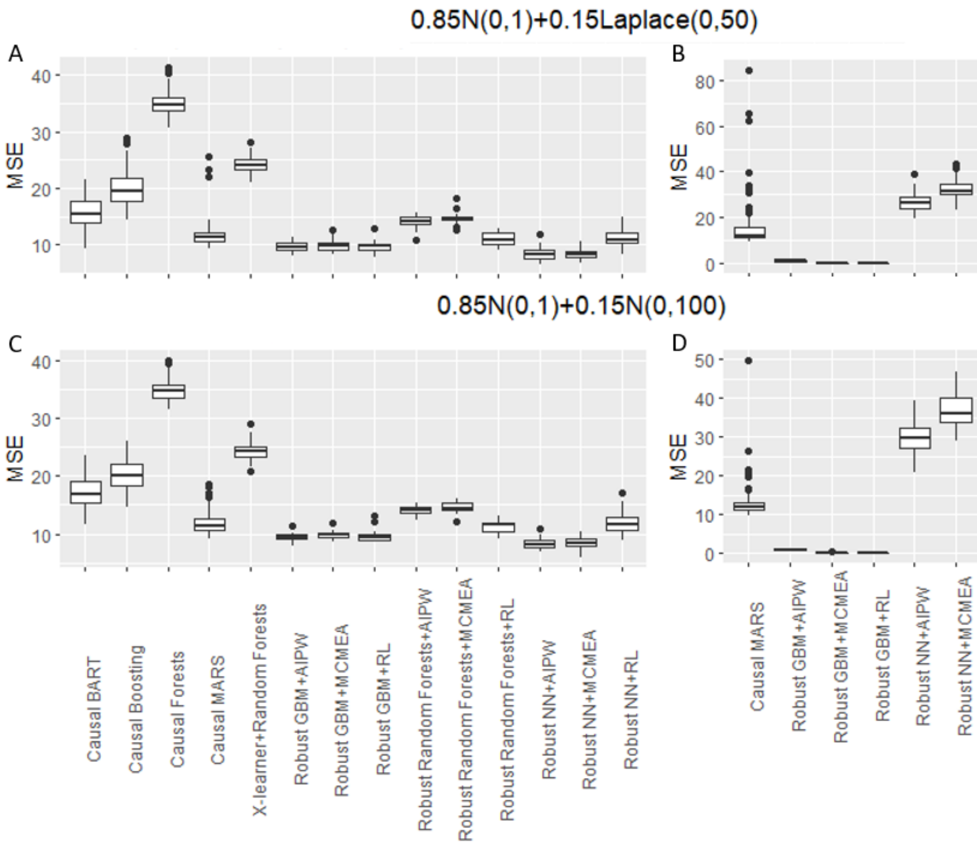


Figure 2: Simulation Study 2 - Mean squared error (MSE) of different algorithms when outliers exist. Figures A and C show the results when $p = 100$, Figures B and D show the results when $p = 2000$.

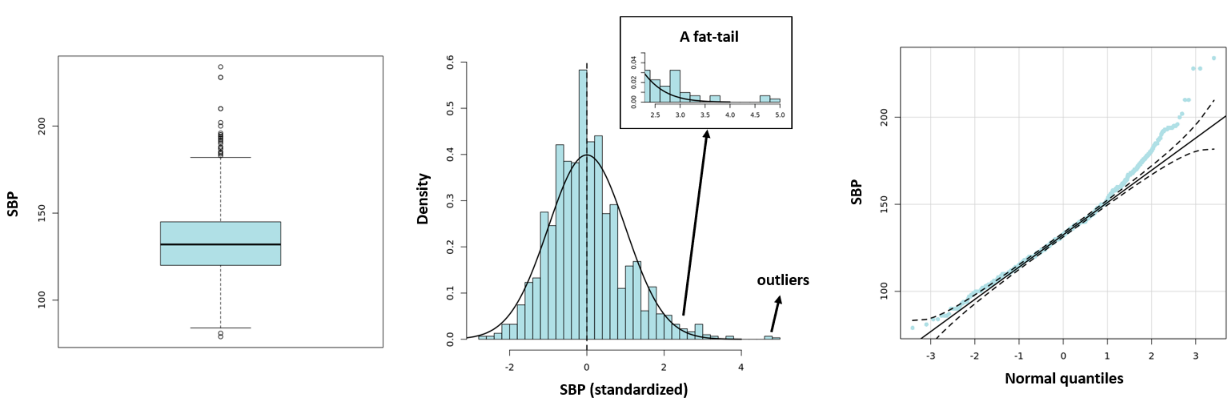


Figure 3: Heavy-tailed and Skewed Systolic Blood Pressure Distribution.

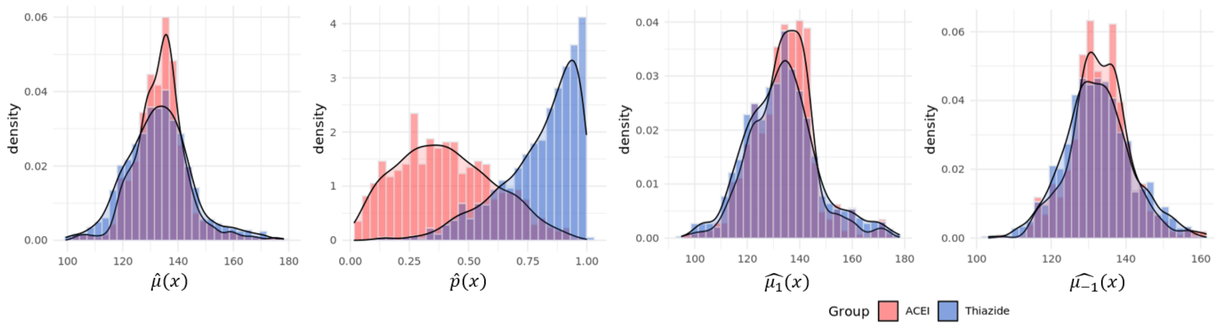


Figure 4: Data example: Estimated nuisance parameters by treatment group.

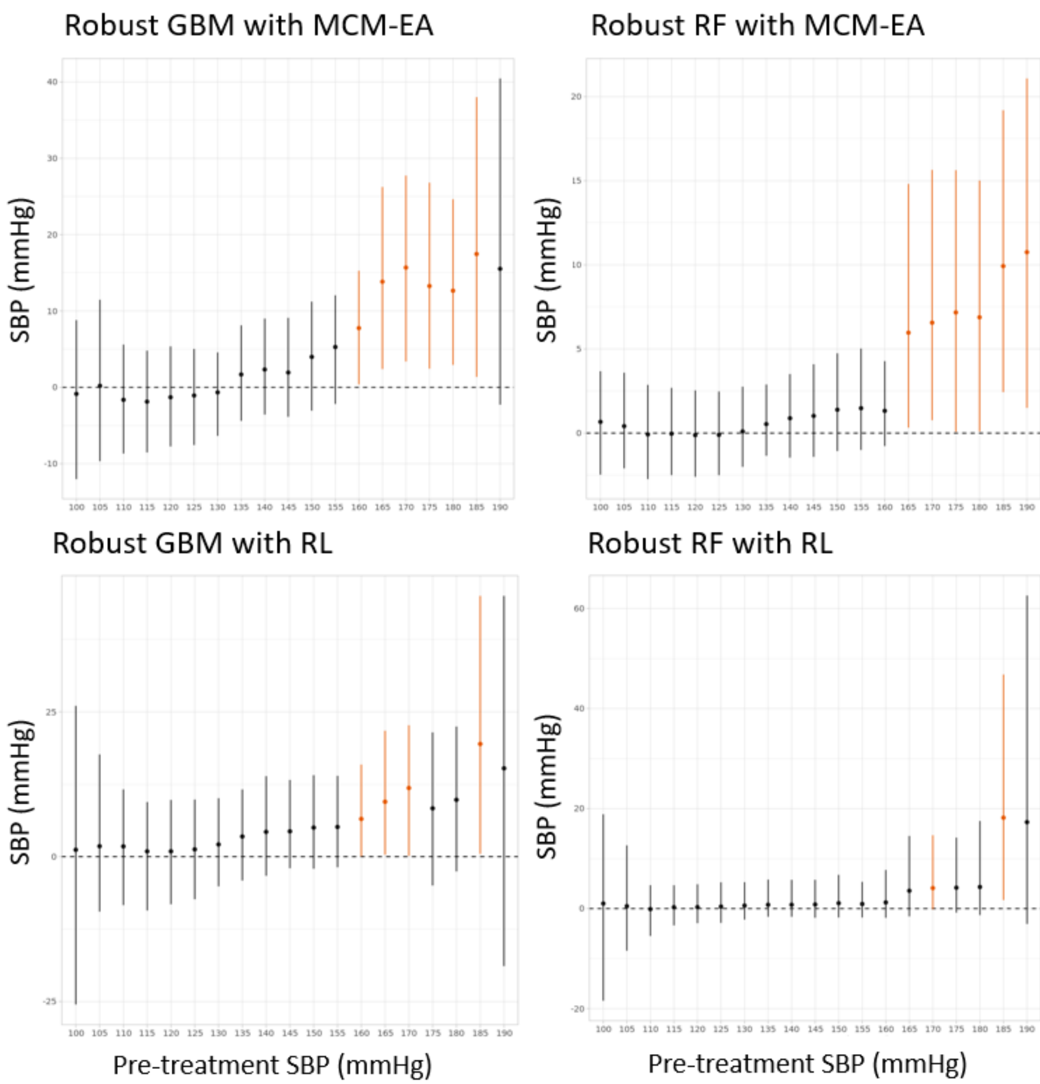


Figure 5: Data example: Marginal treatment effect of pre-treatment SBP. If the empirical 95% pointwise C.I. does not cover zero, the interval segment is colored in orange.

SUPPLEMENTAL MATERIALS

Appendix A: Implementation

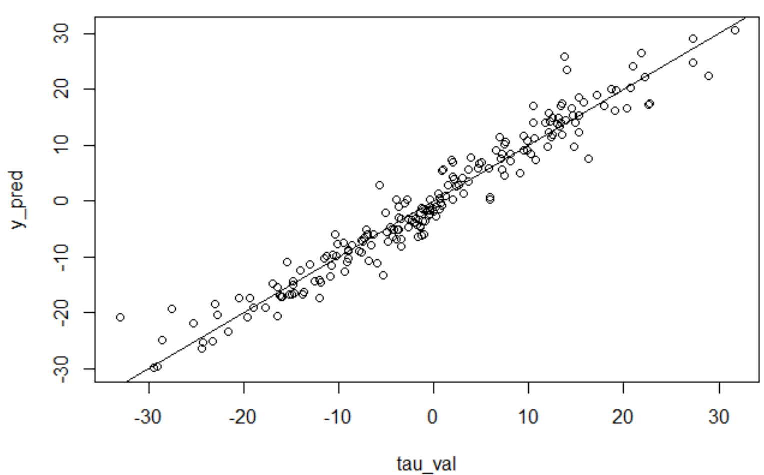
Title: RCATE package

R-package for robust estimation of CATE: R package **RCATE** containing code for 9 robust estimation algorithms of CATE described in the article and also the methods based on additive B-spline LAD regression in R.Li. The package also contains the dataset used as example in the article.

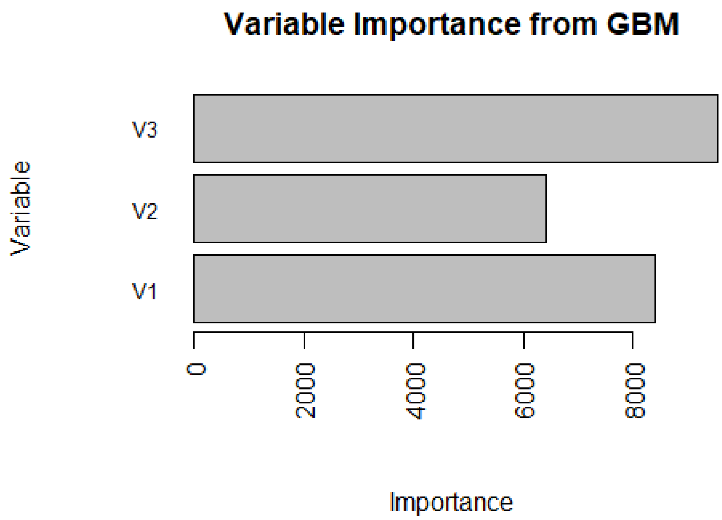
Hypertension dataset: Data set used in the illustration of robust estimation of CATE algorithms in Section 4.

Example of usage:

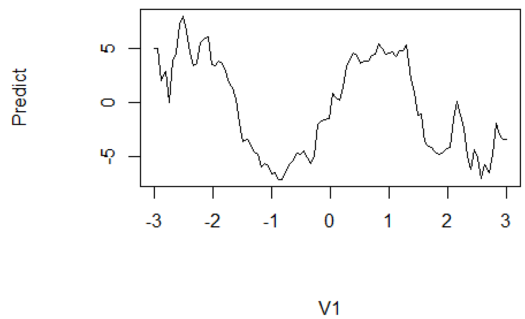
```
## Install package
require(devtools)
devtools::install_github("rhli-Hannah/RCATE")
library(RCATE)

## Data generation
n <- 1000; p <- 3; set.seed(2223)
X <- as.data.frame(matrix(runif(n*p,-3,3),nrow=n,ncol=p))
tau = 6*sin(2*X[,1])+3*(X[,2]+3)*X[,3]
p = 1/(1+exp(-X[,1]+X[,2]))
d = rbinom(n,1,p)
t = 2*d-1
y = 100+4*X[,1]+X[,2]-3*X[,3]+tau*t/2 + rnorm(n,0,1)
set.seed(2223)
x_val = as.data.frame(matrix(rnorm(200*3,0,1),nrow=200,ncol=3))
tau_val = 6*sin(2*x_val[,1])+3*(x_val[,2]+3)*x_val[,3]

## Use robust GBM + R-learning to estimate CATE
fit <- rcate.ml(X,y,d,method='RL',algorithm='GBM')
y_pred <- predict(fit,x_val)$predict
plot(tau_val,y_pred);abline(0,1)
```

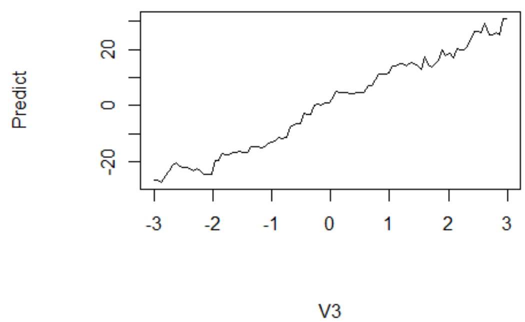


```
## Variable importance level
importance <- importance.rcate(fit)
```




```
## Marginal treatment effect plot
marginal.rcate(fit, 'V1')
marginal.rcate(fit, 'V3')
```





Appendix B: Supplemental Simulation Study Results

Supplemental Simulation Study (S): We compared the algorithm-based robust estimators against the model-based ones when the true treatment effect models were correctly specified. Here we assumed that the true effect effect τ was an additive function of $\mathbf{X}$. In such a situation, robust methods based on generalized additive models (GAM) should provide correct estimates. We also included in the simulation an L_1-based Q-learner (robust QL) for comparison.

Specifically, we defined the last two methods as follows:

$$\text{Robust GAM: } \hat{\beta} = argmin_{\beta} \frac{1}{n} \sum_{i=1}^{n} w_i^*(X_i, T_i) |Y_i^* - B(X_i)^T \beta| + \Lambda_n(\beta),$$

$$\text{Robust QL: } \hat{\gamma}, \hat{\beta} = argmin_{\gamma,\beta} \frac{1}{n} \sum_{i=1}^{n} |Y_i - B(X_i)^T \gamma - \frac{T_i}{2} B(X_i)^T \beta| + \Lambda_n(\gamma, \beta),$$

where Λ is a smoothness-sparsity penalty for group-wise variable selection and for smoothness of the regression line.

Model-based estimators can be more efficient when they depict the treatment effect with the right function. We simulated a situation where the true treatment effect τ is an additive function of $\mathbf{X}$. Since the model-based estimators used GAM to depict $\tau(\mathbf{X})$, we expect them to perform well. Algorithm-based estimators, on the other hand, may have reduced efficiency while offering a greater protection against model misspecification. Here we used model-based methods as a benchmark, and compared the performance of the algorithm-based estimators as sample size increased.

We compared all methods indicated by "(S)" in Table 4. We considered two scenarios: (1) For the robust GAMs, we fixed the sample size at $n_0 = 200$, and for robust GBMs, robust RFs, and robust ANNs, we increased the sample size from 200 to 1000 by an increment of 200; (2) For the robust GAMs, we fixed the sample size at $n_0 = 1000$; for the proposed robust algorithms, we increased the sample size from 1000 to 7000 by an increment of 2000. Specifically, we used two different error distributions $P = N(0, 100)$ and $P = Laplace(0, \sqrt{50})$, while fixing the proportion of outliers at $p_o = 0.15$. The covariates were continuous variables ($\mathbf{X}_i \sim N_{10}(0, 1)$).

Functions $b_0(\mathbf{X}_i)$ and $\tau_0(\mathbf{X}_i)$ in the response surface were

$$b_0(\mathbf{X}_i) = 100 + 4X_{i1} + X_{i2} - 3X_{i3},$$

$$\tau_0(\mathbf{X}_i) = 6sin(2X_{i1}) + 3X_{i2} + X_{i3} + 9tanh(0.5X_{i4}) + 3X_{i5},$$

where the true treatment effect function was an additive model of covariates. We reported the MSE of the CATE estimates graphically in Figure B.1.

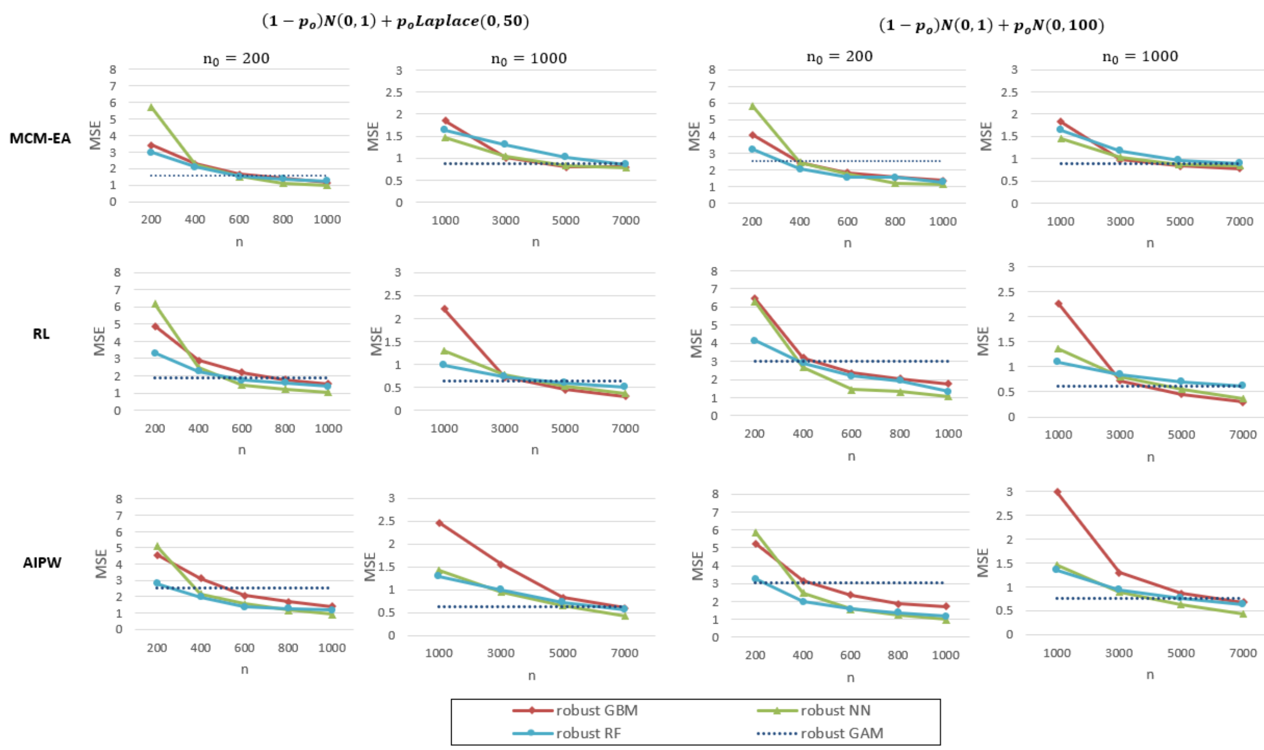


Figure B.1: Simulation results of Simulation S - MSE of different methods under different sample sizes. The robust GBMs were indicated by red solid line, the robust RFs were indicated by blue solid line, the robust ANNs were indicated by green solid line. The robust GAMs were indicated by blue dotted line. In the first and third columns of figures, the sample size of robust GAMs methods was $n_0 = 200$; in the second the fourth columns of figures, the sample size of robust GAMs methods was $n_0 = 1000$.

Figure B.1 showed that machine-learning algorithms' performance improved with the sample size.

Table B.1: Simulation Results of Simulation S ($n_0 = 200$)

		MCM-EA									
		$Laplace(0, \sqrt{50})$					$N(0, 100)$				
	n	200	400	600	800	1000	200	400	600	800	1000
MSE	robust GBM	3.43	2.28	1.66	1.39	1.14	4.08	2.43	1.80	1.55	1.37
	GBM	19.96	13.87	10.95	8.99	7.69	21.64	14.18	11.33	9.36	7.33
	robust NN	5.70	2.25	1.54	1.12	1.00	5.84	2.45	1.74	1.21	1.14
	NN	6.50	3.95	3.01	2.60	2.17	6.79	4.09	3.01	2.69	2.23
	robust RF	2.96	2.10	1.53	1.37	1.24	3.21	2.05	1.55	1.54	1.25
	RF	11.12	9.94	8.77	7.94	7.75	10.05	9.37	8.33	8.38	7.23
	robust GAM	1.54					2.54				
	n	200	400	600	800	1000	200	400	600	800	1000
MAE	robust GBM	1.44	1.32	0.99	0.90	0.80	1.57	1.21	1.03	0.94	0.87
	GBM	3.40	3.56	2.49	2.23	2.05	3.62	2.91	2.58	2.34	2.06
	robust NN	1.86	1.14	0.93	0.80	0.75	1.89	1.19	0.98	0.83	0.79
	NN	1.99	1.53	1.33	1.23	1.14	2.04	1.57	1.34	1.26	1.14
	robust RF	1.27	0.99	0.90	0.84	0.79	1.32	1.03	0.89	0.89	0.80
	RF	2.02	3.56	1.80	1.73	1.70	2.08	4.08	1.89	1.83	1.75
	robust GAM	0.83					1.13				
		RL									
		$Laplace(0, \sqrt{50})$					$N(0, 100)$				
	n	200	400	600	800	1000	200	400	600	800	1000
MSE	robust GBM	4.87	2.88	2.19	1.77	1.52	6.46	3.19	2.36	2.05	1.77
	GBM	36.93	22.98	17.28	14.28	11.88	38.67	24.42	17.34	16.77	11.55
	robust NN	6.19	2.48	1.48	1.23	1.04	6.28	2.67	1.44	1.33	1.07
	NN	6.78	4.28	3.45	2.87	2.60	7.30	4.39	3.51	3.10	2.62
	robust RF	3.32	2.24	1.76	1.59	1.39	4.14	2.92	2.20	1.92	1.34
	RF	49.57	49.48	9.36	43.16	47.60	133.48	92.12	40.54	68.37	50.74
	robust GAM	1.87					3.03				
	n	200	400	600	800	1000	200	400	600	800	1000
MAE	robust GBM	1.71	1.32	1.07	1.01	0.93	1.95	1.38	1.18	1.08	1.00
	GBM	4.56	3.56	2.50	2.73	2.48	4.78	3.71	3.16	2.97	2.53
	robust NN	1.94	1.19	0.91	0.83	0.77	1.96	1.23	0.92	0.87	0.78
	NN	2.03	1.60	1.43	1.31	1.24	2.12	1.63	1.44	1.36	1.26
	robust RF	1.27	0.99	0.86	0.81	0.72	1.36	1.03	0.87	0.85	0.77
	RF	3.61	3.56	1.81	3.57	3.67	4.61	4.08	3.64	4.35	3.89
	robust GAM	0.79					1.10				
		AIPW									
		$Laplace(0, \sqrt{50})$					$N(0, 100)$				
	n	200	400	600	800	1000	200	400	600	800	1000
MSE	robust GBM	4.55	3.13	2.09	1.70	1.41	5.23	3.15	2.36	1.87	1.73
	GBM	19.10	14.01	11.75	8.61	7.32	20.36	13.40	12.07	9.90	7.52
	robust NN	5.12	2.17	1.57	1.18	0.94	5.88	2.48	1.58	1.24	0.99
	NN	1.97	3.77	2.91	2.46	2.23	6.86	4.24	3.14	2.57	2.34
	robust RF	2.80	1.98	1.36	1.29	1.18	3.25	1.98	1.56	1.37	1.18
	RF	10.57	9.66	9.36	8.58	8.33	9.98	8.81	9.21	9.86	8.66
	robust GAM	2.52					3.04				
	n	200	400	600	800	1000	200	400	600	800	1000
MAE	robust GBM	1.65	1.33	1.07	0.94	0.84	1.77	1.33	1.13	0.99	0.92
	GBM	3.35	2.78	2.50	2.13	1.98	3.50	2.81	2.59	2.31	2.05
	robust NN	1.76	1.11	0.93	0.81	0.73	1.89	1.19	0.96	0.84	0.76
	NN	1.97	1.50	1.31	1.20	1.15	2.05	1.59	1.37	1.23	1.17
	robust RF	1.21	0.99	0.86	0.82	0.78	1.32	1.01	0.89	0.84	0.77
	RF	1.98	1.89	1.81	1.73	1.68	2.09	1.94	1.92	1.87	1.76
	robust GAM	0.99					1.11				

Table B.2: Simulation Results of Simulation S ($n_0 = 1000$)

		MCM-EA							
		$Laplace(0, \sqrt{50})$				$N(0, 100)$			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MSE	robust GBM	1.85	1.01	0.80	0.80	1.83	0.97	0.83	0.77
	GBM	11.22	2.71	1.99	1.55	11.10	2.83	2.03	1.59
	robust NN	1.48	1.04	0.84	0.78	1.46	1.03	0.86	0.84
	NN	2.76	2.61	2.18	1.88	2.74	2.76	2.20	1.95
	robust RF	1.63	1.31	1.01	0.84	1.62	1.17	0.96	0.88
	RF	5.12	5.00	4.28	4.13	5.26	4.85	4.11	3.81
	robust GAM	0.87				0.88			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MAE	robust GBM	1.06	0.75	0.67	0.61	1.06	0.74	0.68	0.68
	GBM	2.56	1.23	1.05	0.93	2.58	1.28	1.07	0.94
	robust NN	0.94	0.77	0.69	0.62	0.93	0.77	0.70	0.72
	NN	1.28	1.23	1.13	1.04	1.28	1.27	1.13	1.07
	robust RF	0.94	0.84	0.72	0.67	0.92	0.79	0.71	0.68
	RF	1.51	1.48	1.40	1.36	1.58	1.53	1.44	1.37
	robust GAM	0.60				0.63			

		RL							
		$Laplace(0, \sqrt{50})$				$N(0, 100)$			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MSE	robust GBM	2.21	0.72	0.45	0.30	2.27	0.72	0.45	0.30
	GBM	29.31	4.55	3.01	2.14	27.49	5.02	3.04	1.80
	robust NN	1.29	0.77	0.53	0.36	1.36	0.80	0.55	0.37
	NN	3.58	3.26	2.58	1.90	3.66	3.51	2.55	1.93
	robust RF	0.98	0.73	0.60	0.50	1.09	0.85	0.70	0.62
	RF	107.18	91.18	91.12	72.19	127.63	91.68	81.65	70.37
	robust GAM	0.64				0.61			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MAE	robust GBM	1.14	0.64	0.50	0.40	1.15	0.63	0.49	0.40
	GBM	3.80	1.51	1.20	0.97	3.75	1.59	1.25	0.96
	robust NN	0.88	0.67	0.55	0.46	0.90	0.68	0.56	0.46
	NN	1.45	1.37	1.22	1.04	1.47	1.42	1.22	1.06
	robust RF	0.71	0.62	0.54	0.46	0.70	0.60	0.51	0.47
	RF	5.44	4.89	4.61	4.04	5.73	4.96	4.65	4.19
	robust GAM	0.56				0.53			

		AIPW							
		$Laplace(0, \sqrt{50})$				$N(0, 100)$			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MSE	robust GBM	2.46	1.55	0.82	0.60	2.99	1.29	1.07	0.98
	GBM	14.43	3.98	2.47	1.54	15.25	3.51	4.17	1.88
	robust NN	1.43	0.94	0.66	0.43	1.46	0.89	0.63	0.43
	NN	3.83	3.27	2.54	2.04	3.61	3.15	2.67	2.06
	robust RF	1.28	1.00	0.73	0.57	1.35	0.93	0.75	0.63
	RF	7.80	7.76	7.51	6.20	7.97	7.39	7.09	6.72
	robust GAM	0.63				0.75			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MAE	robust GBM	1.48	0.77	0.59	0.48	1.15	0.76	0.61	0.48
	GBM	2.82	1.37	1.10	0.85	2.87	1.37	1.11	0.85
	robust NN	0.92	0.73	0.61	0.49	0.93	0.72	0.60	0.49
	NN	1.46	1.35	1.20	1.07	1.46	1.34	1.23	1.07
	robust RF	0.81	0.71	0.59	0.47	0.83	0.69	0.60	0.53
	RF	1.65	1.64	1.56	1.45	1.69	1.69	1.59	1.55
	robust GAM	0.52				0.53			

Table B.3: Simulation Results (Coverage Probabilities) of Simulation 1

	MCM-EA													
	$Laplace(0, \sqrt{50})$							$N(0, 100)$						
p_o	0.00	0.05	0.10	0.15	0.20	0.30	0.50	0.00	0.05	0.10	0.15	0.20	0.30	0.50
Robust GBM	0.95	0.94	0.91	0.93	0.93	0.95	0.97	0.95	0.94	0.95	0.94	0.94	0.98	0.97
GBM	0.95	0.93	0.96	0.96	0.94	0.97	0.96	0.95	0.96	0.97	0.99	0.99	0.91	0.95
Robust NN	0.95	0.94	0.96	0.92	0.94	0.99	0.98	0.95	0.92	0.98	0.92	0.95	0.98	0.93
NN	0.96	0.76	0.87	0.47	0.83	0.79	0.89	0.96	0.39	0.43	0.27	0.32	0.35	0.29
Robust RF	0.94	0.96	0.97	0.94	0.92	0.93	0.98	0.94	0.95	0.94	0.93	0.92	0.96	0.98
RF	0.96	0.96	0.98	0.97	0.97	0.89	0.96	0.96	0.90	0.92	0.92	0.93	0.89	0.87
Robust GAM	0.23	0.38	0.49	0.54	0.57	0.66	0.74	0.23	0.39	0.48	0.54	0.59	0.66	0.76
Robust QL	0.47	0.49	0.49	0.52	0.53	0.53	0.00	0.47	0.49	0.50	0.51	0.53	0.33	0.00
	RL													
	$Laplace(0, \sqrt{50})$							$N(0, 100)$						
p_o	0.00	0.05	0.10	0.15	0.20	0.30	0.50	0.00	0.05	0.10	0.15	0.20	0.30	0.50
Robust GBM	0.95	0.94	0.95	0.96	0.95	0.97	0.97	0.95	0.95	0.96	0.94	0.96	0.91	0.98
GBM	0.93	0.97	0.98	0.98	0.97	0.96	0.95	0.93	0.99	0.98	0.99	1.00	1.00	0.99
Robust NN	0.94	0.95	0.93	0.96	0.92	0.93	0.96	0.94	0.95	0.96	0.93	0.97	0.91	0.98
NN	0.96	0.82	0.94	0.95	0.88	0.85	0.91	0.96	0.94	0.93	0.94	0.85	0.82	0.36
Robust RF	0.95	0.96	0.97	0.94	0.98	0.95	0.94	0.95	0.92	0.91	0.90	0.87	0.82	0.72
RF	0.94	0.89	0.98	0.97	0.87	0.93	0.94	0.94	0.82	0.93	0.89	0.95	0.89	0.86
Robust GAM	0.27	0.48	0.57	0.61	0.65	0.71	0.80	0.27	0.49	0.58	0.64	0.65	0.72	0.81
Robust QL	0.47	0.49	0.49	0.52	0.53	0.53	0.00	0.47	0.49	0.50	0.51	0.53	0.33	0.00
	AIPW													
	$Laplace(0, \sqrt{50})$							$N(0, 100)$						
p_o	0.00	0.05	0.10	0.15	0.20	0.30	0.50	0.00	0.05	0.10	0.15	0.20	0.30	0.50
Robust GBM	0.96	0.93	0.92	0.91	0.90	0.89	0.90	0.96	0.94	0.93	0.92	0.90	0.98	0.93
GBM	0.96	0.92	0.98	0.97	0.95	0.95	0.98	0.96	0.97	0.96	0.99	0.97	0.90	0.91
Robust NN	0.94	0.92	0.95	0.93	0.92	0.96	0.98	0.94	0.98	0.96	0.93	0.95	0.93	0.98
NN	0.95	0.74	0.89	0.96	0.80	0.73	0.88	0.95	0.44	0.88	0.84	0.87	0.83	0.78
Robust RF	0.93	0.94	0.98	0.91	0.97	0.96	0.98	0.93	0.96	0.97	0.94	0.95	0.96	0.95
RF	0.94	0.98	0.97	0.93	0.93	0.82	0.96	0.94	0.94	0.91	0.87	0.95	0.86	0.82
Robust GAM	0.54	0.57	0.59	0.61	0.62	0.64	0.70	0.54	0.58	0.59	0.61	0.64	0.66	0.73
Robust QL	0.47	0.49	0.49	0.52	0.53	0.53	0.00	0.47	0.49	0.50	0.51	0.53	0.33	0.00

Table B.4: Tuning parameters of considered methods in simulation

Method	Parameter	Value
RF-based algorithms	Number of trees	50
	Fraction of feathers used in splitting	0.8
	Minimum node size	3
Boosting-based algorithms	Number of trees	1000
	Depth of trees	2
	Learning rate	0.1
Robust ANN	Number of hidden layers	2
	Number of neurons in hidden layers	p and $p/2$
	Adam optimization	$\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$
	L_1 regularization ($p = 100, 2000$)	0.1, if $p = 100$; 0.02, if $p = 2000$.
	Number of neurons in hidden layers ($p = 2000$)	$p/10$ and $p/40$
Robust GAM and QL	Number of knots	$\sqrt{n}/2$
	Number of degree	3
	γ in SCAD	3.7