MACHINE LEARNING ESTIMATION OF HETEROGENEOUS TREATMENT EFFECTS

Duong Trinh - 2494479T

ECON5108 - ISSUES IN ECONOMICS RESEARCH
Written Assignment

Adam Smith Business School
University of Glasgow

Word Count: 5195
April 2021

Contents

1 Introduction

Causal inference has long been a fundamental topic in economics and finance, especially when it plays an important role as the basis for policy or decision making. While many methods in modern econometric analysis have been developed to estimate treatment effects, recent literature records a promising tendency of applying machine learning methods to this task. Such causal machine learning methods not only prove to be useful in estimating average treatment effect but also allow us to investigate the heterogeneous treatment effects across subsets of the population in experimental and observational studies. Compared to traditional approaches, causal machine learning methods have some advantages: First, they are powerful tools in using data to recover complex interactions among variables and flexibly estimate the relationship between the outcome, the treatment indicator and covariates. Second, they allow for the inclusion of a large number of covariates as well as the implementation of systematic model selection.

This assignment aims to review and evaluate the performance of some machine learning estimators of heterogeneous treatment effects. The structure of this assignment is as follows: In section 2, I describe an overview of the potential outcome framework along with necessary assumptions to construct the parameter of interest (IATE - individual average treatment effect) and interpret it as a causal parameter. Then, I distinguish two major approaches in literature: the generic approach which combines numerous machine learning methods and the specific approach which modifies an existing machine learning method to target estimating treatment effects instead of outcomes. Next, five causal machine learning estimators are selected and explained in more detail in section 3. In section 4, I set up a Monte Carlo study with simulations relied on synthetic data generation processes (DGPs), where the true treatment effects are known, allows for the comparison of five different methods and to observe patterns and similarities. Finally, the simulation results and their implications are discussed in section 5.

2 Framework and Major Aprroaches

2.1 Framework and Definitions

I employ the Neyman-Rubin potential outcome framework, and assume a superpopulation or distribution $\mathcal{P}$ from which a realization of N independent random variables is given as the training data. That is, $(Y_i^0, Y_i^1, X_i, W_i) \sim \mathcal{P}$, where:

- $X_i \in \mathbb{R}^d$ is a $d-$ dimensional covariate or feature vector
- $W_i \in \{0, 1\}$ is the binary treatment assignment indicator (to be defined precisely later)
- $Y_i^0 \in \mathbb{R}$ is the potential outcome of unit i when i is assigned to the control group, and Y_i^1 is the potential outcome when i is assigned to the treatment group

To interpret the estimated parameter as a causal relationship, the following assumptions are needed:

Assumption 1: Conditional Unconfoundedness

$$Y_i^1, Y_i^0 \perp W_i \mid X_i$$

The assumption states that, once we condition on observable characteristics, the treatment assignment is independent to how each person would respond to the treatment (two potential outcomes). In other words, the rule that determines whether or not a person is treated is determined completely by their observable characteristics.

Assumption 2: Exogeneity of covariates

$$X_i^1 = X_i^0$$

According to this assumption, the covariates are not affected by the treatment.

Assumption 3: Overlap Assumption

$$\forall x \in \text{supp}(X), \quad 0 < P(W = 1 \mid X = x) < 1$$

This assumption states that at every point of the covariate space we can always find treated and control individuals to compare their outcomes.

Assumption 4: Stable Unit Treatment Value Assumption (SUTVA)

$$Y_i^{obs} = W_i Y_i^1 + (1 - W_i) Y_i^0$$

This assumption ensures that there is no interference, no spillover effects, and no hidden variation between treated and control observations.

Now, the fundamental problem of causal inference is that each individual can either receive the treatment or not, thus only one of the two potential outcomes (Y_i^w) is observable:

$$Y_i^{obs} = W_i Y_i^1 + (1 - W_i) Y_i^0 \tag{1}$$

To avoid clutter, denote Y_i^{obs} simply by Y_i from now on.

Hence, the *individual treatment effect* (ITE) $\xi_i = Y_i^1 - Y_i^0$ of W_i on Y_i is never observed, and directly training machine learning methods on this difference is not possible as a result. However, the identification of expectations of ξ_i may be possible under plausible assumptions. For example, the identification of the *average treatment effect* (ATE) could be defined as $\tau = E[\xi_i]$. My main focus is the *conditional average treatment effects* ($CATEs$). $CATEs$ take the expectations of ξ_i conditional on exogenous pre-treatment covariates. Define the finest conditioning level that uses all available covariates X_i as the *individualized average treatment effects* ($IATEs$):

$$\tau(x) = E[\xi_i \mid X_i = x] = \mu^1(x) - \mu^0(x) \tag{2}$$

where $\mu^w(x) = \mathbb{E}[Y_i^w | X_i = x] = \mathbb{E}[Y_i | X_i = x, W_i = w]$ denotes the conditional expectation of the unobserved potential outcomes.

As illustrated in equation (2), the fundamental task to estimate IATEs is to have a good approximation of function and hence a good estimate of the difference of two conditional expectations, $\mu^1(x)$ and $\mu^0(x)$. In the recent literature, there are two main approaches to this a non-standard machine learning problem: generic approach (with meta-learners or generic machine learning algorithms) and one-estimator specific approach (with modified machine learning methods). This categorization is used by Knaus et al. [2021] and Jacob [2021].

2.2 Generic Approach

Generic approach decomposes the causal estimation problem into several standard prediction problems and may be combined with a large variety of off-the-shelf machine learning methods (such as the lasso, random forest (RF), Bayesian Adaptive Regression Trees (BART),

boosting methods or neural networks). Since the base learners are not designed to estimate the IATEs directly, they are called meta-learners, or generic ML algorithms.

A very simple and straightforward method in this approach is the T-learner. It is a two-step procedure where the conditional mean functions are estimated separately with any supervised machine learning algorithm: $\mu^1(x) = \mathbb{E}[Y|X = x, W = 1]$ and $\mu^0(x) = \mathbb{E}[Y|X = x, W = 0]$. However, their usual target is to minimize the mean squared error (MSE) in two separate prediction problems rather than to minimize the mean squared error of IATEs. By splitting the sample in two groups there is only information on one group. This might be problematic if the two functions shrink different covariates which are actually important in both groups, especially in RCT studies. Indeed, the T-learner is outperformed in many settings in previous papers [Kuenzel, 2019, Kennedy, 2020, Knaus et al., 2021].

A closed alternative to the T-learner is the idea of modeling only one function and taking the treatment assignment into this function: $\mu(w, x) = \mathbb{E}[Y|W = w, X = x]$. The predicted IATEs are then the difference between the predicted values when the treatment assignment indicator is changed from control to treatment, with all other features held fixed. This method is called the S-learner [Foster et al., 2011, Hill, 2011]. While the S-learner will perform well if the treatment effect is simple, or even zero, if the treatment effect is strongly heterogeneous and the response model of the outcomes under treatment and control are very different, pooling the data will lead to worse finite sample performance [Kuenzel, 2019].

Furthermore, there have been some more advanced meta-learners proposed in recent literature to overcome the previous issues when targeting on IATE estimation, such as X-Learner, DR-Learner, R-Learner, etc. Their common background suggested by Knaus et al. [2021] is that they can be eventually represented as a weighted minimization problem with modified outcomes:

$$\min_{\tau} \left\{ \frac{1}{N} \sum_{i=1}^{N} \omega_i \left[Y_i^* - \tau \left(X_i \right) \right]^2 \right\} \tag{3}$$

The difference among these estimators is how the weights ω_i and the outcome modifications $Y*_i$ are constructed. In terms of implementation, the last step would regress modified outcomes on the covariates to obtain the final estimate and to make predictions on new observations. Before that, estimations of some nuisance parameters, such as conditional outcome means $\mu^w(x)$ and propensity $\pi(x)$, may be required. Any appropriate machine learning prediction method (base learners) can be used to estimate the nuisance parameters as well as the IATEs.

2.3 Specific Approach

Specific approach includes methods that alters existing machine learning methods to move the target from the estimation of outcomes to the estimation of IATEs. In contrast to meta-learners with flexible choice of machine learning algorithm, this approach mostly uses modifications of tree-based methods, such as Causal Tree by Athey and Imbens [2016] (modifying Regression Tree), Causal Boosting by Powers et al. [2018] (modifying Boosting), Causal Forest by Athey et al. [2019] (modifying Random Forest) or Causal BART by Hahn et al. [2020] (modifying Bayesian Additive Regression Tree - BART), etc.

There is a growing strand of literature developing new causal machine learning methods and/or comparing available methods (in both approaches) across different settings. Kuenzel [2019] develop X-learner make a comparison between meta-learners like the S-, T-, and X-learner as well as the causal forest in a simulation study. Nie and Wager [2020] compare their R-learner with the S-, T-, X- and U-learner as well as causal boosting. From a broader perspective, Knaus et al. [2021] compare the meta-learners such as the inverse probability weighting (IPW) estimator, doubly-robust (DR), modified covariate method (MCM), R-learner, and different versions of the causal forest in an empirical Monte Carlo study, while Jacob [2021] makes a comparison among four meta-learners (DR-, R-, T-, X-learner) and three modified ML methods (Causal Forest, Causal Boosting, Causal BART). With regards to the base learners (off-the-shell ML methods), Kuenzel [2019] use a random forest (RF) and Bayesian additive regression trees (BART), Nie and Wager [2020] use Boosting and the Lasso, while Knaus et al. [2021] use RF and the Lasso for the estimation of the nuisance functions. Rather than assign specific machine learning methods as base learner for the estimation, Jacob [2021] employs an ensemble of methods that are stacked together with different weights to minimize the dependence of the ML methods.

3 Causal Machine Learning of Effect Heterogeneity

In this section, I will describe the intuition and implementation of three modified machine learning algorithms (Post Lasso, Causal Tree, Causal Forest) and two meta-learners (X-Learner and DR-Learner). Within the limited scope of this assignment, the choice of base learner is only restricted on Random Forest. Since X-Learner and DR-Learner require estimating nuisance parameters at the first step, I apply cross-fitting based on a 50:50 sample split to reduce the bias due to over-fitting induce if nuisance and main parameters are estimated using the same observations [Chernozhukov et al., 2018, Knaus et al., 2021, Jacob, 2020]. Cross-fitting is implemented as follows: We split the data into two parts of the same size, subset A and B. We use subset A to train the nuisance functions and then use these functions to predict nuisance parameters in subset B. Then we estimate IATEs, $\hat{\tau}_1(x)$, based on the predicted nuisance parameters. Then we switch the roles of the sets, using subset B for training and subset A for estimation to obtain $\hat{\tau}_2(x)$. The two results are then averaged: $\hat{\tau}(x) = \frac{\hat{\tau}_1(x)+\hat{\tau}_2(x)}{2}$.

3.1 Post Lasso

In traditional econometrics, one possible way of modelling IATEs is by adding interaction effects to a linear model. Assume that the true model has the following form:

$$Y_i = \alpha + \beta_w W_i + \beta_x X_i + \beta_{xw} X_i W_i + \epsilon_i$$

Then we can obtain the IATEs as follows:

$$\tau(x) = \mu^1(x) - \mu^0(x) = E\left[Y_i \mid X = x, W = 1\right] - E\left[Y_i \mid X = x, W = 0\right] = \beta_w + \beta_{xw} x$$

It implies that different subpopulations corresponding to $X = x$ will have different treatment effects as long as $\beta_{xw} \neq 0$. The above approach is commonly used along with the OLS estimation method if the dimension of the covariate space is small. However, the problem becomes increasingly harder as the number of covariates increases and it's impossible to simply use OLS in high dimensional setup. In such scenarios, a common assumption is sparsity, i.e., to assume that the true model contains only nonzero coefficients. Thus, the problem then shifts to identifying and selecting out the relevant regressors. While an off-the-shelf method such as Lasso can be used if the objective is simply to produce accurate predictions, it also generates biased estimates of the coefficients for the sake of causal inference . Instead, a convenient alternative is the Post-Lasso estimator, which can be implemented in two following steps:

1. Apply Lasso to uncover relevant covariates (i.e., covariates associated with nonzero coefficients).

2. Run OLS using only these selected covariates.

As shown in Belloni and Chernozhukov (2013) and Belloni, Chen, Chernozhukov, and Hansen (2012), this method has several advantageous properties, provided that the regularization parameter be chosen appropriately.

3.2 Causal Tree

Causal Tree is a modified machine learning method invented by Athey and Imbens [2016] to alter Regression Tree method for the target of causal inference. The main idea is to partition the covariate space into subpopulations that differ in the magnitude of their treatment effects. The approach enables the construction of valid confidence intervals (unbiased and asymptotically normal estimates) for average treatment effects of groups defined by the tree leaves as long as "honesty" condition is satisfied. That is, the training sample is split into two parts: one for building the tree (including the cross-validation step) and the another for estimating the treatment effects given leaves of the tree.

Like Regression Tree, Causal Tree is easy to explain and is more suitable when the model involves non-linear relationships or complex interactions. Meanwhile, this method is limited to trees and does not account for splitting uncertainty, which is important in practical settings.

3.3 Causal Forest

Causal Forest, which was first introduced by Wager and Athey [2018] and generalized by Athey et al. [2019], is the average of a large number of causal trees, where trees differ from one another due to subsampling. The training and prediction procedure of this method is described as follows:

During training, a number of trees are grown on random subsamples of the dataset. Individual trees are trained through the following steps:

- First, a random subsample is drawn by sampling without replacement from the full dataset. A single root node is created containing this random sample.

- The root node is split into child nodes, and child nodes are split recursively to form a tree. The procedure stops when no nodes can be split further. Each node is split using the following algorithm:
  - A random subset of variables are selected as candidates to split on.
  - For each of these variables x, we look at all of its possible values v and consider splitting it into two children based on this value. The goodness of a split (x, v) is determined by how much it increases the heterogeneity in treatment effect between the two child nodes. For computational efficiency, we precompute the gradient of each observation, and optimize a linear approximation of this difference.
  - All examples with values for the split variable x that is less than or equal to the split value v are placed in a new left child node, and all examples with values greater than the v are placed in a right child node.
  - If a node has no valid splits, or if splitting will not result in an improved fit, the node is not split further and forms a leaf of the final tree.

When predicting on a test set, we gather a weighted list of the sample's neighbors based on what leaf nodes it falls in. We then calculate the treatment effect using the outcomes and treatment status of the neighbor examples.

In observational studies where self-selection into treatment is present, the first splits might not be a good representation of the treatment effect rather than differences due to confounding variables. To overcome this issue, Athey et al. [2019] suggest applying Causal Forest local centering. This means that we use the residuals of the outcome and treatment variable as data instead of the original values. It requires two nuisance functions to be trained beforehand to predict the conditional mean which is used to create the residuals.

3.4 X-Learner

The X-Learner is a meta-algorithm proposed by Kuenzel [2019] for IATEs estimation. It exploits the advantages of both the S-learner (pooling the data across treatment and control conditions is beneficial if the treatment effect is simple or when one of the treatment groups is much larger than the other) and the T-learner (separating data is beneficial if the treatment effect is strongly heterogeneous), therefore, it can adapt to many different settings. The estimation procedure is as follows:

1. Use any method to estimate separate response functions $\hat{\mu}^1(x)$ (using only data from the treatment group) and $\hat{\mu}^0(x)$ (using only data from the control group) (the same as the T-Learner).

2. Create imputed individual treatment effects:

$$\tilde{W}_i^1 = Y_i - \hat{\mu}^0(X_i) \quad \text{for all observations in treatment group}$$
$$\tilde{W}_i^0 = \hat{\mu}^1(X_i) - Y_i \quad \text{for all observations in control group}$$

3. Regress imputed treatment effects on covariates to obtain IATE estimates $\hat{\tau}^0(x)$ and $\hat{\tau}^1(x)$.

4. Take a weighted average of the IATE estimates:

$$\hat{\tau}(x) = g(x)\hat{\tau}_0(x) + [1 - g(x)]\hat{\tau}_1(x)$$

   where g is a function mapping to unit interval, typically chosen to be an estimator for the propensity score, $\hat{\pi}(x)$.

In summary, implementation of the X-Learner requires three nuisance parameters $(\mu^1(x), \mu^0(x), \pi(x))$ to be estimated and then two more additional models (regressing only for control observations, and again only for treated observations) to get the final IATE estimate. Regarding the base learners, I choose Random Forest (RF). In the original paper, the X-Learner with BART is considered as well.

3.5 DR-Learner

The DR-Learner is also a meta-algorithm for IATEs estimation which is more efficient than the T-Learner. It combines the T-learner and the inverse probability weighting (IPW) scheme on the residuals of both regression functions $(Y^w - \mu^w(x))$, thereby overcoming their drawbacks such as the minimization goal from the T-learner and a potentially high variance from an IPW model when some propensity scores are small. The estimation procedure is as follows:

1. Use any method to estimate separately response functions $\hat{\mu}^1(x)$ (using only data from the treatment group) and $\hat{\mu}^0(x)$ (using only data from the control group) as well as the propensity function $\hat{\pi}(x)$ (using data from both control and treatment groups).

2. Create the modified outcome based on the doubly robust (DR) estimator of Robins and Rotnitzky [1995]:

$$Y^*_{i,DR} = \mu^1(X_i) - \mu^0(X_i) + \frac{W_i\left(Y_i - \mu^1(X_i)\right)}{\pi(X_i)} - \frac{(1-W_i)\left(Y_i - \mu^0(X_i)\right)}{(1-\pi(X_i))}$$

The idea here is similar to the modified outcome based on inverse probability weighting (IPW) [Horvitz and Thompson, 1952]:

$$Y^*_{i,IPW} = Y_i \frac{W_i - \pi(X_i)}{\pi(X_i)(1-\pi(X_i))}$$

since $\tau(x) = \mathbb{E}\left[Y^*_{i,IPW} \mid X_i = x\right] = \mathbb{E}\left[Y^*_{i,DR} \mid X_i = x\right]$. However, in terms of simulation evidences, Powers et al. [2018] suggests that estimators based on $Y^*_{i,IPW}$ exhibit high variance and Knaus et al. [2021] shows that it is more likely to be outperformed by estimators based on $Y^*_{i,DR}$. The DR-Learner estimators might be more stable because of the double-robustness property, i.e., the estimator remains consistent if either the propensity score model or the conditional outcome model is correctly specified.

3. Regress the modified outcome $Y^*_{i,DR}$ on covariates to obtain IATE estimates $\hat{\tau}(x)$.

In summary, implementation of the DR-Learner requires three nuisance parameters $(\mu^1(x), \mu^0(x), \pi(x))$ to be estimated and then one more additional model (regressing modified outcome on covariates) to get the final IATE estimate. Regarding the base learners, I choose Random Forest (RF). While Knaus et al. [2021] consider both Lasso DR-Learner and Random Forest DR-Learner, their simulation results suggest that Random Forest DR-Learner is more likely to outperform in overall.

4 Simulation Design

4.1 General Simulation for estimating IATEs

In this section, I introduce the general framework of the following simulations. For each simulation, I specify the sample size n, the dimension d of feature space (both low and high dimension setup are considered) as well as the following functions:

- The treatment propensity: $\pi(x) = \mathbb{P}[W = 1 \mid X = x]$
  I consider the constant propensity, i.e. random assignment, balanced case when $\pi(x) = 0.5$ and unbalanced case when $\pi(x)$ is very small; or linear in the sense of logistic function: $\pi(X) = \frac{1}{1+e^{X\beta_w+\epsilon}}$ and $\epsilon \sim N(0,1)$
- The mean effect: $m(x) = 2^{-1}\mathbb{E}\left[Y^{(0)} + Y^{(1)} \mid X = x\right]$
  I consider both linear mean effect $m(x) = X\beta_m$ (when β_m is either dense or sparse)) or non-linear mean effect.
- The treatment effect: $\tau(x) = \mathbb{E}\left[Y^{(1)} - Y^{(0)} \mid X = x\right]$. Since this is main interest, I consider some specific cases of τ to vary the degree of heterogeneity.
- Thus, the conditional mean effect functions are: $\mu_0(x) = m(x) - 2^{-1}\tau(x)$ and $\mu_1(x) = m(x) + 2^{-1}\tau(x)$.

Then, to simulate an observation, i, in the training set, I simulate its feature vector, X_i, its treatment assignment, W_i, and its observed outcome, Y_i as below:

1. First, I simulate a d-dimensional feature vector X:
   - Independent: $X \sim N(0, I_{d\times d})$ or $X \sim U[0,1]^d$
   - Dependent: $X \sim N(0, \Sigma)$
2. Next, I simulate the treatment assignment W according to:
   $$W \sim Bernouli(\pi(X))$$
3. Finally, I create the observed outcome Y:
   $$Y \sim N\left[m(X) + (W - 0.5)\tau(x), \sigma_Y^2\right]$$
   where the conditional variance of Y given X and W is $\sigma_Y^2 = 1$ (noise level).

I train each $IATE$ estimator on a training set of n units and then evaluate its performance against a test set of n_{test} units for which the true effect is known. I replicate each experiment $R = 30$ times for large sample ($n = 5000$) and $R = 100$ times for small sample ($n \leq 1000$).

Simulation 1: Low dimensional data; No confounding; Balanced propensity; Linear mean effect; No treatment effect

$$
\begin{aligned}
n \in \{1000, 5000\}; \quad n_{test} = 1000; \quad d \in \{5, 10, 20\} \\
X \sim N(0, I_{d\times d}) \\
\pi(X) = 0.5 \\
m(X) = \Sigma_{k=1}^{d} \beta_k X_k \quad (\beta_k = \frac{1}{k} \ \forall k = \overline{1, d}) \\
\tau(X) = 0
\end{aligned}
$$

Firstly, I consider a simple setting which is a randomized experiment with a balanced propensity score of 0.5, and there is no treatment effect. The mean effect depends linearly on covariates, and this function tends to be sparse when d is larger.

Simulation 2: Low dimensional data; No confounding; Balanced propensity; Nonlinear mean effect; Linear treatment effect

$$
\begin{aligned}
n \in \{1000, 5000\}; \quad n_{test} = 1000; \quad d \in \{5, 10, 20\} \\
X \sim U[0, 1]^d \\
\pi(X) = 0.5 \\
m(X) = \sin(\pi X_1 X_2) + 2(X_3 - 0.5)^2 + X_4 + 0.5 X_5 \\
\tau(X) = \frac{1}{2}(X_1 + X_2)
\end{aligned}
$$

Departing from simulation 1, a more complicated mean effect is taken into account in this setting, i.e. nuisance components are difficult [Nie and Wager, 2020]. However, the treatment effect is kept simple.

Simulation 3: Low dimensional data; No confounding; Balanced propensity; No mean effect; Nonlinear treatment effect

$$
\begin{aligned}
n \in \{1000, 5000\}; \quad n_{test} = 1000; \quad d \in; \quad n_{test} = 1000; \quad d \in \{2, 4, 6, 8\} \\
X \sim U[0, 1]^d \\
\pi(X) = 0.5 \\
m(X) = 0 \quad (\beta = 0) \\
\tau(X) = \zeta(X_1)\zeta(X_2), \quad \zeta(X) = 1 + \frac{1}{1 + e^{-20(x-1/3)}}
\end{aligned}
$$

In contrast to the first two simulations, the treatment effect function in this setting is complex nonlinear. This is motivated by Wager and Athey [2018] to evaluate the ability of machine learning methods to adapt to heterogeneity in $\tau(X)$. The propensity and mean effect functions ($\pi(X)$ and $m(X)$) are held fixed.

Simulation 4: Low dimensional data; No confounding; Unbalanced propensity; Linear mean effect; Linear treatment effect

$$
\begin{aligned}
n \in \{1000, 5000\}; \quad n_{test} = 1000; \quad d \in \{2, 5, 8\} \\
X \sim N(0, I_{d\times d}) \\
\pi(X) = 0.1 \\
m(X) = X^T\beta \quad (\beta \sim U = [-1, 1]^d) \\
\tau(X) = 6.\mathbb{I}(X_1 > 0) + 8.\mathbb{I}(X_2 > 0)
\end{aligned}
$$

In this setting, the treatment group sizes are very unbalanced - $\pi(X) = 0.1$, i.e., on average only ten percent of the units receive treatment. It reflects the fact that treatment might be expensive in many cases of randomized control trials. Furthermore, I choose the treatment effect function ($\tau(X)$) that is quite simpler to estimate than the mean effect.

Simulation 5: Low dimensional data; Confounding; Linear, sparse mean effect; Nonlinear treatment effect

$$
\begin{aligned}
n \in \{1000, 5000\}; \quad n_{test} = 1000; \quad d \in \{2, 4, 6\} \\
X \sim U[0, 1]^d \\
\pi(X) = \frac{1}{4}(1 + \beta_{2,4}(X_1)) \\
m(X) = 2X_1 - 1 \\
\tau(X) = \zeta(X_1)\zeta(X_2), \quad \zeta(X) = 1 + \frac{1}{1 + e^{-20(x-1/3)}}
\end{aligned}
$$

where $\beta_{a;b}$ is the β-density with shape parameters a and b.

So far, only a randomized experiment has been examined, i.e., constant propensity. This setting is intended to emulate a problem in observational studies: a treatment assignment is often correlated with potential outcomes. Modifying the setup of simulation 3, Athey et al. [2019] introduce an interaction between $\pi(x)$ and $m(x)$. Both the selection bias arising from such interaction and the heterogeneous treatment effect ($\tau(x)$) are presented in this setup.

Simulation 6: High dimensional data; No Confounding; Balanced propensity; No mean effect; Nonlinear treatment effect

$$
\begin{aligned}
(n, n_{test}, d) &\in \{(200, 200, 400); (300, 300, 300); (400, 400, 200)\} \\
X &\sim U[0,1]^d \\
\pi(X) &= 0.5 \\
m(X) &= 0 \\
\tau(X) &= \zeta(X_1)\zeta(X_2), \quad \zeta(X) = 1 + \frac{1}{1 + e^{-20(x-1/3)}}
\end{aligned}
$$

So far, only low dimensional data has been considered. I modify simulation 3 by examining different values of (n, n_{test}, d), in which d is relatively large, and keep other parameters the same to evaluate the performance of machine learning causal estimators. High dimensional settings arise in many empirical problems, especially in current RCT where the number of baseline covariates is potentially very large.

4.2 Comparison Metrics

I consider four major performance measures: Mean Squared Error (MSE), Absolute Bias ($|Bias|$), Standard Deviation (SD) and Coverage Rate for 95% confidence interval ($Coverage$) for the prediction of each observation j in the testing sample:

$$MSE_j = \frac{1}{R}\sum_{r=1}^{R}\left[\xi\left(x_j, y_j^0\right) - \hat{\tau}\left(x_j\right)_r\right]^2$$

$$|Bias_j| = |\underbrace{\frac{1}{R}\sum_{r=1}^{R}\hat{\tau}\left(x_j\right)_r}_{\overline{\hat{\tau}}(x_j)_r} - \xi\left(x_j, y_j^0\right)|$$

$$SD_j = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left[\hat{\tau}\left(x_j\right)_r - \overline{\hat{\tau}}\left(x_j\right)_r\right]^2}$$

$$Coverage = \frac{1}{R}\sum_{r=1}^{R}\mathbb{I}\{\overline{\hat{\tau}}\left(x_j\right)_r - 1.96 * SD_j \leq \hat{\tau}\left(x_j\right)_r \leq \overline{\hat{\tau}}\left(x_j\right)_r + 1.96 * SD_j\}$$

Since 1000 parameters are corresponding to 1000 observations in the testing sample; I summarize the performance over the whole testing sample by taking the averages $\overline{MSE}$, $\overline{|Bias|}$, $\overline{SD}$ and $\overline{Coverage}$.

5 Results

5.1 Results of each simulation

Simulation 1 (Table 1):

In the large sample, Causal Forest and Post Lasso are two estimators with the highest performance. At $d = 5$ (small number of covariates), Post Lasso outperforms Causal Forest regarding all criteria except for the coverage rate (but it is a trade-off of a smaller standard deviation). As d increases, both absolute bias and standard deviation (and thereby MSE) of Post Lasso increases, while there is a fall in standard deviation and MSE of Causal Forest, which makes Causal Forest uniformly perform better than Post Lasso. It can be explained that Post Lasso relies on sparsity assumption, so it is more likely to omit some important variables when the number of covariates in the true model becomes higher.

In the smaller sample, Causal Forest remains its highest ranking in terms of all criteria, whereas Post Lasso is dominated by meta-learners (X-Learner and DR-Learner). The superior of these forest-based methods (Causal Forest, X-Learner and DR-Learner with Random Forest as the base-learner) are also reflected by their improvements with d. It is due to the fact that the variance of a forest depends on the product of the variance of individual trees times the correlation between different trees [Breiman et al., 1984, Hastie et al., 2009]. Apparently, when d is larger, the individual trees have more flexibility in how to place their splits, thus reducing their correlation and decreasing the variance of the full ensemble.

Although X-Learner and DR-Learner are competitive when estimating the zero treatment effect; the former performs slightly better than the latter.

Causal Tree is the worst estimator, producing both the highest absolute bias and highest standard deviation. The reason is that we are considering the model with linear relationship, while Regression Tree is suitable for non-linear relationship. Therefore, Causal Tree (a modification of Regression Tree) is often outperformed.

Table 1: Simulation Setup I

Low dimensional data; No confounding; Balanced propensity;
Linear mean effect; No treatment effect

MSE							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	5	**0.0003**	0.0470	0.0020	0.0063	0.0114
30	5000	10	0.0023	0.0549	**0.0017**	0.0059	0.0065
30	5000	20	0.0060	0.0684	**0.0013**	0.0049	0.0035
150	1000	5	0.0069	0.0449	**0.0050**	0.0136	0.0216
150	1000	10	0.0192	0.0493	**0.0044**	0.0121	0.0148
150	1000	20	0.0377	0.0537	**0.0062**	0.0129	0.0119
Absolute Bias							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	5	**0.0024**	0.0316	0.0047	0.0096	0.0143
30	5000	10	0.0129	0.0349	**0.0055**	0.0116	0.0116
30	5000	20	0.0262	0.0391	**0.0127**	0.0145	0.0128
150	1000	5	0.0205	0.0236	**0.0019**	0.0078	0.0099
150	1000	10	0.0465	0.0251	**0.0055**	0.0071	0.0091
150	1000	20	0.0625	0.0252	**0.0069**	0.0096	0.0106
Standard Deviation							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	5	**0.0150**	0.2149	0.0450	0.0782	0.1052
30	5000	10	0.0445	0.2322	**0.0417**	0.0751	0.0793
30	5000	20	0.0700	0.2583	**0.0337**	0.0670	0.0575
150	1000	5	0.0753	0.2107	**0.0712**	0.1153	0.1461
150	1000	10	0.1228	0.2203	**0.0660**	0.1090	0.1208
150	1000	20	0.1758	0.2303	**0.0787**	0.1125	0.1088
Coverage Rate							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	5	0.9493	0.9492	**0.9571**	0.9539	0.9524
30	5000	10	0.9124	0.9466	0.9435	**0.9532**	0.9516
30	5000	20	0.9266	0.9376	0.9341	0.9432	**0.9457**
150	1000	5	0.9114	0.9407	**0.9555**	0.9516	0.9485
150	1000	10	0.9204	0.9439	**0.9517**	0.9491	0.9516
150	1000	20	0.9220	0.9407	**0.9602**	0.9539	0.9518

Simulation 2 (Table 2):

Post Lasso ranks first in all categories except for the standard deviation which is dominated by Causal Forest in the small sample. This relative superiority may be attributed to the sparse structure of the response surface (only 5 over d covariates presented in the true model), even though the main effect is non-linear. Although X-Learner achieves the highest coverage rate, this amount is below the nominal coverage rate (95%).

Among forest-based estimators, X-Learner almost produces the highest level of prediction accuracy (lowest MSE) driven by the lowest absolute bias across the sample size N and the dimension of feature space d. In this respect, Causal Forest outperforms DR-Learner in the large sample because of its lower standard deviation, while DR-Learner performs better than Causal Forest in small sample thanks to its lower absolute bias. Similar to simulation 1, there is an improvement in the performance of these estimators as d increase since it entails a smaller variance.

Causal Tree is still the worst-performing estimator in terms of prediction accuracy. While the absolute bias of Causal Tree is quite similar to Causal Forest, the standard deviation of Causal Tree is approximately three times higher, which leads to significantly larger MSE but better coverage rate compared to Causal Forest. In fact, this Causal Tree estimator is implemented using the honest version (using two separated samples for constructing the tree and for estimating treatment effects within leaves of the tree). According to Athey and Imbens [2016], the honest versions anticipate correcting for bias in leaf estimates and thus prune less than the adaptive version; the main cost of small leaf size is high variance in leaf estimates. As a result, there is a trade-off between some goodness of fit (of treatment effects) and valid confidence intervals. Furthermore, Causal Forest performs poorly in terms of coverage since the bias effect overwhelms variance especially when d is larger.

Table 2: Simulation Setup II

Low dimensional data; No confounding; Balanced propensity;
Nonlinear mean effect; Linear treatment effect

MSE							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	5	**0.0030**	0.0441	0.0133	0.0118	0.0168
30	5000	10	**0.0025**	0.0452	0.0168	0.0137	0.0173
30	5000	20	**0.0028**	0.0482	0.0202	0.0150	0.0214
150	1000	5	**0.0159**	0.0572	0.0321	0.0241	0.0293
150	1000	10	**0.0192**	0.0600	0.0352	0.0284	0.0317
150	1000	20	**0.0182**	0.0628	0.0382	0.0313	0.0352
Absolute Bias							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	5	**0.0153**	0.0803	0.0769	0.0635	0.0573
30	5000	10	**0.0148**	0.0874	0.0916	0.0786	0.0839
30	5000	20	**0.0162**	0.0910	0.1039	0.0881	0.1059
150	1000	5	**0.0274**	0.1170	0.1295	0.0927	0.0735
150	1000	10	**0.0319**	0.1286	0.1379	0.1154	0.1112
150	1000	20	**0.0305**	0.1434	0.1477	0.1280	0.1321
Standard Deviation							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	5	**0.0502**	0.1857	0.0609	0.0730	0.1091
30	5000	10	**0.0455**	0.1824	0.0576	0.0621	0.0785
30	5000	20	**0.0485**	0.1892	0.0568	0.0546	0.0627
150	1000	5	0.1169	0.1887	**0.0769**	0.1010	0.1429
150	1000	10	0.1283	0.1858	**0.0757**	0.0860	0.1111
150	1000	20	0.1258	0.1763	**0.0666**	0.0766	0.0903
Coverage Rate							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	5	**0.9266**	0.9130	0.7093	0.8213	0.9049
30	5000	10	**0.9245**	0.9036	0.6112	0.6970	0.7442
30	5000	20	**0.9388**	0.9049	0.5586	0.6144	0.5866
150	1000	5	**0.9348**	0.8737	0.5877	0.7912	0.8989
150	1000	10	**0.9348**	0.8616	0.5604	0.6782	0.7771
150	1000	20	**0.9395**	0.8327	0.4877	0.5939	0.6415

Simulation 3 (Table 3):

Meta-learners performs the best in terms of MSE. X-learner ranks first in the large sample with small dimensions of the feature space, while DR-Learner performs better in all the remaining cases. With respects to nonlinear treatment effect, these meta-learners employs random forest as a base learner so that they can outperform lasso-related methods since random forests have a data-driven way to determine which nearby observations receive more weight, something that is especially important in environments with complex interactions among covariates. Compared to Post Lasso, DR-Learner and X-Learner produce significantly lower absolute bias which outweighs their higher variance. As a result, both their MSE and Coverage Rate is superior to Post Lasso. In contrast to preceding setups, this simulation illustrates how DR-Leaner and X-Learner can adapt to estimate the complex and smooth treatment effect.

Causal Forest performs comparably to two meta-learners in the large sample; unfortunately, its performance becomes clearly very poor with small sample size because of the bias effect (the absolute bias is six times higher compared to the case of large sample size, and three times higher compared to the corresponding Causal Tree estimator).

Although Causal Tree almost outperforms Post Lasso, it is clearly dominated by the forest-based methods.

Table 3: Simulation Setup III

Low dimensional data; No confounding; Balanced propensity; No mean effect; Nonlinear treatment effect

MSE							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	0.2313	0.0687	0.0231	**0.0190**	0.0443
30	5000	4	0.2323	0.0827	0.0276	**0.0230**	0.0275
30	5000	6	0.2315	0.0874	0.0299	0.0307	**0.0270**
30	5000	8	0.2326	0.0893	0.0307	0.0356	**0.0278**
150	1000	2	0.2410	0.2396	0.4558	0.0673	**0.0661**
150	1000	4	0.2478	0.2519	0.4849	0.1425	**0.0816**
150	1000	6	0.2508	0.2407	0.4846	0.1884	**0.0945**
150	1000	8	0.2500	0.2426	0.4767	0.2182	**0.1059**
Absolute Bias							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	0.3776	0.0586	0.0924	0.0432	**0.0354**
30	5000	4	0.3772	0.0936	0.1095	0.0870	**0.0571**
30	5000	6	0.3772	0.0877	0.1136	0.1134	**0.0767**
30	5000	8	0.3773	0.1068	0.1182	0.1273	**0.0885**
150	1000	2	0.3776	0.2020	0.6204	0.1789	**0.0898**
150	1000	4	0.3776	0.2039	0.6352	0.3103	**0.1770**
150	1000	6	0.3782	0.2061	0.6358	0.3669	**0.2139**
150	1000	8	0.3790	0.1990	0.6260	0.3969	**0.2335**
Standard Deviation							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	**0.0484**	0.2337	0.0987	0.1259	0.2066
30	5000	4	**0.0525**	0.2442	0.0971	0.1029	0.1495
30	5000	6	**0.0489**	0.2528	0.0992	0.0960	0.1308
30	5000	8	**0.0582**	0.2420	0.1004	0.0959	0.1233
150	1000	2	0.3643	0.8442	**0.1328**	0.7330	0.9243
150	1000	4	0.4222	0.8489	**0.1891**	0.4463	0.8103
150	1000	6	0.4453	0.8359	**0.1728**	0.3777	0.7103
150	1000	8	0.4309	0.8509	**0.2152**	0.3627	0.6704
Coverage Rate							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	0.2031	0.9413	0.7808	0.9308	**0.9489**
30	5000	4	0.2133	**0.9352**	0.7290	0.8213	0.9263
30	5000	6	0.2042	**0.9387**	0.6500	0.6678	0.8846
30	5000	8	0.2323	**0.9158**	0.6760	0.6530	0.8560
150	1000	2	0.3643	0.8442	0.1328	0.7330	**0.9243**
150	1000	4	0.4222	**0.8489**	0.1891	0.4463	0.8103
150	1000	6	0.4453	**0.8359**	0.1728	0.3777	0.7103
150	1000	8	0.4309	**0.8509**	0.2152	0.3627	0.6704

Simulation 4 (Table 4):

When unbalanced propensity is introduced into the model, DR-Learner is the best estimator in all criteria except for standard deviation which is dominated by Post Lasso (lowest), X-Learner and Causal Forest. Regarding the mean squared error criterion, X-Learner is the second-best estimator. In fact, X-Learner is designed for addressing unbalance issue since it can use information from the control group to derive better estimators for the treatment group and vice versa [Kuenzel, 2019]. Since data on treated units is much smaller than data on control units, the model for $\mu^1(x) = \mathbb{E}[Y^1|X^1 = x]$ and hence the model for $\tau_0(x) = \mathbb{E}[\tilde{W}^0|X^1 = x]$ are relatively poor. However, the final estimator $\hat{\tau}$ imposes a small weight (estimated propensity score $\hat{\pi}(x)$) on $\hat{\tau}_0$ so that prediction is more accurate. Furthermore, the dominance of DR-Learner, in this case, may be attributed to its doubly robust property: the estimator is unbiased if either propensity score or outcome regressions are correctly specified [Curth and Schaar, 2021], and its mean squared error is only affected by the product of the mean squared errors of the propensity score and outcome regression estimators [Kennedy, 2020]. Hence, the effect of a poor model for $\mu^1(x)$ is alleviated, thus DR-Learner estimator still enjoys relatively lower absolute bias and lower MSE compared to others which are not doubly robust (include the X-Learner, which is expected to inherit larger error rates from individual $\hat{\mu}^w$ estimators).

The explanation for poor performance of Causal Forest and Causal Tree may be their splitting mechanism. Since there are only few treated observations, not many leaves can be split. Consequently, leaf estimates would suffer bias and lead to a large MSE. It is worth noting that even though X-Learner and DR-Learner employ Random Forest as base learner, the effect is trivial since they put a very small weight on poor estimations as mentioned above.

Table 4: Simulation Setup I

Low dimensional data; No confounding; Unbalanced propensity propensity; Linear mean effect; Linear treatment effect

MSE							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	18.1389	23.1169	31.2220	8.5894	**1.4594**
30	5000	5	18.1771	23.4805	30.4210	10.3012	**2.4337**
30	5000	8	18.1541	22.8435	29.9059	10.5214	**2.1894**
150	1000	2	18.1498	22.5110	31.1415	8.6160	**1.3966**
150	1000	5	18.1605	22.8288	30.3738	10.0053	**2.0926**
150	1000	8	18.1423	22.8507	29.8875	10.5617	**2.3017**
Absolute Bias							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	2.9403	2.3590	2.9490	1.1427	**0.2485**
30	5000	5	2.9326	2.3425	2.9239	1.4276	**0.4058**
30	5000	8	2.9378	2.3263	2.9498	1.5186	**0.4851**
150	1000	2	2.9389	2.3110	2.9412	1.1345	**0.2127**
150	1000	5	2.9387	2.3150	2.9199	1.4163	**0.3913**
150	1000	8	2.9400	2.3159	2.9517	1.5329	**0.4689**
Standard Deviation							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	**0.1712**	1.5602	0.4450	0.3677	0.5635
30	5000	5	**0.1870**	1.6403	0.4589	0.3879	0.5202
30	5000	8	**0.1701**	1.5514	0.5605	0.4103	0.5176
30	5000	2	**0.1768**	1.4673	0.4939	0.3825	0.5653
150	1000	5	**0.1929**	1.5713	0.5062	0.3988	0.5200
150	1000	8	**0.2013**	1.6035	0.5330	0.4176	0.5326
Coverage Rate							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	0.0571	0.6897	0.3865	0.5804	**0.9321**
30	5000	5	0.0631	0.7094	0.3475	0.4269	**0.8920**
30	5000	8	0.0566	0.7052	0.2702	0.3735	**0.8573**
150	1000	2	0.0584	0.7037	0.3958	0.5984	**0.9350**
150	1000	5	0.0642	0.7121	0.3700	0.4198	**0.8966**
150	1000	8	0.0652	0.7111	0.2964	0.3739	**0.8694**

Simulation 5 (Table 5,6):

In comparison to simulation 3 (pure heterogeneity), the presence of confounding does not change the relative ranking among causal machine learning estimators across all criteria, though it drives all of their performance worse slightly.

Since the Causal Forest is specialized to maximize heterogeneity in experimental settings but it is not built to explicitly account for confounding. As a result, its tendency for choosing splits does not sufficiently remove selection bias. in contrast, Causal Forests with local centering address this problem by partialling out the selection effects at first step. Thus, they are specialized to maximize effect heterogeneity as well as to account for selection bias. Consequently, they uniformly perform better than Causal Forests. The moderate improvement of Causal Forest with local centering (in Table 6) compared to Causal Forest (in Table 5) is driven by a relatively low mean absolute bias, but a higher mean SD partly offsets this advantage. However, the version with local centering is still the better choice if the goal is to minimize MSE or maximize coverage rate.

Nonetheless, using the version with local centering does not change the relative ranking of Causal Forest in comparison to other estimators we are considering in this simulation setup, while it comes at the cost of estimating two additional nuisance parameters.

Table 5: Simulation Setup V

Low dimensional data; Confounding;
Linear, sparse mean effect; Nonlinear treatment effect

MSE							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	0.2326	0.0964	0.0304	**0.0204**	0.0525
30	5000	4	0.2332	0.1075	0.0414	**0.0298**	0.0333
30	5000	6	0.2324	0.1087	0.0365	0.0333	**0.0303**
150	1000	2	0.2444	0.4396	0.6233	0.0814	**0.0756**
150	1000	4	0.2499	0.3913	0.6270	0.1614	**0.0894**
150	1000	6	0.2472	0.4221	0.6252	0.2110	**0.1093**
Absolute Bias							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	0.3785	0.0891	0.1105	0.0496	**0.0353**
30	5000	4	0.3787	0.1180	0.1397	0.1055	**0.0725**
30	5000	6	0.3778	0.1100	0.1251	0.1178	**0.0776**
150	1000	2	0.3780	0.3827	0.7002	0.2062	**0.0954**
150	1000	4	0.3795	0.3428	0.7001	0.3339	**0.1882**
150	1000	6	0.3788	0.3789	0.6975	0.3940	**0.2334**
Standard Deviation							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	**0.0491**	0.2624	0.1036	0.1276	0.2253
30	5000	4	**0.0589**	0.2688	0.1102	0.1076	0.1583
30	5000	6	**0.0511**	0.2676	0.1069	0.1019	0.1422
150	1000	2	0.1186	0.4897	**0.1090**	0.1620	0.2502
150	1000	4	0.1343	0.4711	**0.1184**	0.1520	0.2032
150	1000	6	0.1265	0.4790	**0.1137**	0.1426	0.1927
Coverage Rate							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	5000	2	0.1963	0.9313	0.7038	0.9225	**0.9495**
30	5000	4	0.2300	**0.9251**	0.6549	0.7755	0.9152
30	5000	6	0.2019	**0.9268**	0.6547	0.6843	0.8857
150	1000	2	0.3904	0.7868	0.1222	0.6961	**0.9245**
150	1000	4	0.4262	0.7924	0.1354	0.4389	**0.8062**
150	1000	6	0.4096	**0.7805**	0.1380	0.3344	0.7146

Table 6: Simulation Setup V

Low dimensional data; Confounding;
Linear, sparse mean effect; Nonlinear treatment effect

MSE							
R	N	d	Post Lasso	Causal Tree	Causal Forest (lc)	X-Learner	DR-Learner
30	5000	2	0.2326	0.0964	0.0300	**0.0204**	0.0525
30	5000	4	0.2332	0.1075	0.0364	**0.0298**	0.0333
30	5000	6	0.2324	0.1087	0.0346	0.0333	**0.0303**
150	1000	2	0.2444	0.4396	0.6250	0.0814	**0.0756**
150	1000	4	0.2499	0.3913	0.6272	0.1614	**0.0894**
150	1000	6	0.2472	0.4221	0.6213	0.2110	**0.1093**
Absolute Bias							
R	N	d	Post Lasso	Causal Tree	Causal Forest (lc)	X-Learner	DR-Learner
30	5000	2	0.3785	0.0891	0.1097	0.0496	**0.0353**
30	5000	4	0.3787	0.1180	0.1309	0.1055	**0.0725**
30	5000	6	0.3778	0.1100	0.1243	0.1178	**0.0776**
150	1000	2	0.3780	0.3827	0.7011	0.2062	**0.0954**
150	1000	4	0.3795	0.3428	0.7021	0.3339	**0.1882**
150	1000	6	0.3788	0.3789	0.6937	0.3940	**0.2334**
Standard Deviation							
R	N	d	Post Lasso	Causal Tree	Causal Forest (lc)	X-Learner	DR-Learner
30	5000	2	**0.0491**	0.2624	0.1031	0.1276	0.2253
30	5000	4	**0.0589**	0.2688	0.0995	0.1076	0.1583
30	5000	6	**0.0511**	0.2676	0.1023	0.1019	0.1422
150	1000	2	0.1186	0.4897	**0.1097**	0.1620	0.2502
150	1000	4	0.1343	0.4711	**0.1134**	0.1520	0.2032
150	1000	6	0.1265	0.4790	**0.1170**	0.1426	0.1927
Coverage Rate							
R	N	d	Post Lasso	Causal Tree	Causal Forest (lc)	X-Learner	DR-Learner
30	5000	2	0.1963	0.9313	0.7279	0.9225	**0.9495**
30	5000	4	0.2300	**0.9251**	0.6378	0.7755	0.9152
30	5000	6	0.2019	**0.9268**	0.6472	0.6843	0.8857
150	1000	2	0.3904	0.7868	0.1283	0.6961	**0.9245**
150	1000	4	0.4262	0.7924	0.1323	0.4389	**0.8062**
150	1000	6	0.4096	**0.7805**	0.1531	0.3344	0.7146

Simulation 6 (Table 7):

In comparison to simulation 3 (low dimensional data), in the context of high dimensional data, the performance of all causal machine learning estimators become much worse. Post Lasso turns into the best-performing estimator, especially in terms of coverage rate. Meanwhile, the relative rankings among others stay the same, i.e. DR-Learner, X-Learner, causal Forest and Causal Tree are in descending order of MSE. This phenomenon can be explained by the superiority of Lasso in the setting with a sparse model (there are only 2 over d variables that truly affect treatment effect as well as outcomes in this case).

Table 7: Simulation Setup VI

High dimensional data; No Confounding; Balanced propensity propensity; No mean effect; Nonlinear treatment effect

MSE							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	200	400	**0.8201**	0.9930	0.9902	0.9844	0.9618
30	300	300	**0.6639**	0.9683	0.9489	0.9359	0.8878
30	400	200	**0.5568**	0.9760	0.9570	0.9287	0.8278
Absolute Bias							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	200	400	**0.6468**	0.8964	0.9010	0.8990	0.8876
30	300	300	**0.6105**	0.8833	0.8797	0.8740	0.8513
30	400	200	**0.5621**	0.8836	0.8824	0.8699	0.8225
Standard Deviation							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	200	400	0.5738	0.1380	0.1225	0.1209	**0.1204**
30	300	300	0.4713	0.1597	**0.0916**	0.0935	0.1002
30	400	200	0.4129	0.1706	**0.0993**	0.1013	0.1091
Coverage Rate							
R	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
30	200	400	**0.7762**	0.0900	0.0778	0.0753	0.0757
30	300	300	**0.7127**	0.1080	0.0612	0.0626	0.0670
30	400	200	**0.6879**	0.1222	0.0680	0.0703	0.0781

Computational Time (Table 8):

Table 8 illustrates the average computation times (in seconds) of the different estimation approaches. It can be seen that meta-learners take a significantly large amount of computational time since they involve estimating many models for nuisance parameters, computing complicated modified outcome and implementing cross-fitting procedure. DR-Learner requires more time than X-Learner. The computational time for Causal Forest is about a third of X-Learner, but still much larger than other modified ML algorithms. Owning to simple estimation procedure, Post Lasso and Causal Tree only require a few time. The former becomes slower when it selects more variables, while the latter becomes slower when it splits more leaves.

Table 8: Average computation time of one replication in seconds

Simulation	N	d	Post Lasso	Causal Tree	Causal Forest	X-Learner	DR-Learner
1	5000	5	**0.24**	0.43	9.54	29.54	42.78
2	5000	5	**0.25**	0.62	11.25	31.38	43.92
3	5000	2	**0.22**	0.44	7.25	22.46	36.17
4	5000	2	0.31	**0.1**	6.92	19.64	31.02
5	5000	2	1.08	**0.34**	7.53	20.92	32.35
6	200	400	2.98	**0.24**	0.81	3.03	4.25

5.2 Overall comparison

The result implies that no causal machine learning estimator is uniformly superior for all settings and sample sizes.

Post Lasso tends to outperform in the settings with simple IATEs (no treatment effect/ linear treatment effect), sparse model and high dimensional model (simulation 1,2,6) but it works poorly if the data structure is quite complicated and non-linear.

Meta-learners (X-Leaner and DR-Learner) are dominant choice in the settings with nonlinear treatment effect, confounding, unbalanced propensity (simulation 3,4,5). They still have a relative good performance in the settings with simple IATEs (simulation 1,2) but perform very poor in high dimensional setting (simulation 6). Furthermore, DR-Learner almost outperform X-Leaner in terms of both MSE and coverage rate in all settings. This superiority might be attributed to the doubly robust property of DR-Learner. This result is consistent with simulation result in Kennedy [2020], although the base learners and setups are different.

Among modified machine learning algorithm, Causal Forest always outperform Causal Tree in terms of MSE, while Causal Tree are more likely to achieve better coverage rate in many settings. In particularly, Causal Forest often share the similarly good patterns with Random Forest meta-learners, although being moderately dominated. In contrast, Causal Tree never shows the best performance regarding prediction accuracy (MSE). Additionally, Causal Forest performs poorly in terms of coverage since the bias effect overwhelms variance especially when d becomes larger or sample size is small.

As the sample size (N) is larger, the performance of all estimators are improves. As the dimension of feature space (d) is higher, all forest-based estimators (Causal Forest, X-Learner, DR-Learner) tend to perform better owing to the flexibility of individual trees when placing their splits.

In summary, (RF) DR-Learner is most likely to be the best estimator among five causal machine learning estimators in this essay. (RF) X-learner and Causal Forest could be seen as alternatives if computational constraints are binding. Post Lasso should be considered in the case of sparse model. In the context of high dimensional data, more causal machine learning methods are required to make better performance, such as Causal Boosting [Powers et al., 2018]. This conclusion, nonetheless, needs further investigation in future studies because of some simplifications: choice of tuning parameters, choice of base learners for meta algorithms, methods of sample splitting and cross-fitting should be paid more attentions.

References

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

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

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

V. Chernozhukov, M. Demirer, E. Duflo, and I. Fernandez-Val. Generic machine learning inference on heterogenous treatment effects in randomized experiments. Technical report, National Bureau of Economic Research, 2018.

A. Curth and M. Schaar. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1810–1818. PMLR, 2021.

J. C. Foster, J. M. Taylor, and S. J. Ruberg. Subgroup identification from randomized clinical trial data. *Statistics in medicine*, 30(24):2867–2880, 2011.

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

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

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

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

D. Jacob. Cross-fitting and averaging for machine learning estimation of heterogeneous treatment effects. *arXiv preprint arXiv:2007.02852*, 2020.

D. Jacob. Cate meets ml-conditional average treatment effect and machine learning. *Available at SSRN 3816558*, 2021.

E. H. Kennedy. Optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 2020.

M. C. Knaus, M. Lechner, and A. Strittmatter. Machine learning estimation of heterogeneous causal effects: Empirical monte carlo evidence. *The Econometrics Journal*, 24(1):134–161, 2021.

S. R. Kuenzel. *Heterogeneous Treatment Effect Estimation Using Machine Learning*. PhD thesis, UC Berkeley, 2019.

X. Nie and S. Wager. Quasi-oracle estimation of heterogeneous treatment effects. 2020.

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

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

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