THE CAUSAL TREE ESTIMATOR FOR HETEROGENEOUS TREATMENT EFFECTS: OPTIMAL DATA SPLITTING RULES IN SMALL SAMPLES

by

Nomin Margad-Erdene

Submitted to

Central European University

Department of Economics and Business

In partial fulfillment of the requirements for the degree of Master of Arts in Economics

Supervisor: Róbert Lieli

Abstract

Causal Trees leverage the supervised machine learning algorithm decision trees to estimate heterogeneous treatment effects across data-driven groups in a randomized treatment assignment setting. In my thesis, I modify the Causal Tree estimator by introducing a parameter θ that lets the user control allocation of data into training and estimation subsamples. The estimator implements "honest" sample splitting by default, which divides the sample into two equal parts: training and estimation subsamples. The new input parameter θ lets the user select the portion of data to be allocated to the estimation subsample. I test the performance of the estimator under various data allocations through Monte-Carlo simulations. My results suggest that in large samples $\theta \in [0.3, 0.7]$ can be an appropriate parameter value that minimizes the MSE of estimation. On the other hand, in small samples and in data sets with noise the recommended parameter range is $\theta \in [0.5, 0.7]$ with optimal value of $\theta = 0.6$.

Acknowledgements

I would like to thank my thesis advisor Róbert Lieli for his time and effort. While confusing at times, the thesis writing process has never been stressful thanks to your support, guidance, and prompt responses to all my questions. Also, thank you for teaching me my favorite course at CEU.

I am thankful to Alisher Batmanov and Ainura Karabayeva for their friendship and company during the past two years. Intimidating problem sets feel exciting and nerve-racking moments feel fun when you are with your friends.

Table of Contents:

1 Introduction

A new drug developed by an R&D firm may have an adverse effect on women of childbearing age only. A digital marketing company may find that customers of a certain age have a significantly high conversion rate for their advertisements (Wager and Athey, 2018). In cases like above, where an average treatment effect of an intervention or a policy is not informative enough we may be interested in estimating the heterogeneous treatment effect. HTE is observed when exposure to the same treatment results in varying effects in terms of the sign, magnitude, or both on individuals based on their characteristics. Estimation methods of HTE are gaining more and more interest not only in the economic and clinical research but in the corporate world too, where companies aim to make use of the availability of the data to make critical business decisions (Powers et. al., 2018).

However, researchers cannot exclusively rely on their expertise when finding relevant groups for heterogeneity of treatment effects, especially if the data is high dimensional. On one hand, the dimension of the data itself results in too many potential candidate groups and makes it nearly impossible to analyze without data-driven estimation tools. On the other hand, the process of searching for such groups invalidates the statistical inference. We want to avoid cases where the policymaker or developer mines through the data to find sub-groups where the effect is maximized and overstates the average treatment effect by testing on those specific sub-groups only (Cook et. al., 2004).

To address this problem, economists have utilized tree-based supervised machine learning algorithms such as Bayesian Additive Regression Trees (Green and Kern, 2012), Minimum Impurity Decision Assignment Trees (Laber and Zhao, 2015), Decision Lists (Lakkaraju and Rudin, 2017) and Random Forests (Foster et. al, 2011). One of the most notable solutions to the problem has been provided by Athey and Imbens (2016), where they have modified the random forest algorithm to construct trees that estimate treatment effect across data-driven groups in randomized experiments or observational studies where the unconfoundedness assumption is satisfied.

The central principle of the algorithm, which makes it possible to maximize the heterogeneity of treatment effect across groups without overfitting the model is the "honest" data splitting approach. With "honest" splitting, the data is divided into two mutually exclusive subsamples: training subsample S_{Tr} with N^{Tr} observations and estimation subsample S_{Est} with N^{Est} observations. The training subsample is used to fit the model and find groups across which the heterogeneity is maximized. However, the estimation subsample is used when estimating the treatment effect. By having two independent samples, we can avoid over-fitting and reduce bias (Athey and Imbens, 2016). However, the cost of this approach is that the sample size is effectively cut in half, as the original algorithm in R by Athey and Imbens[1] and following adaptations to python[2] allocate an equal number of observations to the two subsamples by default ($N^{Tr} = N^{Est}$) when constructing the tree.

One can easily imagine logistical and financial constraints researchers and companies may face that can result in a limited sample size. It is costly to run large-scale experiments. Given the limitations of data availability once we move from theory to application, cutting the sample size in half can be critical and lead to inaccurate estimation. However, having equal-sized training and estimation subsamples is not a requirement. Therefore, knowing where to allocate more data when met with constraints can help us improve the performance of the algorithm, especially when working with treatment effects where the ground truth cannot be observed.

While many recent works have modified the original Causal Tree algorithm, they focused on the tree fitting part and the training sample only by adding a penalty term to the minimization problem (Lechner, 2018) or by introducing a threshold to the individual level treatments (Tran and Zheleva, 2019). Causal Trees have been used in the applied literature as well: to estimate the effect of summer jobs on long-term employment (Davis and Heller, 2017), financing on investment (Gulen et. al., 2020), and E-Commerce Cart Targeting (ECT) on shopping patterns (Luo et. al. 2019).

[1]https://github.com/susanathey/causalTree

[2]EconML package by Microsof Research: (https://github.com/microsoft/EconML/tree/master/econml/dml); cforest package by Tim Mensinger: (https://github.com/timmens/causal-forest)

However, these works do not modify the original algorithm. To my best knowledge, the simple problem of data allocation stays unexplored at the moment.

My thesis contributes to the current literature by evaluating the performance of the Causal Tree algorithm under various allocations of data to training and estimation subsamples. I modify it by introducing a new parameter θ that allows the user to select the size of the estimation subsample S_{Est}. Then, I search for the value of θ that minimizes the mean squared error (MSE) of estimated conditional average treatment effects and the total MSE by running Monte-Carlo simulations with different size and noise levels of data. In addition, I report the bias and variance of the estimator for each value of θ on the parameter grid. By testing the estimator on simulated data where the true treatment effects are known, I will be able to make recommendations on data allocation when implementing the Causal Tree algorithm on various sample sizes in applied work.

The parameter $\theta \in (0, 1)$ represents the share of observations allocated to the estimation subsample from the total sample. When the user gives it as input to the modified Causal Tree, the size of the training sample is automatically selected as $1 - \theta$ and all observations not used in estimation are used in training. It is expected that the default Causal Tree split which is equivalent to my case with $\theta = 0.5$ will be a satisfactory choice in cases where the sample size is large and when there is little noise in the data. This is mainly because I expect large samples to allow for flexibility in the parameter choice of θ without deteriorating the accuracy of the estimation. However, when the sample size is small and/or when the data is noisy, allocation of data to the two subsamples may be critical in minimizing the MSE and maximizing the accuracy of estimation. This can be achieved by optimally selecting the parameter θ, the task which I will be investigating in my thesis.

The results of my Monte-Carlo simulations suggest that in large samples with little noise, a range of $\theta \in [0.3, 0.7]$ is acceptable and minimizes the MSE of ATE and the Total MSE. In other words, allocating from 30 % to 70 % of the whole sample to the estimation subsample gives us acceptably accurate estimation of HTE. However, in small and noisy samples the minimum MSE is achieved when $\theta \in [0.5, 0.7]$ suggesting that more observations should be allocated to

the estimation subsample. This is because bias is proportional to the value of θ, while variance is inversely proportional. However, the honest splitting method itself diminishes the bias significantly leaving variance as the main contributor to the MSE. As a result, the allocation that minimizes the MSE in small and noisy samples is found when $\theta \geq 0.5$.

The rest of the thesis is organized as follows: Chapter 2 describes the Causal Tree model setup and "Honest" split method. Chapter 3 illustrates the estimation process which consist of data generation process, definition of the reported statistics and Monte-Carlo simulation design. Results and robustness checks are presented in Chapter 4. Chapter 5 concludes, discusses limitations and possible extensions of the analysis. Figures which were omitted from the results are found in the Appendix. Python scripts of the modified Causal Tree algorithm and the Monte-Carlo simulation design can be found in my github repository[3].

[3]https://github.com/nominmar

2 Causal Tree Setup

Following the setup of the model proposed by Athey and Imbens (2016), I have N independent and identically distributed observations indexed as $i = 1, 2, ..., N$. Each observation is randomly assigned a binary treatment $D_i \in \{0, 1\}$ and I assume existence of a pair of Rubin's potential outcomes for each observation:

$$Y_i(D_i) = \begin{cases} Y_i(0) & \text{if } D_i = 0 \\ Y_i(1) & \text{if } D_i = 1 \end{cases} \tag{1}$$

Given an $(N \times K)$ matrix of covariates and under the assumption of unconfoundedness[4], which requires independence of the treatment assignment and potential outcomes conditional on covariates, the conditional average treatment effect is defined as:

$$\tau(x) = \mu_1(x) - \mu_0(x) \tag{2}$$

where:

$$\mu_1(x) = \mathbb{E}[Y_i(1)|X_i = x]$$

$$\mu_0(x) = \mathbb{E}[Y_i(0)|X_i = x]$$

Causal tree will partition the covariate space (**X**) until we reach a set of terminal nodes (leaves). Within each leaf, the predicted outcome and the estimated treatment effect $\hat{\tau}_i(\Pi, X_i, l)$ is constant for all observations. For example: given some sample where y: wage, x_1: gender, x_2: age, and D: employment training program, we may find 3 leaves: l_1: men under 45 years, l_2: men over 45 years, and l_3: women. The partitions (Π) in this case are: $\Pi_{x_1} = \{\{x_1 = male\}, \{x_1 = female\}\}$ and $\Pi_{x_2} = \{\{x_2 \geq 45\}, \{x_2 < 45\}\}$. Then, the estimated conditional average treatment effects are the difference in mean wages between treated and untreated within each of these leaves.

[4]This assumption is satisfied without conditioning on **X** if treatment is randomly assigned in an experimental setting and in my case where data is simulated.

2.1 The Honest Split:

With honest splitting, the Causal Tree algorithm performs the following steps:

1. Divide the sample into 2 mutually exclusive subsamples S_{Tr} with N^{Tr} observations and S_{Est} with N^{Est} observations
2. Use S_{Tr} to train a decision tree which predicts outcome $\hat{Y}_i$ given the vector of covariates, $\mathbf{X}$.
3. Use fitted tree to estimate treatment effects on S_{Est} subsample. Each observation in estimation subsample is passed through the tree and assigned to a leaf (terminal node) following the set of rules defined in part 2. In each leaf, calculate the conditional average treatment effect: $\hat{\tau} = \hat{\mu}_1(x) - \hat{\mu}_0(x)$ by finding difference in means of the treated and untreated observations.

Training subsample is used to build the tree by finding rules for partitioning the covariate space. However, the estimation subsample is used when estimating the treatment effect. Athey and Imbens (2016) define the criterion that is maximized by the algorithm as:

$$-EMSE_\tau(\Pi) = E_{X_i}[\tau^2(X_i;\Pi)] - E_{S_{Est},X_i}[V(\hat{\tau})^2(X_i, S_{Est}, \Pi)]$$

which can be estimated using:

$$-\widehat{EMSE}_\tau(S_{Tr}, N^{Est}, \Pi) = \frac{1}{N^{Tr}} \sum_{i \in S_{Tr}} \hat{\tau}^2(X_i, S_{Tr}, \Pi) - \Big(\frac{1}{N^{Tr}} + \frac{1}{N^{Est}}\Big) \cdot \sum_{l \in \Pi} \Big(\frac{S^2_{S_{Tr}, D=1}(l)}{p} + \frac{S^2_{S_{Tr}, D=0}(l)}{1-p}\Big)$$

The first part of the equation is the variance of estimated treatment effect across leaves. Second part of the equation is the uncertainty about these estimates expressed as the variance of these estimators. This estimator rewards the heterogeneity across leaves due to the first part and penalizes variance of the estimators via second part.

By introducing the parameter θ into algorithm, I do not make modifications to the $-EMSE_\tau$ criteria. The parameter only affects the number of observations: N^{Tr} and N^{Est}.

3 Estimation Design

There are a total of 3 types of samples: estimation and training subsamples are used when fitting and estimating the model. I indicate them as S_{Est} and S_{Tr}. Each have N^{Est} and N^{Tr} number of observations respectively. They are generated through one data generation process (DGP) and a total of $N^{Tr}+N^{Est}$ observations are given as input to the algorithm. Causal Tree will allocate them into two sub-samples according to the parameter value of θ. In addition, there is a test sample S_{Te} with 5000 observations, which is generated independently outside the Monte-Carlo simulations following the same DGP. Test sample is used to evaluate the performance of the algorithm. The statistics mentioned in Section 3.2 are calculated on the test sample.

3.1 Data Generation Process

I have designed three data generation processes with 2, 4 and 8 distinct conditional average treatment effects respectively. In all designs the treatment is assigned randomly with a probability P=0.5:

$$\text{for D = } \{0,1\} \begin{cases} Pr(D_i = 1) = 0.5 \\ Pr(D_i = 0) = 0.5 \end{cases} \tag{3}$$

For all 3 designs, the potential outcome follows the structure

$$Y_i = D \cdot \gamma(X_i) + \eta(X_i) + \epsilon_i \tag{4}$$

where $\gamma(x)$ is the part of the model accounting for treatment effect and $\eta(x)$ for mean effect. The heterogeneity of the treatment is independent of the covariates which enter the function $\eta(x)$, but the outcome depends on them. $X_i \sim \mathcal{N}(0,1)$ is a $(N \times K)$ vector of covariates independent of ϵ_i. For each of the designs below I consider cases with variance of error term $Var(\epsilon) = [0.01, 1.0, 2.5]$ to account for noise in the data and total number of observations $N^{Tr} + N^{Est} = [500, 300, 100]$ to

account for various data sizes. This gives me 27 variations of the data to test the model on.

The following indicator function maps the sign of the relevant covariate to the treatment effect:

$$\mathbb{I}(x) := \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases} \tag{5}$$

In the first design, the treatment effect depends on x_1 only. In the second design, the treatment effect depends on x_1, x_2 and their interaction. In the third design, the treatment effect depends on x_1, x_2 and x_3.

DGP 1:

$$Y_i = -1.5D + 3D \cdot \mathbb{I}_{\{x_1 \geq 0\}} + \sum_{k=2}^{5} x_k + e_i$$

Treatment effect: $\gamma(x) = -1.5 + 3 \cdot \mathbb{I}_{\{x_1 \geq 0\}}$

Mean effect: $\eta(x) = \sum_{k=2}^{5} x_k$

Average Treatment Effects Conditional on x_1

	$x_1 \geq 0$	$x_1 < 0$
τ	1.5	-1.5

DGP 2:

$$Y_i = -2D + 3D \cdot \mathbb{I}_{\{x_1 \geq 0\}} + D \cdot \mathbb{I}_{\{x_2 \geq 0\}} + D \cdot \mathbb{I}_{\{x_1 \geq 0 \ \& \ x_2 \geq 0\}} + \sum_{k=3}^{5} x_k + e_i$$

Treatment effect: $\gamma(x) = -2 + 3 \cdot \mathbb{I}_{\{x_1 \geq 0\}} + \mathbb{I}_{\{x_2 \geq 0\}} + \mathbb{I}_{\{x_1 \geq 0 \ \& \ x_2 \geq 0\}}$

Mean effect: $\eta(x) = \sum_{k=3}^{5} x_k$

Average Treatment Effects Conditional on x_1, x_2

	$x_1 \geq 0$	$x_1 < 0$
$x_2 \geq 0$	3	-1
$x_2 < 0$	1	-2

DGP 3:

$$Y_i = -5D + 6D \cdot \mathbb{I}_{\{x_1 \geq 0\}} + 2.5D \cdot \mathbb{I}_{\{x_2 \geq 0\}} + 1.5D \cdot \mathbb{I}_{\{x_3 \geq 0\}} + \sum_{k=4}^{5} x_k + e_i$$

Treatment effect: $\gamma(x) = -5 + 6 \cdot \mathbb{I}_{\{x_1 \geq 0\}} + 2.5 \cdot \mathbb{I}_{\{x_2 \geq 0\}} + 1.5 \cdot \mathbb{I}_{\{x_3 \geq 0\}}$

Mean effect: $\eta(x) = \sum_{k=4}^{5} x_k$

Average Treatment Effects Conditional on x_1, x_2, x_3

	$x_3 \geq 0$		$x_3 < 0$	
	$x_1 \geq 0$	$x_1 < 0$	$x_1 \geq 0$	$x_1 < 0$
$x_2 \geq 0$	5	-1	3.5	-2.5
$x_2 < 0$	2.5	-3.5	1	-5

3.2 Reported Statistics

Samples generated by the data generation processes above will be used in Monte-Carlo simulation described in the next chapter and the following statistics will be reported for each case.

MSE, Bias and Variance of Conditional Average Treatment Effects (CATE):

MSE of the estimated conditional average treatment effects is calculated by finding the squared difference between the estimated and true conditional average treatment effects and averaging it across all Monte-Carlo iterations:

$$\widehat{MSE}_{CATE} = \frac{1}{R} \sum_{r=1}^{R} (\widehat{CATE}_r(\mathbf{X}) - CATE_{True})^2 \tag{6}$$

Here, R is the total number of Monte-Carlo iterations. Number of $\widehat{MSE}_{CATE}$ to be reported depends on the data generation process and true average treatment effects. For example, I will have two estimated MSEs for the two true conditional average treatment effects when using DGP

1. In the first case $\widehat{CATE}_r(\mathbf{X})$ is the average treatment effect of observations in test sample where $x_1 \geq 0$ during the r-th iteration of MC simulation. In the second case, it is the ATE of observations where $x_1 < 0$.

Table 1: Number of estimated CATES for each DGP

DGP1:	2
DGP2:	4
DGP3:	8

Mean Sqaured Error estimates are decomposed into bias and variance terms:

$$\widehat{MSE}_{CATE} = \widehat{BIAS}^2_{CATE} + \widehat{VAR}_{CATE}$$

$$\widehat{BIAS}_{CATE} = \frac{1}{R}\sum_{r=1}^{R}(\widehat{CATE}_r(\mathbf{X}) - CATE_{True}) \tag{7}$$

$$\widehat{VAR}_{CATE} = \frac{1}{R}\sum_{r=1}^{R}(\widehat{CATE}_r(\mathbf{X}) - \overline{\widehat{CATE}})^2 \tag{8}$$

Total MSE, Toal Bias and Total Variance of Individual Treatment Effects:

In addition to the conditional average treatment effect statistics, I report the Total MSE, Bias and Variance. The algorithm estimates the treatment effect for each observation. Upon the end of Monte-Carlo simulations, I will have an array of size $(N^{Te} \times R)$ of estimated treatment effects. $\hat{\tau}_{ir}$ indicates the estimated treatment effect of i-th observation in test sample during the r-th iteration of the Monte-Carlo simulation.

- Total Bias:

  For each observation in the test sample $i \in S_{te}$, I average the estimated treatment effect $\hat{\tau}_r(X_i)$ across all Monte-Carlo simulations $r = 1, .., R$. It gives me a $(N^{Te} \times 1)$ vector, where each element is the estimated Monte-Carlo average of treatment effect corresponding

to an observation in the test sample. Then I subtract the true treatment effect $\tau(X_i)$ for each observation, take square and average it across the test sample. This will be the total squared bias:

$$\widehat{BIAS}_T^2 = \frac{1}{N^{Te}} \sum_{i \in S_{Te}} (\overline{\hat{\tau}}(X_i) - \tau_i(X_i))^2 \tag{9}$$

where:

$$\overline{\hat{\tau}}(X_i) = \frac{\sum_{r=1}^{R} \hat{\tau}_{ir}}{R} \quad \forall \ i \in S_{Te}$$

- Total Variance:

  For each observation in the test sample $i \in S_{te}$, I find the sample variance across Monte-Carlo simulations. It gives me a $(N^{Te} \times 1)$ vector, where each element is the variance of Monte-Carlo estimated treatment effects corresponding to an observation in the test sample. Then I find the average across test sample.

$$\widehat{VAR}_T = \frac{1}{N^{Te}} \sum_{i \in S_{Te}} \widehat{V(\tau)}_i(X_i) \tag{10}$$

where:

$$\widehat{V(\tau)}_i(X_i) = \frac{\sum_{r=1}^{R} (\hat{\tau}_{ir} - \overline{\hat{\tau}}_i)^2}{R} \quad \forall \ i \in S_{Te}$$

- Total MSE is found by summing the total squared bias and total variance:

$$\widehat{MSE}_T = \widehat{BIAS}_T^2 + \widehat{VAR}_T \tag{11}$$

The statistics above will be calculated for each value of θ on the parameter grid. The goal is to (1) find the value of θ that minimizes the MSE of estimated CATEs and Total MSE and (2) understand how the bias and variance behave as I change the value of θ.

3.3 Monte-Carlo Simulation Design

New training S_{Tr} and estimation S_{Est} subsamples are generated in each Monte-Carlo iteration. The test sample S_{Te} of size 5000 is generated outside the Monte-Carlo function once and is used for each iteration. The parameter of my interest θ is indicated as **est_size** in the script and takes values between 0.2 and 0.8 with a step size of 0.1. Flowchart 1 summarizes the python script for the simulation design, which is applied to all 3 data generation processes. Input parameters that are required to start the script are described in Table 2.

Table 2: Input Parameters

Name of the parameter	Name in the script	Values	Description
$N^{Tr} + N^{Est}$	**n**	500, 300, 100	number of observations in $N^{Tr} + N^{Est}$ sample
N^{Te}	**n_test**	5000	number of observations in test sample
θ	**est_size**	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	share of **n** devoted to the estimation subsample
R	**reps**	500	Monte-Carlo repetitions
$Var(e)$	**var_e**	0.01, 1.0, 2.5	variance of error term in DGP

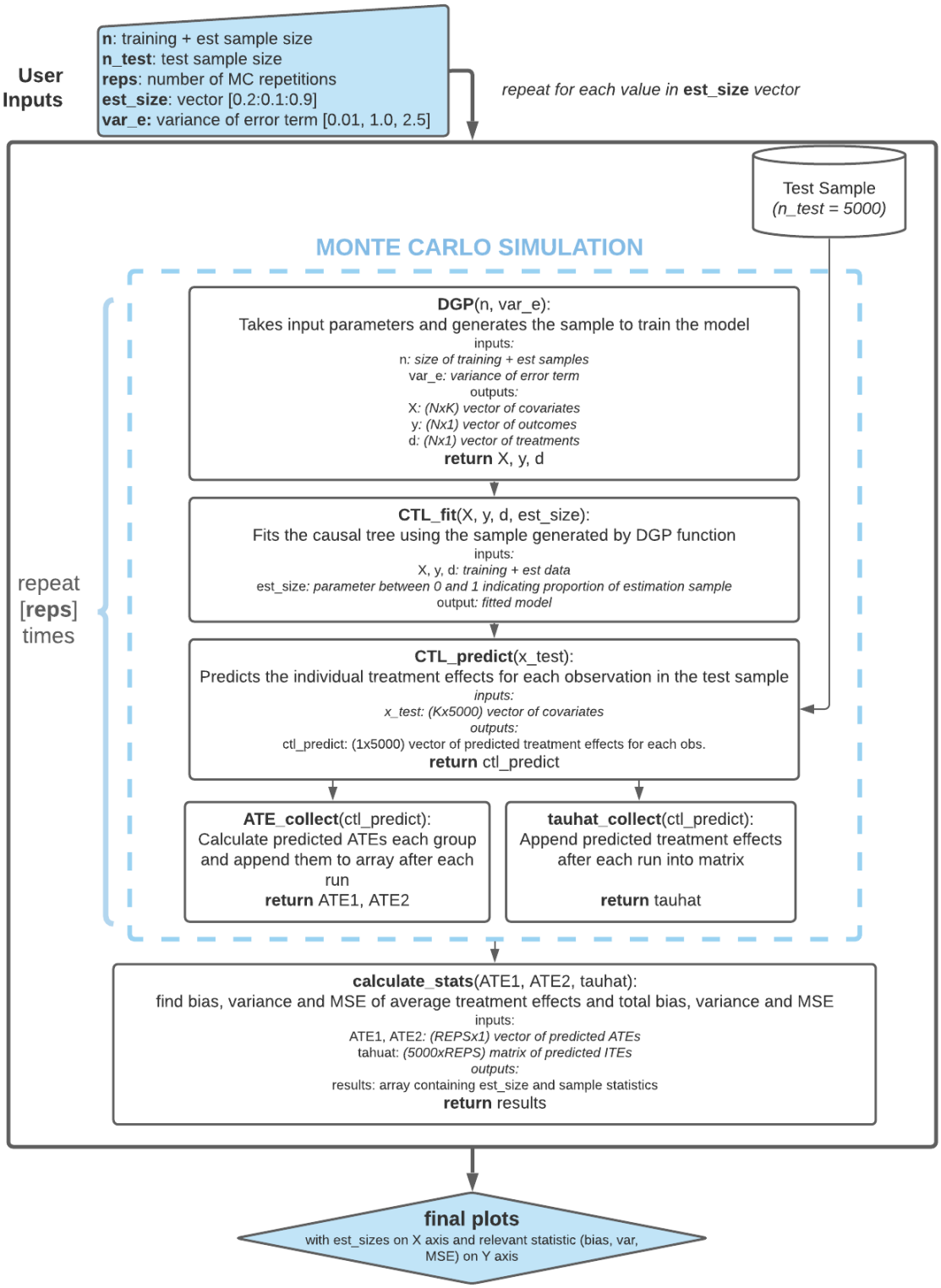
Monte-Carlo function (single iteration):

1. **DGP()** function takes the sample size parameter **[n]** and variance parameter **[var_e]** as inputs and generates a new sample with $N^{Tr} + N^{Est}$ observations.

2. **CTL_fit()** function takes the generated sample and uses the input parameter **[est_size]** to allocate it between training and estimation subsamples and fit the tree.

3. **CTL_predict()** function predicts the treatment effect (TE) for each observation in test sample N^{Te} and returns an array of predicted treatment effects $\hat{\tau}$ with size $(N^{Te} \times 1)$.

Two CTL functions are adapted from relevant parts of the Causal Tree Learn package by Tran and Zheleva (2019), which is a Python adaptation of the Causal Tree algorithm by Athey and Imbens



Flowchart 1: Monte-Carlo simulations and modified Causal Tree functions.

(2016) in R. Causal Tree Learn package is published with an MIT license which allows for free modification and publication.[5] I have modified the codes and added the parameter θ as user input. The reason for using their version is due to my preference of python over R. All other functions, as well as the Monte-Carlo simulation script is written by myself from scratch and are saved in my github repository.[6]

Monte-Carlo iterations:

Total number of MC iterations, $R = 500$ is given as input parameter **[reps]**. After 500 iterations, I will have collected an array $\hat{\tau}$ of size $(N^{Te} \times R)$ containing all predicted treatment effects. Using the them, I calculate the statistics described in equations (6-11) of Section 3.2. I run 500 Monte-Carlo iterations and calculate the statistics above for each value in the parameter grid of θ. Then, I plot each statistic against the parameter grid in the results section. The plots and results are described in the next section.

[5] https://github.com/edgeslab/CTL/blob/master/LICENSE
[6] https://github.com/nominmar

4 Results

This section consists of 3 parts. In the first part, I report the conditional average treatment effect statistics. In the second section, I report the Total MSE, Total Bias, and Total Variance. Finally, the third section discusses the robustness check process.

Figures in subsections 4.1 and 4.2 are organized as follows: parameter grid of θ is plotted on x-axis. These are the values that were given as input to the modified Causal Tree. On y-axis: Column 1 plots the bias, Column 2 plots the variance, and Column 3 plots MSE statistics. Row 1 presents DGP cases where $Var(e) = 0.01$, Row 2 presents cases where $Var(e) = 1.0$ and Row 3: $Var(e) = 2.5$. Each panel contains three cases: $N^{Tr} + N^{Est}$ = 500, 300 and 100 (distinguished by line styles). I only present the plots of DGP 1 in detail, as more complicated designs do not provide any additional contribution to the main results. Detailed plots for DGP 2 and DGP 3 can be found in the appendix section.

4.1 MSE, Bias and Variance of Conditional Average Treatment Effects (CATE)

Figure 1 presents the estimation results for DGP 1 which has two true conditional average treatment effects: $\tau_1 = 1.5$ and $\tau_2 = -1.5$. For the sake of brevity, I am presenting the averages of the two estimated conditional average treatment effect statistics. Detailed plots of each CATE can be found in the appendix section.

Bias-variance trade-off: panels on bias and variance columns indicate that as we move along the x-axis and change the parameter θ, we face a trade-off between bias and variance. The trade-off is milder in large samples and in cases with less noise. The slopes of solid lines in panels (1),(2),(4) and (5) are not steep. This also translates into MSE being uniformly low for most values of θ as we can see in panels (3) and (5) for the cases with $N^{Tr+Est} = 500$.

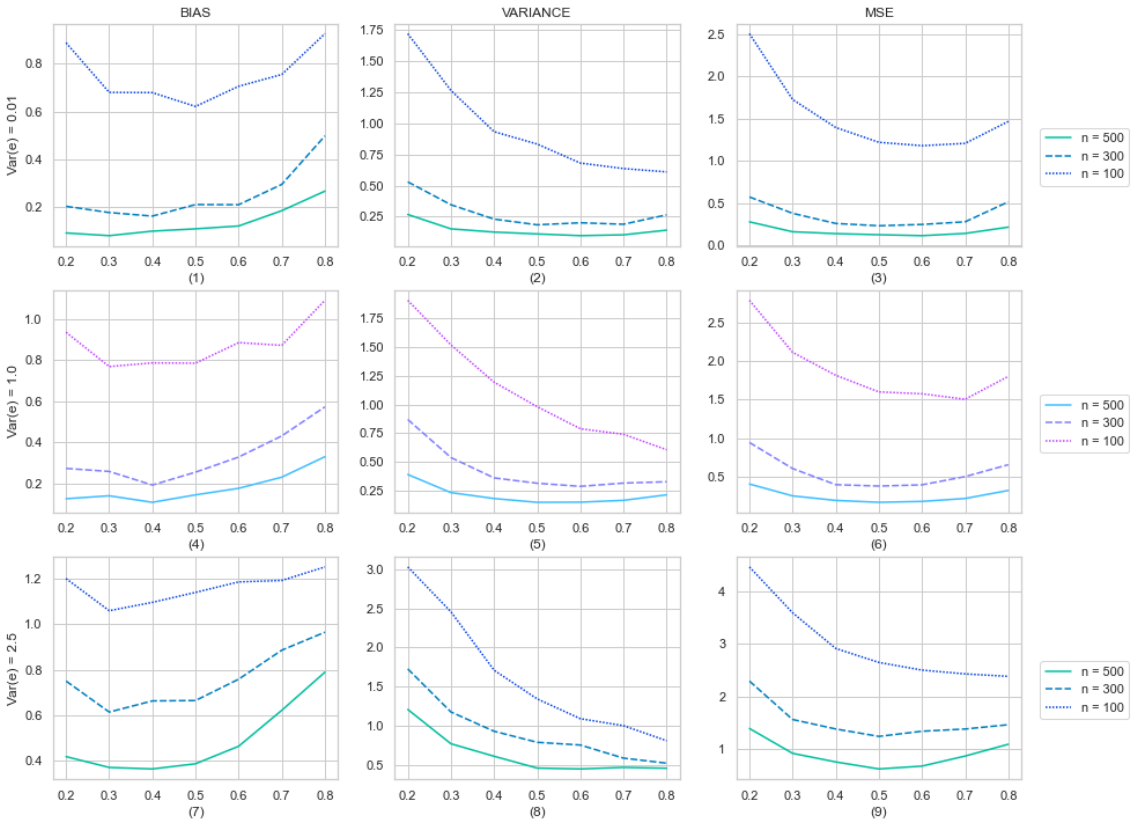


Figure 1: CATE bias, variance and MSE for DGP 1

Small samples: However, in small samples (ex. $N^{Tr+Est} = 100$) the trade-off only occurs on the right half of the x-axis. In cases where variance of error term is high, we can observe it even when $N^{Tr+Est} = 300$ as well. Both bias and variance decrease until θ = 0.5. Once I increase the share of estimation sample above 0.5, the trade-off kicks in. As a result, the minimum MSE is found when $N^{Tr} \leq N^{Est}$.

Optimal split: Panels in the third column suggest that in small or noisy samples indicated by dotted lines, MSE is minimized and optimal data allocation is found when $N^{Tr} \leq N^{Est}$ (when $\theta \geq 0.5$). This observation is consistent across all DGP processes, and is one of the main findings. On the other hand, in large samples and especially when $Var(e) \leq 1.0$, any non-extreme data allocation returns a satisfactory MSE. This can be seen by eyeballing the solid lines in panels (3)

and (6). Tables 3-5 below discuss these points in detail.

The first row of Table 3 shows the minimum value of MSE of CATE achieved in each of the 9 cases. Rows 2-4 indicate value of θ where MSE, variance and bias were minimized. Row 5 presents the standard deviation estimated MSEs. Standard deviations are calculated from 7 estimated MSEs each corresponding to a value of θ on parameter grid. Low standard deviation corresponds to a flatter line in plots, which further indicates flexibility in choosing the parameter. High standard deviations will indicate that we cannot diverge from the optimal θ without reducing the accuracy of the estimation. Row 6 presents the standard deviation of MSE estimates only for $\theta \in [0.3, 0.7]$ to remove the outliers.

Table 3: Summary of results (Design 1 with 2 ATEs)

N^{Tr+Est}	500			300			100		
$Var(e)$	0.01	1.00	2.50	0.01	1.00	2.50	0.01	1.00	2.50
Minimum MSE of ATE	0.113	0.170	0.618	0.230	0.380	1.237	1.180	1.503	2.380
θ where MSE is minimzed	0.6	0.5	0.5	0.5	0.5	0.5	0.6	0.7	0.8
θ where variance is minimzed	0.6	0.5	0.6	0.5	0.6	0.8	0.8	0.8	0.8
θ where bias is minimzed	0.3	0.4	0.4	0.4	0.4	0.3	0.5	0.3	0.3
SD of MSE ($\theta \in [0.2 : 0.8]$)	0.059	0.087	0.268	0.139	0.203	0.354	0.473	0.446	0.773
SD of MSE ($\theta \in [0.3 : 0.7]$)	0.018	0.033	0.125	0.059	0.097	0.117	0.230	0.247	0.472

In larger samples ($n = 500, n = 300$) the MSE is minimized at $\theta = 0.5$ indicating that original $N^{Tr} = N^{Est}$ works the best. However, we can also see that the standard deviation estimations are very low and any data allocation in range $\theta \in [0.3 : 0.7]$ will not deteriorate the accuracy of the prediction significantly. In small samples ($N^{Tr+Est} = 100$) the MSE is minimized at $\theta = 0.6$ if variance of error term is low and at $\theta = 0.8$ if it is high. In addition, the standard deviation is much higher, indicating that we cannot diverge from the optimal parameter without reducing estimation accuracy. The results are in line with plots in Figure 1.

Finally, the honest splitting method itself effectively combats bias because training and estimation are performed on 2 different subsamples. Therefore, contribution of variance to the MSE is

always higher than the contribution of bias. As a result, in small samples the MSE is minimized in the region where variance is minimized (to the right from $\theta = 0.5$).

Table 4: Summary of results (Design 2 with 4 ATEs)

N^{Tr+Est}	500			300			100		
$Var(e)$	0.01	1.00	2.50	0.01	1.00	2.50	0.01	1.00	2.50
Minimum MSE of ATE	0.514	0.572	0.978	0.675	0.817	1.622	1.916	2.595	4.759
θ where MSE is minimzed	0.6	0.5	0.5	0.5	0.5	0.5	0.6	0.6	0.6
θ where variance is minimzed	0.7	0.6	0.6	0.6	0.6	0.7	0.6	0.8	0.8
θ where bias is minimzed	0.2	0.2	0.2	0.2	0.2	0.2	0.4	0.4	0.3
SD of MSE ($\theta \in [0.2 : 0.8]$)	0.069	0.094	0.340	0.162	0.215	0.558	0.806	0.752	0.885
SD of MSE ($\theta \in [0.3 : 0.7]$)	0.041	0.032	0.120	0.052	0.073	0.234	0.373	0.343	0.488
θ where MSE1 is minimzed	0.3	0.4	0.5	0.4	0.4	0.5	0.5	0.4	0.6
θ where MSE2 is minimzed	0.6	0.7	0.6	0.8	0.6	0.6	0.7	0.7	0.8
θ where BIAS1 is minimzed	0.2	0.2	0.2	0.2	0.2	0.4	0.5	0.4	0.3
θ where BIAS2 is minimzed	0.7	0.5	0.5	0.5	0.4	0.4	0.5	0.4	0.3

Table 4 presents the results of estimation on DGP 2, which has 4 true conditional average treatment effects. Table 5 presentds results on DGP 3 with 8 conditional average treatment effects. Overall, the minimum MSE decreases when I have more conditional average treatment effects which a are close in magnitude (DGP 3) and interactions among covariates (DGP 2). However, we see that θ which minimizes the MSE is still between 0.5 and 0.6 in most cases. I do not report the $N^{Tr+Est} = 100$ case for DGP 3 in Table 5 because the sample size is too small to calculate 8 CATEs.

Table 5: Summary of results (Design 3 with 8 ATEs)

N^{Tr+Est}	500			300		
$Var(e)$	0.01	1.00	2.50	0.01	1.00	2.50
Minimum MSE of ATE	1.722	1.808	2.198	1.956	2.045	2.723
θ where MSE is minimzed	0.4	0.4	0.6	0.5	0.5	0.4
θ where variance is minimzed	0.6	0.6	0.6	0.6	0.6	0.5
θ where bias is minimzed	0.3	0.2	0.2	0.2	0.2	0.4
SD of MSE ($\theta \in [0.2 : 0.8]$)	0.075	0.093	0.237	0.083	0.140	0.562
SD of MSE ($\theta \in [0.3 : 0.7]$)	0.045	0.052	0.066	0.028	0.079	0.181

4.2 Total MSE, Bias and Variance

Figure 2 presents the plots of Total MSE, bias and variance displayed in equations 9-11.

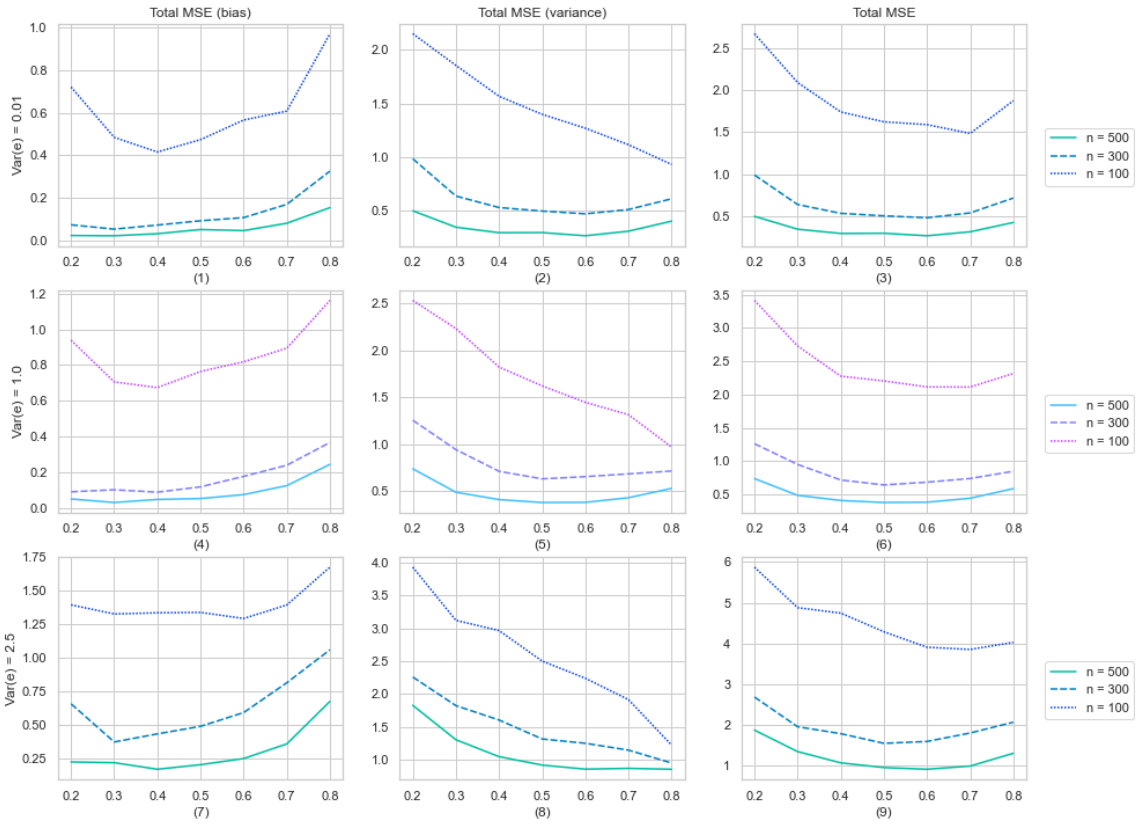


Figure 2: Total MSE, bias and variance for DGP 1

Visually, the results are similar with the conditional average treatment effect results presented in Figure 1. In addition, the magnitude of Total MSE is in line with the CATE MSE in previous subsection. I observe some bias-variance trade-off which is mitigated in large samples. In small samples, I can find more trade-off and MSE is minimized in the region to the right of the $\theta = 0.5$.

The monotonic fall in variance found in cases where the $N^{Tr+Est} = 100$ raises concerns about the accuracy of the simulations. I suspect that it is due to the estimator finding only one ATE in a small sample. When $N^{Tr+Est} = 100$ and $\theta = 0.8$, we only have 20 observations in the training

sample. The tree may have predicted the same $\hat{\tau}_i$ for all observations, which explains the small variance and high bias. I will address this issue in detail in the robustness check chapter. The results with $n \geq 300$ and $Var(e) \leq 1.0$ do not suffer from this issue and are reliable. Table 6 provides further details.

Table 6: Summary of results (DGP 1 Total MSE)

N^{Tr+Est}	500			300			100		
$Var(e)$	0.01	1.00	2.50	0.01	1.00	2.50	0.01	1.00	2.50
Minimum Total MSE	0.269	0.380	0.920	0.483	0.644	1.556	1.485	2.114	3.858
θ where MSE is minimzed	0.6	0.5	0.6	0.6	0.6	0.5	0.7	0.7	0.7
θ where variance is minimzed	0.6	0.5	0.8	0.6	0.5	0.8	0.8	0.8	0.8
θ where bias is minimzed	0.3	0.3	0.4	0.3	0.4	0.3	0.4	0.4	0.6
SD of MSE (θ = [0.2:0.8])	0.083	0.131	0.338	0.179	0.216	0.384	0.407	0.473	0.721
SD of MSE (θ = [0.3:0.7])	0.029	0.045	0.173	0.060	0.120	0.166	0.233	0.257	0.469

MSE is minimized in the region between 0.5 and 0.6 when $N^{Tr+Est} \geq 300$. When $N^{Tr+Est} = 100$, MSE is minimized at 0.7. To find out if some of the results are driven by fall in variance due to the model finding only 1 ATE instead of 2, I will check the number of leaves of the fitted tree in the next section. Overall, the trends found in tables 1-3 repeat here as well. We see that the standard deviation of total MSE also increases as we move on to smaller and noisy samples.

4.3 Robustness Checks

I can observe a sharp increase in bias after $\theta = 0.7$ in the plots. However, to find the exact value of θ after which the estimates are not reliable due to sample size, I plot the number of estimated leaves averaged across all Monte-Carlo simulations in Figure 3. As we can see below, when $\theta = 0.9$, the model finds only one ATE on average. When $\theta = 0.8$, the average number of leaves is 1.4 when $Var(e) = 2.5$ and 1.9 when $Var(e) = 2.5$. Therefore, the monotonic fall of variance is due to not having enough observations and results for $\theta \geq 0.8$ when $N^{Tr+Est} = 100$ are not reliable. However, this does not have effect on of $\theta < 0.8$ and the results discussed in previous sections.

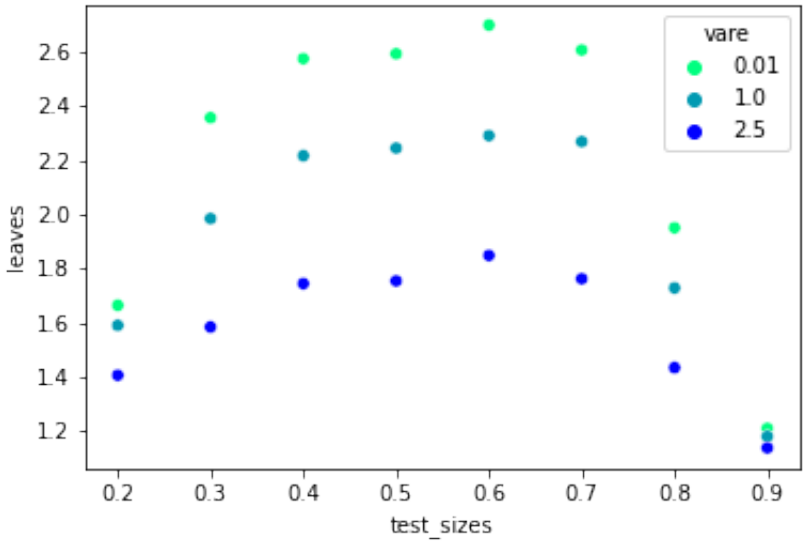


Figure 3: average number of estimated leaves for cases when $N^{Tr+Est} = 100$

5 Conclusion

Testing the estimation methods of HTE is proven to be tricky because we cannot observe the true counterfactual. We can never know what would have been outcome for a person who has taken the drug if he had never taken it. However, data driven methods and supervised machine learning provide us with more and more tools to deal with HTE in high dimensional data. Causal Trees are one of the prime examples of how a blend of machine learning and causal inference can create new tools for HTE estimation.

This thesis evaluates the performance of the Causal Tree algorithm under various data allocations by introducing a new parameter θ that controls for the size of the estimation subsample. By designing Monte-Carlo experiments, I find the recommended values for θ in large and small samples. Finding the optimal θ in simulated data creates a reference value which can be used when

applying Causal Tree estimator to real data where the true treatment effects are not observed.

I design 3 data generation processes which create samples with distinct conditional average treatment effects. Then, I use the modified Causal Tree algorithm and test the accuracy of the estimator by running Monte-Carlo simulations. The accuracy of the estimator is measured by reporting the (1) MSE of conditional average treatment effects and it's bias-variance decomposition as well as (2) Total MSE and it's bias-variance decomposition. I search for the parameter value of θ that minimizes the above statistics on a parameter grid $\theta \in [0.2, 0.8]$.

Results of the Monte-Carlo simulations suggest two main findings. First, when the sample size is large and when the data is not noisy, setting the parameter $\theta = [0.3, 0.7]$ maxmimizes the accuracy of the estimation. Second, when the sample size is small and when the data has lots of noise, higher values of the parameter are preferred. Instead of the default 50/50 split where $N^{Tr} = N^{Est}$, parameter values in range $\theta = [0.5, 0.7]$ minimize the MSE statistics. This suggests that in small samples, we should allocate more observations to the estimation subsample to improve the accuracy of the estimator.

The reason for such findings is due to the "honest" splitting method which minimizes bias by dividing the sample into two exclusive groups. We use different samples for training and estimation of the model. Therefore, the main contributor to the MSE is variance. Since variance is inversely proportional to the parameter θ as seen in result plots, higher values of θ result in better estimation. However, bias is proportional to θ. This creates a bias-variance trade-off which is then minimized at $\theta = 0.6$ in small samples.

5.1 Limitations

While my results provide some diversity in sample size and noise of the data, they are still limited and cannot be generalized to every data in applied work. I only consider 3 different types

of conditional average treatment effects which are mapped by an indicator function. As a next step, it is worth considering treatment effects generated as a continuous function of covariates. The number of repetitions and the test sample size can be increased further when testing for different data generation processes. However, in my case, I found no improvement in the variance of Monte-Carlo simulations for higher values of repetitions and test sample size. Therefore, I have decided to use R = 500 and $N^{Te} = 5000$ which helped me save computing time. It is also worth noting that by adding the parameter θ, we are still in the framework of sample splitting. Therefore, one may be interested in testing other sampling methods such as bootstrapping or bagging to see if the performance of the estimator can be further improved.

6 References:

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

2. Cook, D. I., Gebski, V. J., Keech, A. C. (2004). Subgroup analysis in clinical trials. Medical Journal of Australia, 180(6), 289.

3. Davis, J., & Heller, S. B. (2017). Using causal forests to predict treatment heterogeneity: An application to summer jobs. American Economic Review, 107(5), 546-50.

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

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

6. Gulen, H., Jens, C., & Page, T. B. (2020). An application of causal forest in corporate finance: How does financing affect investment?. Available at SSRN.

7. Laber, E. B., & Zhao, Y. Q. (2015). Tree-based methods for individualized treatment regimes. Biometrika, 102(3), 501-514.

8. Lakkaraju, H., & Rudin, C. 2017. Learning cost-effective and interpretable treatment regimes. In Artificial Intelligence and Statistics, 166–175.

9. Lechner, M. (2018). Modified causal forests for estimating heterogeneous causal effects. arXiv preprint arXiv:1812.09487.

10. Luo, X., Lu, X., & Li, J. (2019). When and how to leverage e-commerce cart targeting: The relative and moderated effects of scarcity and price incentives with a two-stage field experiment and causal forest optimization. Information Systems Research, 30(4), 1203-1227.

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

12. Tran, C., & Zheleva, E. (2019, July). Learning triggers for heterogeneous treatment effects. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 5183-5190).

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

7 Appendix:

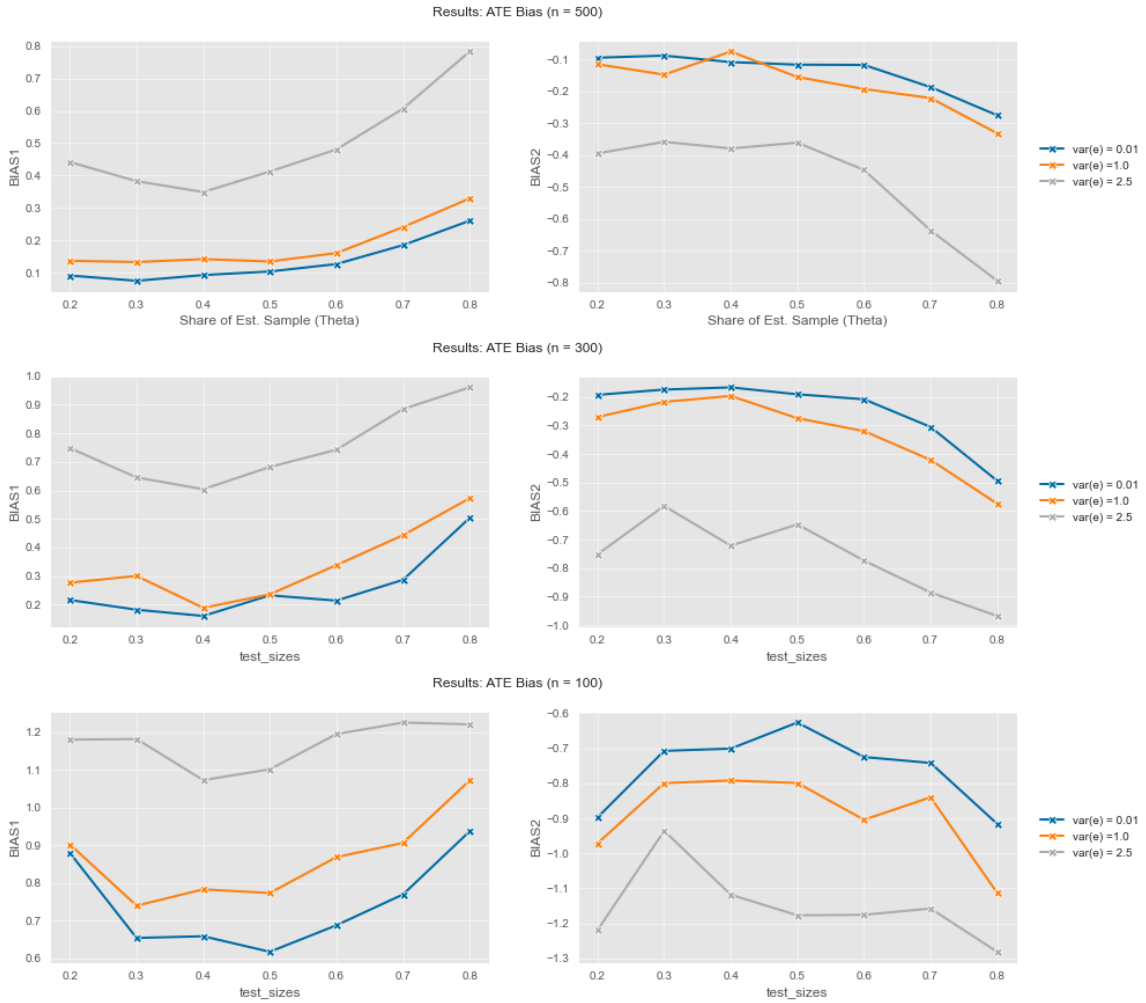


Figure A1: Bias of conditional average treatment effect, DGP 1

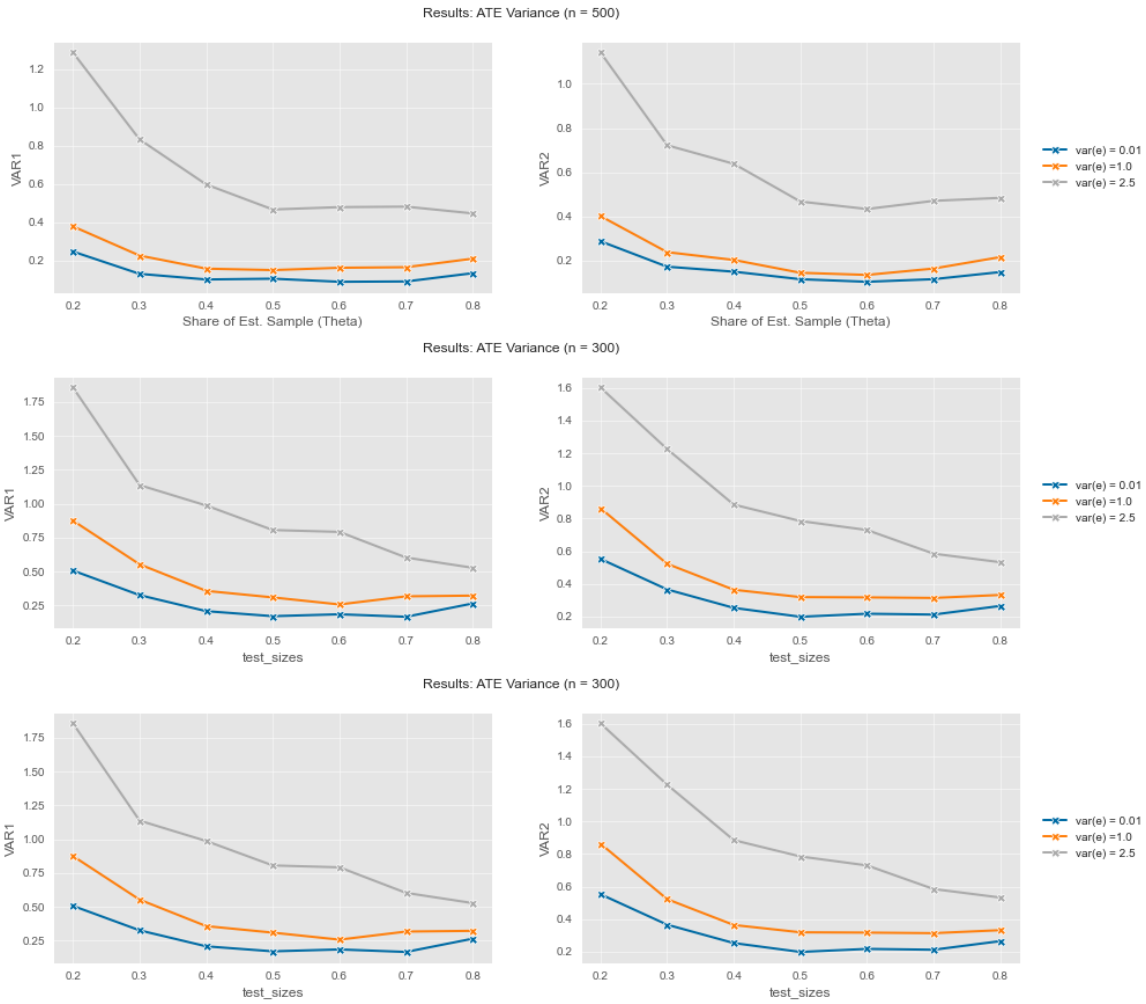


Figure A2: Variance of conditional average treatment effect, DGP 1

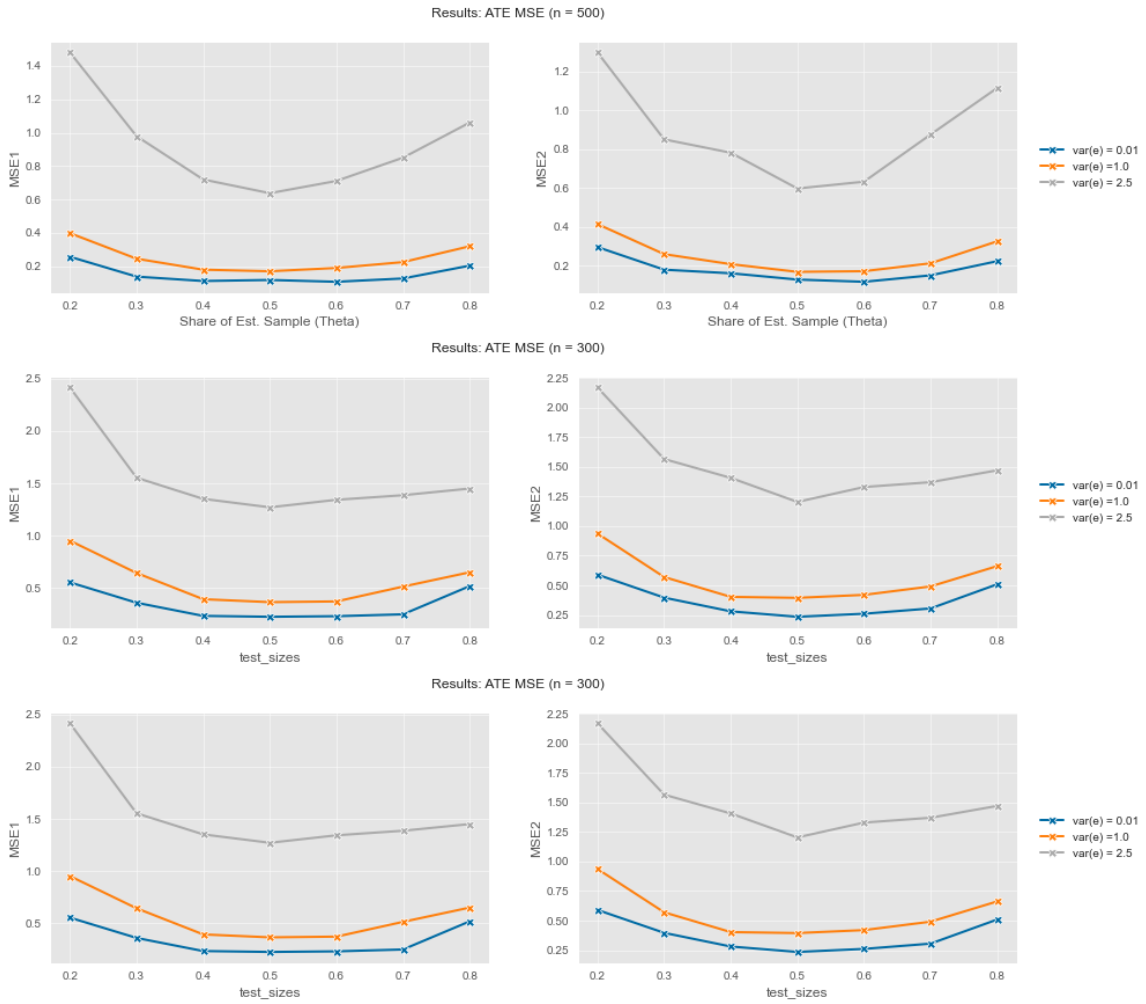


Figure A3: MSE of conditional average treatment effect, DGP 1

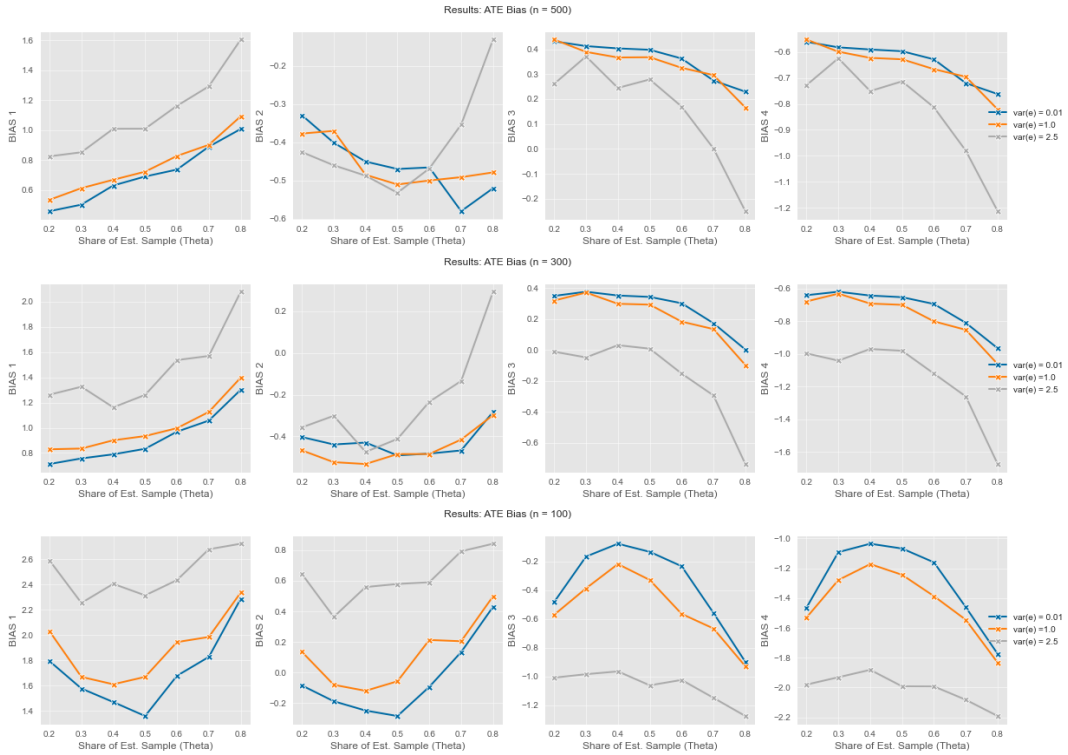


Figure A4: Bias of conditional average treatment effect, DGP 2

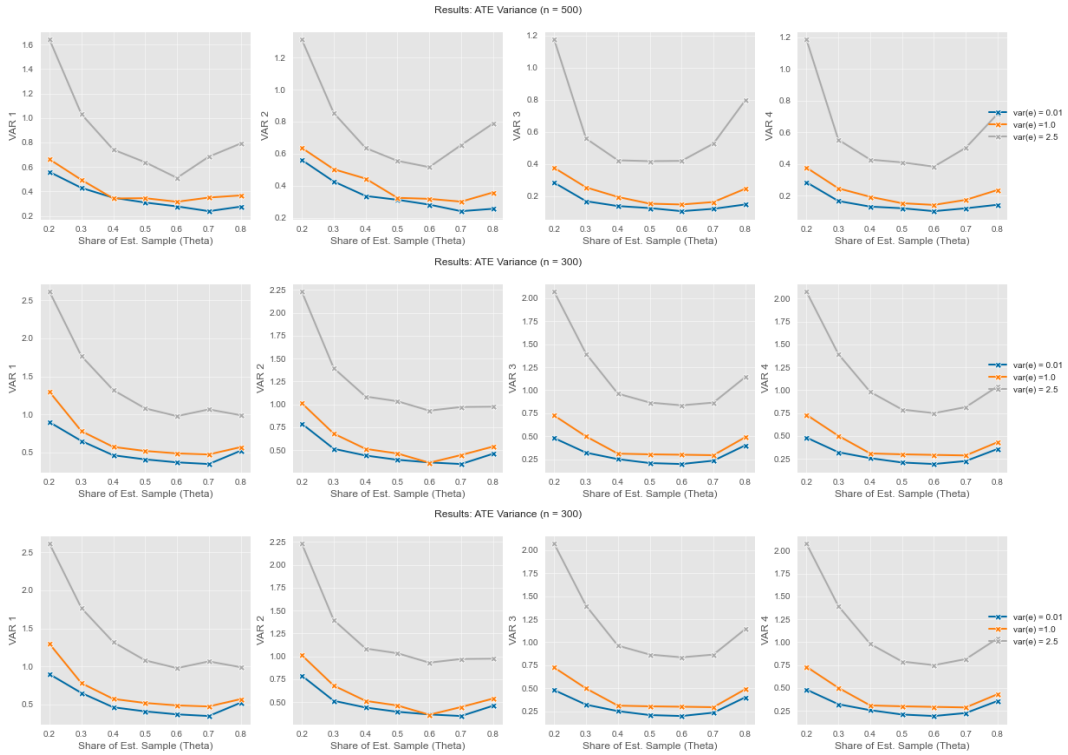


Figure A5: Variance of conditional average treatment effect, DGP 2

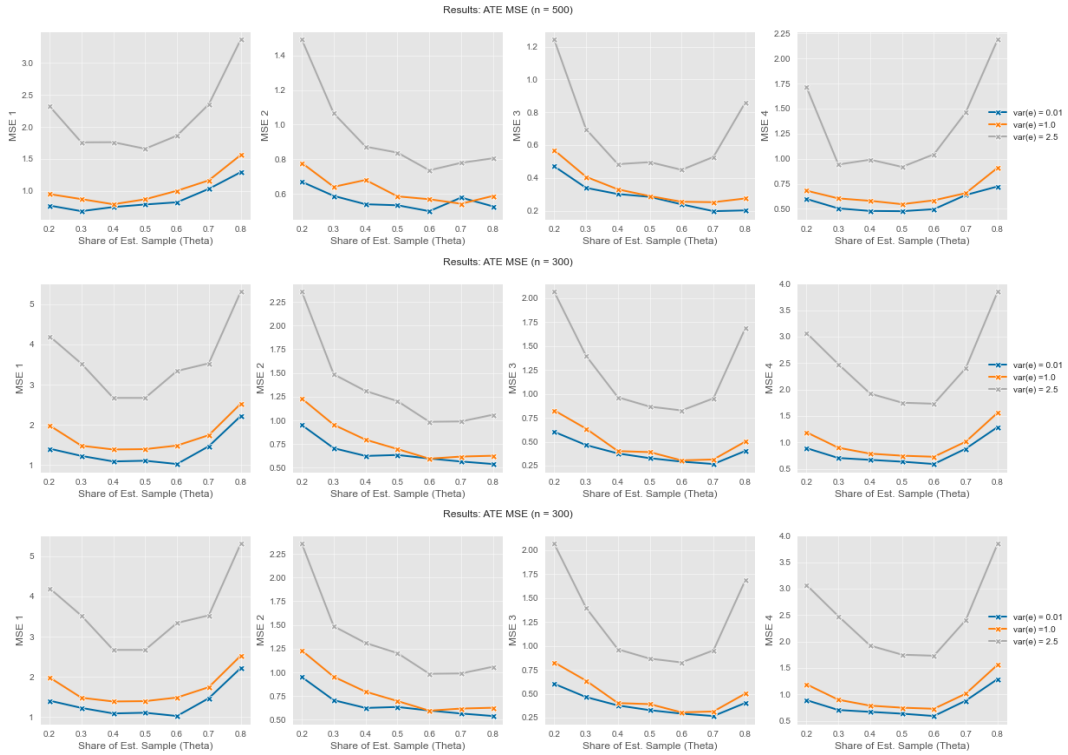


Figure A6: MSE of conditional average treatment effect, DGP 2

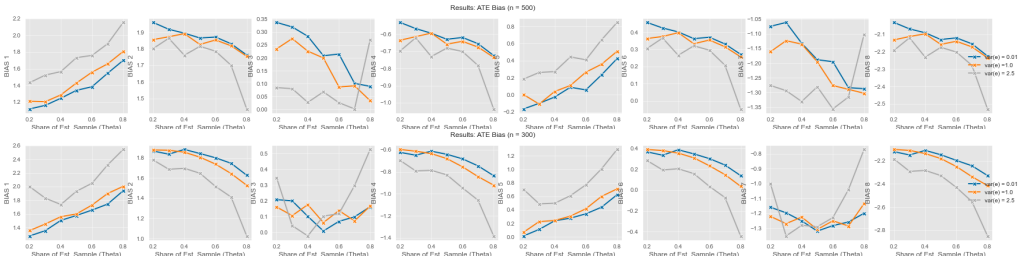


Figure A7: Bias of conditional average treatment effect, DGP 3

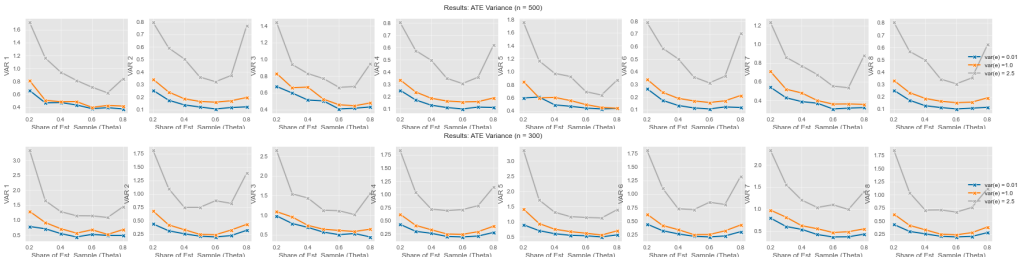


Figure A8: Variance of conditional average treatment effect, DGP 3

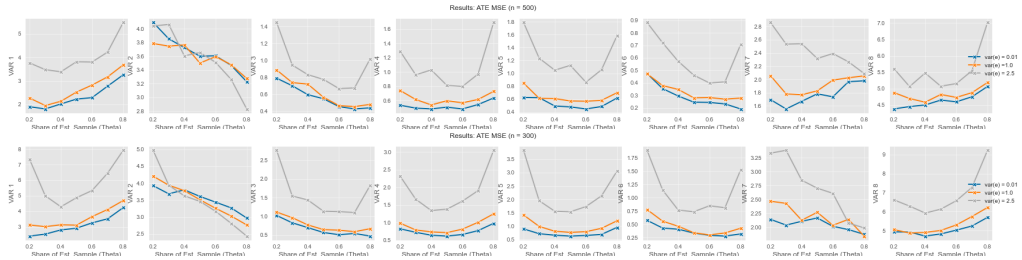


Figure A9: MSE of conditional average treatment effect, DGP 3