CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Estimating Causal Effects with Interpretable Decision Trees

Master's thesis in Computer science and engineering

Nicolas Audinet de Pieuchon

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2023

Master's thesis 2023

Estimating Causal Effects with Interpretable Decision Trees

Nicolas Audinet de Pieuchon

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2023

Estimating Causal Effects with Interpretable Decision Trees
Nicolas Audinet de Pieuchon

Supervisor: Fredrik Johansson, DSAI
Examiner: Peter Damaschke, Computer Science and Engineering

Master's Thesis 2023
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Description of the picture on the cover page (if applicable)

Typeset in LaTeX
Gothenburg, Sweden 2023

Abstract

In this work we explore three methods for estimating treatment effects from observational data using interpretable decision trees: the outcome variance tree, the propensity tree and the linear dependence tree. Each tree attempts to split the covariate space into balanced partitions from which treatment effects can be inferred. The outcome variance tree focuses on reducing the variance in the outcome variable, and makes use of a sensitivity analysis based on the residual standard deviation in the outcome. The propensity tree attempts to build a tree that approximates a separate estimate of the propensity score whilst remaining interpretable. The linear dependence tree measures the linear dependence in the partitions and attempts to minimize it directly. The three methods are compared, along with other benchmark methods, on two data sets: a synthetic data set generated from a simple model and the more realistic semi-synthetic IHDP data set. Performance is evaluated by comparing interval widths and coverage for confidence and sensitivity intervals. A functionally-grounded evaluation of interpretability is given with tree size as proxies. The results show that the outcome variance tree and the linear dependence tree perform better than the benchmarks in terms of sensitivity intervals but worse in terms of confidence intervals. The propensity tree however did not perform as well as expected and requires more work to better understand its behavior.

Keywords: decision trees, causality, interpretability

Acknowledgements

Thank you to Fredrik Johansson for being a patient and enthusiastic supervisor and for inspiring me to explore the field of causal inference. I thoroughly enjoyed working and learning together, and hope to continue in the future. Thank you to Kilian Freitag for offering support during the thesis and for offering to be my opponent. Thank you also to Sophie Auckram for your continued love and support. And finally a big thank you to my family for always being there for me and enabling me to do this fantastic Masters degree!

Nicolas Audinet de Pieuchon, Gothenburg, 2023-08-12

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

Causal inference is the act of estimating the causes and effects of phenomena from data. Estimating these is an important task in many fields such as genomics and personalized medicine [1] [2], economics and marketing [3], and recommendation systems [4]. Causal inference allows one to make better predictions of the effect of an intervention on a system, perform counterfactual estimation and deal with selection bias [5].

Cause and effect relationships can be estimated either from experimental data, for instance via Randomized Control Trials (RCTs), or from observational data. RCTs correspond to the classic scientific experiment: data is gathered in carefully controlled settings which allow one to make strong assumptions about the influence of external factors on the assignment of treatment and the outcome. However, in many domains it is often prohibitively expensive or outright impossible to perform such experiments for either practical or ethical reasons. Instead, researchers have to rely on observational data, which is passively collected and therefore does not have the same strong guarantees required for causal estimation. More specifically, observational data may be subject to confounder bias, where external factors may affect both treatment assignment and the outcome, as represented in Figure 1.1.

Several techniques have been developed for adjusting for confounder bias in observational data, each with their own set of advantages and disadvantages. Regression adjustment accounts for confounder bias by fitting a model to the outcome, but can suffer from model miss-specification and typically does not perform well on small sample sizes [6]. Propensity score weighting attempts to model how the covariates affect treatment assignment, and weigh the data accordingly [7]. These methods often rely on accurate models of the propensity score, and may overly exaggerate the importance of extreme samples, effectively reducing the sample size. Matching methods find similar pairs of individuals across treatment groups in order to make the distribution of the treatment and control group more alike [8]. Individuals are considered "similar" if they would have the same response when given the same treatment, which can be hard to measure from available data. These methods also often reduce the sample size when good enough matches are not found and tend to be sensitive to the distance metric used.

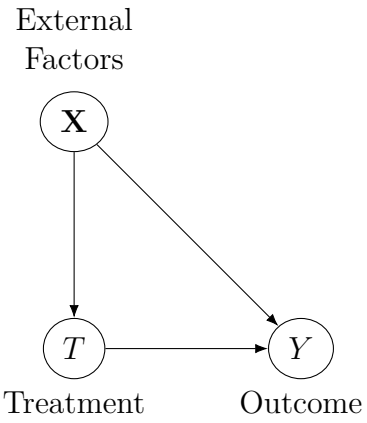


Figure 1.1: A causal graph with confoudning. Observed correlation between T and Y may be due to either the direct relationship between T and Y or by the effect of X on T and Y.

Another concern that many current techniques for causal estimation ignore is model interpretability. Broadly, interpretability is defined as "the ability to explain or to present in understandable terms to a human" [9] (although other definitions exist in the literature). Interpretability can be thought as the bridge between the formal objectives used to train the model and the real-world objectives the model encounters in deployment. Since these can differ significantly, it is important to design models to be understandable in order to trust that the model will meet the real-world objectives [10]. Decision tree models are often considered to be interpretable [11], especially when small and when the features are themselves easily interpretable.

In this project, we develop three methods for estimating causal treatment effects from observational data using interpretable decision trees: linear dependence trees, outcome variance trees and propensity trees. Each method uses decision trees with custom splitting and pruning criteria to partition the data such that each partition is unconfounded, or balanced (as if the data in the partition comes from a random experiment). The partitions learned by the tree are then used to compute the treatment effects and associated confidence and sensitivity intervals.

1.2 Context and Limitations

The application of decision trees and other tree-based algorithms from the machine learning world to causal inference has started to be explored in the past few years.

Causal trees are a partitioning estimator developed by Susan Athey and Guido Imbens to estimate heterogeneous treatment effects [12]. This work uses the potential outcome framework and assumes that the data is already unconfounded, so balancing the partitions is not considered. They approach the problem by adapting regression trees to optimize for goodness of fit in treatment effects and define an "honest" approach to estimation, where a sample is either used to construct the partitions or to estimate the treatment effect. This work was later extended to consider

causal forests, which adapt the random forest algorithm for heterogeneous treatment estimation [13], and to using R-learners in order to use observational data [14].

In "Bayesian nonparametric modeling for causal inference", Jennifer Hill demonstrates the use of tree-based techniques for estimating treatment effects which focuses on estimating the relation between the covariates and the outcome [15]. This technique is based on Bayesian Additive Regression Trees (BART) [16] which yield coherent uncertainty intervals and can deal with many covariates of different kinds. However, this is an ensemble tree model which could lead to models with many weak learners that are difficult to interpret.

A different method for estimating treatment effects from observational data with decision trees was proposed in a paper by Yahoo researchers [17]. This method focuses on the relation between the treatment assignment and the covariates instead of the relation between the outcome and the covariates. The algorithm they present involves several steps. First, a tree model is trained to predict treatment assignment based on covariates measured before the intervention. The tree is used to partition the covariate space to ensure constant propensity in each leaf in order to eliminate the dependence between treatment assignment and other factors. The training parameters for the tree are tuned using 10-fold cross-validation. Next, they compute the treatment effect within the leaves, which varies depending on the nature of the treatment assignment. Finally, the treatment effects are aggregated through weighted averaging to estimate the Average Treatment Effect. The paper also highlights the use of bagging (bootstrap aggregating) to enhance the model's robustness against small fluctuations in the data that could significantly impact the construction of a single tree. Bagging also provides a confidence interval for the ATE. However, the paper does not address sensitivity nor attempts to model it within the leaves. They also employ standard tree-based models without incorporating custom splitting criteria that optimize the sensitivity interval for the ATE. Finally, they make no mention of the interpretability of their model.

Causal Decision Trees use decision trees to uncover causal relationships from data (causal discovery) [18]. They are based on Causal Bayesian networks, but take advantage of tree-based to increase efficiency. Their goal is different from our own: to create a causal relationship tree which aims to provide interpretable and actionable information, rather than to estimate the causal effect of an intervention from data. However, they use decision trees in the same way by splitting the data according to a statistical test in order to obtain efficient and interpretable results.

2

Background

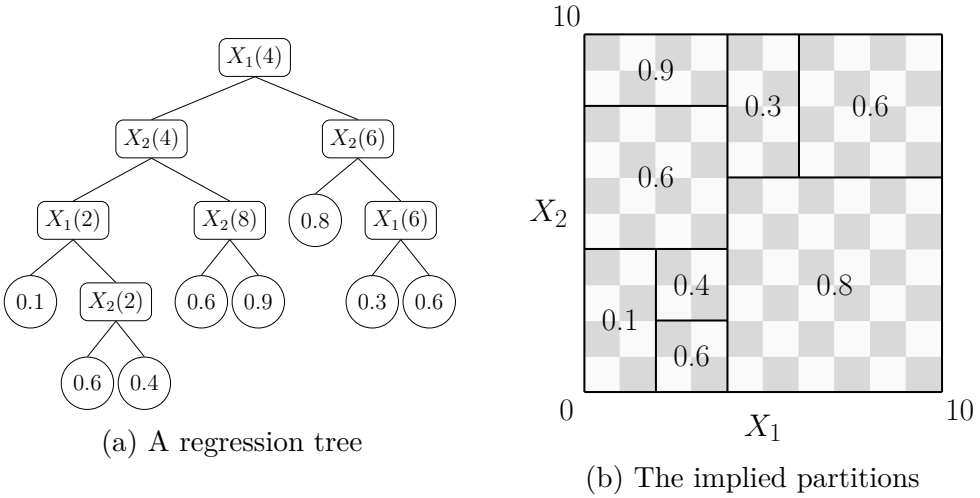
This chapter provides some of the necessary theoretical background which underpins the rest of the thesis. Section 2.1 introduces decision trees and describes a potential implementation in a regression context. Section 2.2 discusses the theoretical framework and the key assumptions behind causal effect estimation. Section 2.3 presents balancing and propensity scores as well as Inverse Propensity Weighting, a common method of confounder adjustment. Section 2.4 discusses two methods for computing confidence intervals, one based on the Central Limit Theorem and the other on bootstrapping. Section 2.5 discusses methods to deal with unobserved confounders, an important source of uncertainty in causal effect estimation. Finally, Section 2.6 contains an overview of interpretability in machine learning models.

2.1 Decision Trees

Decision trees are a simple but powerful machine learning technique that allows one to learn a piece-wise constant function from a data set. They are a non-parametric regression, in that the model form adapts to the data and no assumption is made about the underlying distribution of the data. Decision trees are typically trained in a supervised setting with a data set of input and label pairs $\{(\mathbf{x_i}, y_i)\}$. **Classification trees** are a subclass of decision trees where the output variable y is discrete (i.e. where each input belongs to a particular class). Decision trees where the output variable y is continuous are called **regression trees**. This work focuses on the latter.

The piece-wise constant function learned by regression trees can be thought of as a mapping from some d-dimensional input $\mathbf{x} \in \mathbb{R}^d$ to some continuous output $y \in \mathbb{R}$. The model first recursively splits the input space into contiguous partitions, and then assigns each partition a value learned from the training data. Inputs in the same partition are always mapped to the same value. Splits are represented by **branch nodes** with two children nodes which are stacked hierarchically to form a tree structure. Each node splits the partition represented by its parent into two by thresholding on a particular feature of the input data $f \in \{1..d\}$ with a threshold $t \in \mathbb{R}$. In other words, input samples $\mathbf{x} = (x_1, x_2, ..., x_d)^T$ where $x_f \leq t$ are sent to the left child, whereas samples where $x_f > t$ are sent to the right child. Splitting stops at **leaf nodes** which assign an output value to the partition defined by the path from the root node to the leaf node.

Figure 2.1 shows an example of a small decision tree. The tree maps inputs vectors with two integer features $\mathbf{x} \in [0, 10]^2$ to a continuous output $y \in [0, 1]$. This tree has 7 branch nodes (the boxes) and 8 leaf nodes (the circles). The top-most node is called the **root node** and represents the top-level split. In this case, the root node assigns inputs $\mathbf{x} = (x_1, x_2)^T$ where $x_1 \leq 4$ to the left child, and inputs where $x_1 > 4$ to the right child. The final partition of the input space given by the tree is represented in Figure 2.1b.



(a) A regression tree

(b) The implied partitions

Figure 2.1: Example of a small decision tree mapping a 2-dimensional integer input $(x_1, x_2) \in [0, 10]^2$ to a continuous output $y \in [0, 1]$. The left image shows the graph of the tree, with the root node at the top. The right image shows the partitions in the input space implied by the tree

Training optimal decision trees from a data is known to be NP-complete [19]. Therefore, instead of attempting to find an optimal smallest tree directly, decision trees are typically trained using a greedy algorithm. At each splitting step, the best-performing split is chosen by ranking all possible splits of the training data according to some **splitting criterion**. Splitting criteria measure the performance of a split by comparing the performance of the child partitions of the training data to the performance of the parent partition by some metric. Once a best split is selected, the algorithm then recursively repeats the splitting procedure on the two children nodes. The recursion stops when some stopping criteria is met in the leaf. A summary of this algorithm is given by Algorithm 1.

Once the tree is fully grown, the training data is partitioned into the various leaves of the tree and a constant prediction for the outcome is computed for each partition (the values observed in Figure 2.1b). When a decision tree receives a new input, it first finds the leaf the new input belongs to and then returns the outcome prediction for that leaf. For example, imagine we give a new input $\mathbf{x} = (6, 2)$ to the tree in Figure 2.1a. The model would then find the partition the new input belongs to (the large square one on the bottom right), and give a prediction of 0.8 for the outcome. In this way, decision trees act like a piece-wise constant approximation of the relationship between the inputs and their labels.

Algorithm 1: Greedy splitting algorithm

```
Function BuildTree(X, Y):
    split_best ← Criterion(X, Y)
    if split_best = None then
        o ← Predict(X, Y)
        return Leaf(o)
    end
    X_L, X_R ← Split(split_best, X)
    Y_L, Y_R ← Split(split_best, Y)
    node_L ← BuildTree(X_L, Y_L)
    node_R ← BuildTree(X_R, Y_R)
    return Branch(split_best, node_L, node_R)
Function Main():
    X, Y ← LoadData()
    return BuildTree(X, Y)
```

A common splitting criterion used to train regression trees is **variance reduction** [20]. This criterion aims to find splits that reduce the variance of the output variable Y as much as possible. Under assumptions of smoothness, decision trees with less output variance in the leaves will be more performant, since the constant approximation for the outcome in each leaf is more likely to be close to the true value of Y. Given the output values of the training data at a node, $Y = y_1, y_2, ..., y_N$, its variance can be estimated as:

$$Var(Y) = \frac{1}{|N|} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{1}{2}(y_i - y_j)^2$$

The variance reduction for a particular split s is then defined as the difference between the variance in the parent node and the average variance in the child nodes:

$$VR(Y; s) = Var(Y) - \frac{|Y_L|}{|Y|} Var(Y_L) - \frac{|Y_R|}{|Y|} Var(Y_R)$$

where Y are the output values in the parent node, Y_L are the output values in the left child and Y_R are the output values in the right child.

When using variance reduction as the splitting criteria, the algorithm will tend to want to continue splitting the training data into increasingly smaller partitions to reduce the variance as much as possible, potentially leading to overfitting [12]. Although the goal of decision tree learning is to find a tree that is large enough to approximate the distribution of the underlying data well, it also needs to be small enough to generalize well [21]. Pruning methods help reduce the complexity of tree models by removing redundant or non-critical nodes [22].

2.1.1 Pruning

Pruning methods are broadly split into two classes: pre-pruning and post-pruning methods. Pre-pruning methods introduce an early stopping criterion which prevents further splits while the tree is growing. These methods can suffer from the **horizon effect**, where potentially useful splits are not explored and the tree remains underfitted. Instead, Breiman et al [20] suggest post-pruning, a method which splits learning into two phases: a growing phase with loose stopping criteria (potentially allowing the tree to overfit the data), and a pruning phase which removes non-critical nodes. These methods tend to perform better than pre-pruning, since they allow the tree to explore structure which is not immediately apparent in the splitting criteria, but may be more computationally expensive [21].

A common pruning method for post-pruning is **cost-complexity pruning** [20]. In the first stage, a set of candidate trees $T_0, T_1, ..., T_k$ is constructed where T_0 is the original tree, T_k is the root node and each T_{i+1} tree is constructed by removing one or more nodes from T_i. Nodes are removed according to a **pruning criterion** which finds the least performant leaf node. In the second stage, the generalization error is estimated for each candidate tree and the best-performing one is selected. Figure 2.2 shows an example of cost-complexity pruning.

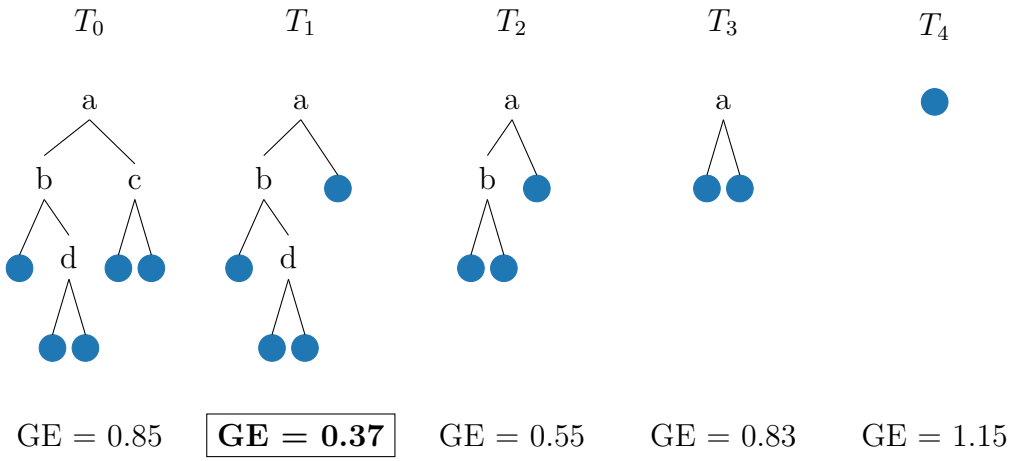


Figure 2.2: Example of post-pruning using cost-complexity pruning. T_0 represents the original tree that is being pruned. $T_0...T_4$ is the set of possible pruned trees generated in the first phase. GE is the estimate for the generalization error of each tree. In the second phase T_1 is chosen as the pruned tree as it has the best generalization error.

A drawback of regression trees is that creating splits based on thresholds may produce sharp borders between partitions (as seen in Figure 2.1b). This could lead to instability, where small differences in the training data can produce large changes in the regression tree. Random forests [23] are a more robust method which trains many simple trees and then aggregates the predictions.

2.2 Causal Effect Estimation

Imagine that DrugCorp, a pharmaceutical company, just developed a new treatment for some disease (a vaccine, for example). Before moving to production, they want to know whether the new treatment actually has the desired effect or not. To answer this question, the company could find a population of test units and start gathering some data. To measure the impact of the new treatment, they need to compare the health outcome of administering the new treatment to a unit with the health outcome when the treatment is not administered. In a perfect world, they would be able to have access to both outcomes *for each unit*. However, in practice this is not possible: it is not possible to measure the outcome of both administering and not administering a treatment on the same unit. This problem was termed by Holland as the Fundamental Problem of Causal Inference [24], and is what makes causal inference particularly tricky.

The potential outcome framework [25] is a well established framework for reasoning about this problem. Let $T \in \{0, 1\}$ be a random variable that indicates whether a unit was assigned the treatment. Let $Y \in \mathbb{R}$ be a random variable that indicates how well the patient recovered from the disease (larger Y is better). The potential outcome framework then denotes $Y_i(1)$ as the outcome measured for unit i had they been administered the treatment and $Y_i(0)$ as the outcome measured for unit i had they *not* been administered the treatment. The "had" is important, as $Y_i(1)$ and $Y_i(0)$ are counterfactual quantities that describe what could have happened rather than what actually happened. With this notation, the individual treatment effect can be described as the difference of the two potential outcomes:

$$ITE(i) = Y_i(1) - Y_i(0)$$

A positive ITE indicates that taking the treatment would have a net positive effect on unit i. The ITE can be aggregated into the average treatment effect (ATE) in order to find the treatment effect of the whole population:

$$\tau = \mathbb{E}[ITE(i)] = \mathbb{E}[Y_i(1)] - \mathbb{E}[Y_i(0)] \tag{2.1}$$

Causal effects can also be computed for a particular subpopulation described by some constraint on a set of **covariates**. Covariates are additional variables that are measured before treatment is assigned. These are modelled as a vector of random variables $\mathbf{X} = \mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_N$. For example, DrugCorp might be interested in understanding the effect of the new treatment on women between the age of 25 and 35. In this case, the two covariates would be the age $\mathbf{X}_1 \in \mathbb{I}^+$ and the sex $\mathbf{X}_2 \in \{male, female\}$ of each unit. The set of covariates would be $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2\}$. The average treatment effect in a subpopulation is known as the conditional average treatment effect (CATE) and is formally defined as:

$$\tau(\mathbf{x}) = \mathbb{E}[ITE(i) \mid \mathbf{X} = \mathbf{x}] = \mathbb{E}[Y_i(1) \mid \mathbf{X} = \mathbf{x}] - \mathbb{E}[Y_i(0) \mid \mathbf{X} = \mathbf{x}] \tag{2.2}$$

(a) T direct affects Y (b) $\mathbf{X}$ confounds T and Y

Figure 2.3: Figure demonstrating two possibilities of causal dependencies when an association is observed between a treatment assignment T and an outcome Y. In (a), T has direct causal effect on Y. In (b), a confounder $\mathbf{X}$ affects both T and Y, giving the illusion of a direct causal relation between the two.

However, the fundamental problem of causal inference is still at play here: $Y(1)$ and $Y(0)$ cannot be observed simultaneously for the same unit i or the same population! One way around this problem is to split the population into two groups: a treatment group which will receive the treatment and a control group which will not. Then, the outcomes of the two groups can be measured and compared to estimate the effect of the treatment.

To get a good estimate of the treatment effect using this method, it is imperative that the two groups be as similar as possible with respect to factors that affect the outcome, since they are supposed to represent the same population. For example, imagine that DrugCorp wants to assess whether their new treatment lowers the severity of the symptoms in sick patients. Let's say that age is an important factor for the severity of symptoms: older people tend to have worse symptoms than young people. If age also influences how the treatment and control groups are formed, for example by assigning the older half of the population to the treatment group and the younger half to the control group, then it becomes impossible to tell whether the observed difference in severity of symptoms between the two groups is due to the effect of the treatment or due to the age difference of the units. Such factors which affect both treatment assignment and outcome are called **confounders**, and are the reason behind the famous adage "correlation does not imply causation". Figure 2.3 shows a graphical representation of confounders.

Note that having factors that affect only the treatment assignment or the outcome but not the other do not impact treatment effect estimation. For example, assume hair color does not affect symptom severity. Then, by assigning treatments based on hair color would make no difference since people with different hair colors still appear the same to the process that generates the outcome.

One approach to prevent confounders from affecting the estimation of causal effects are **randomized control trials** (RCTs), where the treatment and control groups are chosen at random from the population. Choosing at random prevents all other factors from affecting treatment assignment, ensuring there is no confounding. The average outcome of the two groups becomes a good estimate for the average of the potential outcomes:

$$\mathbb{E}[Y(1)] \sim \frac{1}{N_1} \sum_{i:t_i=1} Y \quad \text{and} \quad \mathbb{E}[Y(0)] \sim \frac{1}{N_0} \sum_{i:t_i=0} Y$$

Provided there are enough samples, it becomes reasonable to compare the outcome of the two groups to estimate the treatment effects as shown in Equation 2.1 and 2.2. Correlation implies causation. Well-blinded RCTs are considered the gold standard for clinical trials and can provide strong evidence for causal effects [26].

Formally, RCTs work by guaranteeing two important properties: positivity and ignorability.

Assumption 1 (Positivity)**.** *Each unit under study has a chance of being treated and a chance of not being treated:*

$$0 \leq P(T = 1 \mid \mathbf{X} = \mathbf{x}) \leq 1 \qquad \forall \mathbf{x} \tag{2.3}$$

Positivity, also known as overlap or common support, guarantees that with a large enough sample size there will be units in both the treatment and control group for any subpopulation. This is important because treatment effects are defined as the difference between the outcomes of the two groups; if we only observe the outcome of one of the groups we have nothing to compare it to! This is guaranteed by RCTs since one can always set the probability of being assigned the treatment to be between 0 and 1.

Assumption 2 (Ignorability)**.** *Potential outcomes are independent of treatment assignment:*

$$Y(1), Y(0) \perp T \tag{2.4}$$

Ignorability, also known as unconfoundedness or exchangeability, guarantees that there are no confounders. If there were to be a confounder which affects both the treatment assignment and the outcome, then they would not be independent as they would both vary with the confounder. Ignorability is guaranteed by RCTs via the randomization process described earlier.

The third and final assumption necessary for estimating treatment effects is consistency:

Assumption 3 (Consistency)**.** *The observed outcome for a treatment is the same as the potential outcome had the unit been assigned to the same treatment:*

$$T = t \Leftrightarrow Y = Y(t) \qquad \forall t \in \{0, 1\} \tag{2.5}$$

Consistency ensures that the treatments are consistent, in that there is only one outcome per treatment assignment. In the DrugCorp study, consistency takes into account situations such as units not taking the new treatment even though they were assigned to the treatment group, maybe because of some external reason like what they read on the internet that day. Situations like these would bias the estimate for the average outcome of the treatment group, since there would be more than

one outcome per treatment assignment. Consistency also eliminates dependence between treatment assignment of one unit and the potential outcome of other units (interference).

These three assumptions - positivity, ignorability and consistency - allow the construction of an identification proof for the treatment effect, which relates the potential outcome to observable quantities:

$$\begin{aligned}
\mathbb{E}[Y(t)] &= \mathbb{E}_{\mathbf{X}}[\mathbb{E}[Y(t) \mid \mathbf{X} = \mathbf{x}]] && \text{by law of iterated expectation} \\
&= \mathbb{E}_{\mathbf{X}}[\mathbb{E}[Y(t) \mid \mathbf{X} = \mathbf{x}, T = t]] && \text{by conditional ignorability} \\
&= \mathbb{E}_{\mathbf{X}}[\mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}, T = t]] && \text{by consistency and positivity}
\end{aligned}$$

Given the assumptions and the identification proof, we can construct estimators for the ATE and the CATE which only use observable values for each unit: the covariates $\mathbf{X}$ (assumed to form an adjustment set), the treatment assignment T and the observed outcome Y.

$$\tau = \mathbb{E}[Y \mid T = 1] - \mathbb{E}[Y \mid T = 0]$$

$$\tau(\mathbf{x}) = \mathbb{E}[Y \mid T = 1, \mathbf{X} = \mathbf{x}] - \mathbb{E}[Y \mid T = 0, \mathbf{X} = \mathbf{x}]$$

In practice, these quantities are estimated as follows:

$$\hat{\tau} = \frac{N_1}{N} \sum_{i:t_i=1} y_i - \frac{N_0}{N} \sum_{i:t_i=0} y_i$$

$$\hat{\tau}(\mathbf{x}) = \frac{N_1}{N} \sum_{\substack{i:\mathbf{x}_i=\mathbf{x} \\ t_i=1}} y_i - \frac{N_0}{N} \sum_{\substack{i:\mathbf{x}_i=\mathbf{x} \\ t_i=0}} y_i$$

where N_1 is the number of units in the treatment group, N_0 is the number of units in the control group and N is the total number of units.

2.2.1 Observational Studies

Although useful for generating unconfounded data, RCTs may be prohibitively difficult or impossible to implement due to excessive cost, ethical concerns or other reasons. In these situations, one often has to rely on **observational studies** where data is gathered passively; that is, when the data generating process is out of the control of the experimenter. Since treatment assignment cannot be randomized, observational data may be confounded, which weakens the ignorability assumption. However, one can still achieve conditional ignorability for parts of the population by measuring and controlling for all relevant confounders. Then, one can estimate

the conditional average treatment effect for each part of the population, and aggregate the conditional estimates to find the average treatment effect for the whole population. In other words, it's as if the units in each sub-population came from a randomized experiment.

Assumption 2′ (Conditional Ignorability)**.** *The outcome is independent of the treatment effect given a set of covariates* $\mathbf{X}$*:*

$$Y(1), Y(0) \perp T \mid \mathbf{X}$$

A set of covariates $\mathbf{X}$ *for which conditional ignorability holds is called an* ***adjustment set****.*

Finding a valid adjustment set in practice can be challenging. One might be tempted to condition on as many covariates as possible in order to try to eliminate as much confounding as possible. However, it is not always the case that conditioning on more covariates decreases the amount of confoundedness in the data. This is known as collider bias [27]. Additionally, increasing the dimensionality of the covariate set weakens the positivity assumption due to the curse of dimensionality [28]. When the covariate space increases in dimensions, the number of possible combinations of the covariates also increases. This leads to the data being split into increasingly smaller partitions, since units are grouped by their covariates. In turn, this reduces the chances of groups containing units from both the treatment and control groups, which violates positivity. This is known as the Positivity-Unconfoundedness Tradeoff [29]. Section 2.3 discusses methods to reduce the dimensionality of the covariate space in order to combat this problem.

2.3 Balancing Scores and Propensity Methods

As discussed in the Section 2.2, estimating causal effects from observational data requires assuming conditional ignorability on a valid adjustment set. Under this assumption, units can be grouped by their covariates into partitions where the outcomes of the treatment and control groups can be compared directly. However, conditioning on the covariates directly may be difficult or even impossible if the covariates are too high-dimensional, since there will not be enough data to fill all the partitions meaningfully.

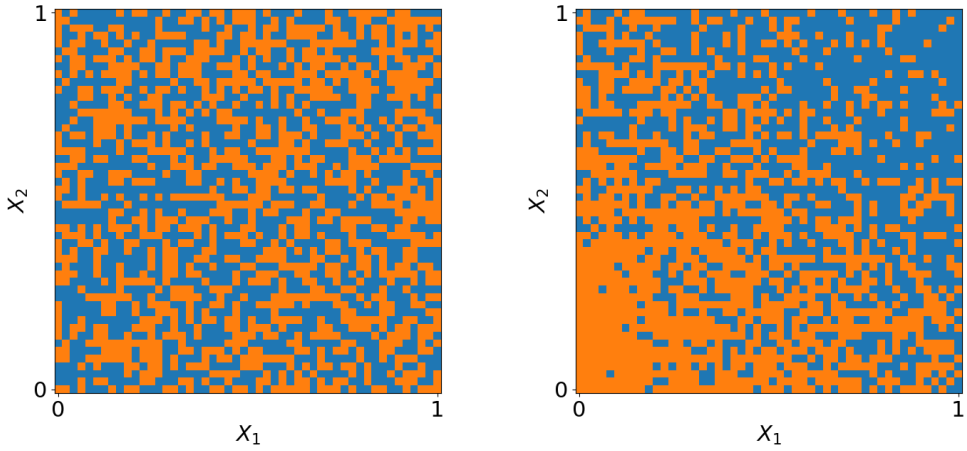
To get around this problem can use balancing scores [7]. Balancing scores are defined as a function of the covariates, $b(\mathbf{x})$, for which:

$$Y(t) \perp T \mid b(\mathbf{x}) \qquad \forall t \in \{0, 1\}$$

The simplest balancing score is the one-to-one identity function $b(\mathbf{x}) = \mathbf{x}$, which is always a balancing score by definition of conditional ignorability. More interesting balancing scores are many-to-one functions which group together some of the fine-grained partitions whilst maintaining conditional ignorability. It can be shown that the coarsest such function is the propensity score $e(\mathbf{x})$ [7], defined as the probability of assigning a unit to the treatment group given some covariates $\mathbf{x}$:

$$e(\mathbf{x}) = p(t = 1 \mid \mathbf{X} = \mathbf{x})$$

The shape of the propensity score tells us whether there is some causal dependence between the covariates and the treatment assignment. In an RCT, there is no association between the covariates and the treatment assignment, resulting in a uniform (flat) propensity score. A non-uniform propensity score implies an effect of the covariates on the treatment assignment. A visual example of the effect of the propensity score is given in Figure 2.4.



(a) $e(x) = 0.5$ (b) $e(x) = (x_1 + x_2)/2$

Figure 2.4: An illustration of the effect of the propensity score. Orange squares are units assigned to the control group, whilst blue squares are units assigned to the treatment group. In the left image, the propensity is constant (i.e. does not depend on the covariates). As one can see, units from all over the covariates space are assigned to both groups. In the right image, the propensity is linearly dependent on the two covariates, leading units in the control group to tend to be on the bottom left corner and units in the treatment group to tend to be in the top right corner. This would be a problem if $\mathbf{X}_1$ and $\mathbf{X}_2$ also affect the outcome, since it would be unclear whether the measured difference in outcomes between the two groups is due to the treatment or the covariate effect.

The propensity score allows one to side-step the Positivity Unconfoundedness trade-off by squashing the covariate space into a single dimension. This works particularly well when one has access to the propensity score directly, for example by knowing how the treatment was assigned in the data generating process. In most cases, however, the propensity score is not known beforehand and needs to be estimated. A popular method for estimating the propensity score are logistic regression models, but other methods such as neural networks or decision tree models have also been used [30].

Once acquired, the propensity score can be used to remove confounding effects from the data using **Inverse Propensity Weighting (IPW)** [31]. In IPW, one creates

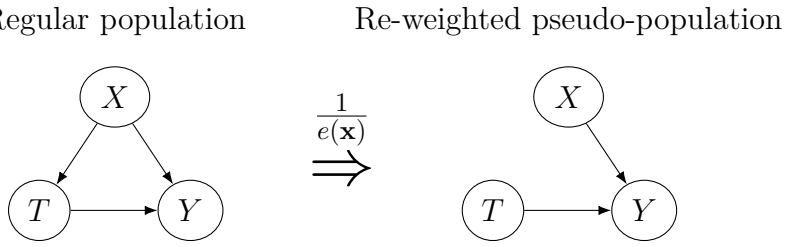


Figure 2.5: Diagram demonstrating the effect of Inverse Propensity Weighting.

a balanced pseudo-population by re-weighting the outcomes by the inverse of the propensity score. Multiplying by the inverse ensures that samples that had a low probability of receiving treatment are given more importance than samples with high probability of receiving treatment. In this way, the effect of the covariates on the treatment assignment in the regular population is reversed in the pseudo-population and conditional ignorability is restored. Figure 2.5 shows a graphical representation of the effect of IPW.

Mathematically, the re-weighting is described as:

$$\mathbb{E}[Y(t)] = \mathbb{E}\left[\frac{P(T=t)}{e(\mathbf{x})}Y \mid T=t\right]$$

Then, the ATE is given by:

$$\tau = \mathbb{E}[Y(1)-Y(0)] = \mathbb{E}\left[\frac{p(T=1)}{e(\mathbf{x})}Y \mid T=1\right] - \mathbb{E}\left[\frac{p(T=0)}{1-e(\mathbf{x})}Y \mid T=0\right]$$

2.4 Confidence Intervals

Managing uncertainty is important for any predictive endeavor. In frequentist statistics, uncertainty may arise from small sample sizes that don't reflect the underlying distribution well, or from the variability of the quantity of interest. In order to communicate this uncertainty, point predictions are often paired with a **confidence interval**. Confidence intervals are a range of estimates in which the model thinks that the true answer might lie. They are constructed based on a particular **confidence level**, which corresponds to the likelihood that the true value resides within the confidence interval. For example, for a confidence level of 95% one would expect the true value to be within the confidence interval 95% of the time.

In this project, we will consider two methods (among many) to compute confidence intervals for our point estimates of the ATE. Given a tree fitted on the training data set and a estimation data set, the goal is to generate a confidence interval for the global ATE. The major challenge for generating confidence intervals is that they need to be independent of the underlying distribution of the data since it is often not known. The estimation data set does not overlap with the training data set, following the honesty approach from Athey [12].

A popular approach for constructing confidence intervals is to make a **large-sample assumption**. This allows one to use the Central Limit Theorem, which says that, for independent and identically distributed samples with finite variance, the sample mean is normally distributed independently of the underlying distribution of the samples as the number of samples goes to infinity. Here, "large-sample" is intended to mean "large enough for the Central Limit Theorem to apply". Exactly how large this needs to be depends on the underlying distribution and is domain-specific. Normal distributions use standard deviations to describe the spread that the values of a random variable can take. A confidence interval can then be described in terms of the standard deviation. For example, an interval with a confidence level of 95% centered around a mean corresponds to saying that 95% of the time the true mean is expected to be about 2 standard deviations from the mean. However, often one does not have access to the true standard deviation of the distribution of the mean either. Instead, this can be approximated by the standard error, the standard deviation of the sampling mean.

An alternative method for constructing confidence intervals is **bootstrapping**, a methodology for computing standard errors and confidence intervals pioneered by Efron in 1979 [32]. Like the large-sample approaches, it is non-parametric and makes no assumption about the underlying distribution of the data. They also rely on assumptions of independent and identically distributed samples with finite variance. However, rather than using the Central Limit Theorem, they rely on a re-sampling approach to estimate the standard error of the statistic in question. The central analogy of these methods is: the population is to the sample as the sample is to the bootstrap samples [33, Chapter 21].

Given a sample $X = x_1, x_2, ..., x_N$ from an unknown distribution, bootstrapping methods have the following structure:

1. For $B = 1000$ times:

   (a) Create a bootstrap sample X_b by re-sampling with replacement from original sample X

   (b) Compute the relevant statistic for X_b

2. Compute the confidence interval from the bootstrap statistics

Bootstrap methods can be asymptotically more accurate than the standard large-sample intervals obtained with assumptions of normality [34]. Their plug-and-play nature also lends itself well to estimating bounds from complex sampling designs such as stratification [34]. However, bootstrapping methods can be computationally intensive, and only work if the original sample is large enough. More bootstrap samples do not give more information than is contained in the original sample, but can help to mitigate random variations. Bootstrap methods can also be more difficult to reason about theoretically.

2.5 Sensitivity Analysis

In causal effect estimation, noise may come from another source: **unobserved confounders**. As discussed in Section 2.2, causal effect estimation in observational studies requires the assumption of conditional ignorability. In turn, to justify this assumption, the researcher needs to observe enough confounders to form a valid adjustment set - but what happens when some of the confounders remain unobserved? Figure 2.6 shows a graphical representation of this problem: the unobserved confounders U are not contained within $\mathbf{X}$, yet still affect both treatment assignment T and outcome Y.

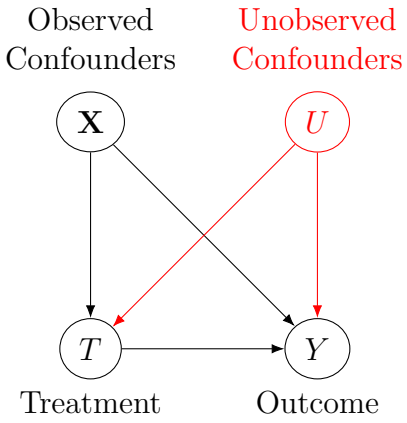


Figure 2.6: A causal graph with observed and unobserved confounders

Sensitivity analysis methods help researchers understand the robustness of causal estimates to unobserved confounders in observational studies by asking the question: how wrong do I have to be about my assumptions for my treatment effect estimate to be qualitatively different? [35] A well-known example of sensitivity analysis is a study by Cornfield et al. from 1959, in which they showed that the association between smoking and lung cancer was robust: an unobserved confounder, such as a genetic factor, would have to increase the probability of smoking nine-fold in order to explain away the association between smoking and lung cancer [36].

2.5.1 Additive Bias Model

One way to think about unobserved confounders is as a violation of conditional ignorability. With unobserved confounders, $\mathbf{X}$ is no longer a valid adjustment set and conditional ignorability with respect to $\mathbf{X}$ ceases to hold:

$$Y(1), Y(0) \not\perp T \mid \mathbf{X}$$

In turn, this implies:

$$\mathbb{E}[Y(t) \mid T = 1, \mathbf{X} = \mathbf{x}] \neq \mathbb{E}[Y(t) \mid T = 0, \mathbf{X} = \mathbf{x}] \qquad \forall t \in \{0, 1\}$$

The difference between the two equations above is known as the **confounding bias**. One approach to reason about the bias is to model is as an unknown additive quantity dependent on the covariates that balances out the two sides:

$$\mathbb{E}[Y(t) \mid T = 1, \mathbf{X} = \mathbf{x}] = \mathbb{E}[Y(t) \mid T = 0, \mathbf{X} = \mathbf{x}] - \eta_t(\mathbf{x}) \qquad \forall t \in \{0, 1\}$$

Some intuitions for what the biases $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$ represent (assuming that larger Y is better):

- $\eta_0(\mathbf{x}) < 0 \Rightarrow \mathbb{E}[Y(0) \mid T = 1, \mathbf{X}] < \mathbb{E}[Y(0) \mid T = 0, \mathbf{X}]$: the control outcome for people in the treatment group is worse than the control outcome in the control group. People in the treatment group were more likely to need the treatment since they would be worse-off without it. An example of a unobserved confounder with this effect could be age for heart-attack medication where the treatment group was on average older than the control group and therefore are more likely to need the treatment.
- $\eta_0(\mathbf{x}) < 0$ and $\eta_1(\mathbf{x}) < 0 \Rightarrow \mathbb{E}[Y(t) \mid T = 1, \mathbf{X}] < \mathbb{E}[Y(t) \mid T = 0, \mathbf{X}]$: regardless of whether people got the treatment or not, people in the treatment group are less healthy than their control group counterparts. Treatment is assigned to overall unhealthier people.
- $\eta_0(\mathbf{x}) < \eta_1(\mathbf{x}) \Rightarrow \mathbb{E}[Y(1) - Y(0) \mid T = 0, \mathbf{X}] < \mathbb{E}[Y(1) - Y(0) \mid T = 1, \mathbf{X}]$: the treatment effect for the treated group is larger that for the control group. The treatment was preferentially assigned to people for whom it would have a larger effect.

By quantifying the the bias in this fashion, one can derive equations for the expected value of the potential outcomes in terms of observable values and $\eta_t(\mathbf{x})$:

$$\mathbb{E}[Y(t) \mid \mathbf{X} = \mathbf{x}] = \mathbb{E}[Y \mid T = t, \mathbf{X} = \mathbf{x}] - \eta_t(\mathbf{x}) P(T = (1 - t) \mid \mathbf{X} = \mathbf{x}) \qquad (2.6)$$

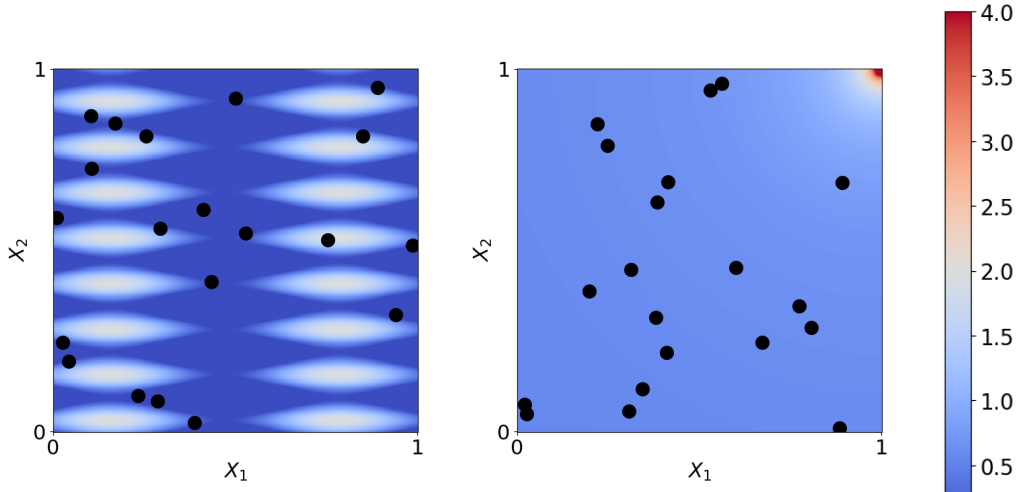
Intuitively, this equation say that, in the presence of unobserved confounders, the expected value of the potential outcomes are what would be observed under ignorability with some added bias scaled by the probability of being in the other outcome. The full derivation for the equations above is given in Appendix A.1. Notice that there is a sign change in the two equations because of how $\eta_0(\mathbf{x})$ is defined.

Let $\tau^*(\mathbf{x})$ be the naive conditional average treatment effect we would observe under conditional ignorability:

$$\tau^*(\mathbf{x}) = \mathbb{E}[Y \mid T = 1, \mathbf{X} = \mathbf{x}] - \mathbb{E}[Y \mid T = 0, \mathbf{X} = \mathbf{x}]$$

Then, Equation 2.6 can be used to find a definition for the conditional average treatment effect $\tau(\mathbf{x})$ in terms of $\tau^*(\mathbf{x})$ and the biases:

$$\tau(\mathbf{x}) = \tau^*(\mathbf{x}) - \eta_1(\mathbf{x}) P(T = 0 \mid \mathbf{X} = \mathbf{x}) - \eta_0(\mathbf{x}) P(T = 1 \mid \mathbf{X} = \mathbf{x})$$



(a) Smooth (b) Jagged

Figure 2.7: Comparison of a smooth vs. jagged outcome space. Diagram (a) shows a smooth space where the outcome does not vary quickly. In this case it reasonable to assume that the range of the samples (the black dots) will represent the range of the outcome well. Diagram (b) shows a jagged outcome space. In this case, the range of the samples is unlikely to represent the full range of the outcome since sampling the small region of covariate space with high values (the top-right corner) is unlikely.

Similarly, let $\tau^* = \mathbb{E}[\tau^*(\mathbf{x})]$ be the global average treatment effect under ignorability. Then we can decompose the global average treatment effect as follows:

$$\tau = \tau^* - \mathbb{E}[\eta_1(\mathbf{x})P(T = 0 \mid \mathbf{X} = \mathbf{x}) + \eta_0(\mathbf{x})P(T = 1 \mid \mathbf{X} = \mathbf{x})]$$

Yet, there's still a problem: the biases cannot be estimated directly, since they depend on unobserved confounders, which by definition there is no data for. One approach to solve this problem is to assume smoothness and regularity (absence of sharp spikes) in the outcome variable (see Figure 2.7 for an example). If these assumptions hold - and given enough samples - the samples collected are representative of the entire space of outcome, including the highest and lowest values of Y. The largest observed Y in the sample will be close to the maximum possible Y, and the smallest observed Y approximates the minimum Y. With these assumptions, we can then parametrize the unknown biases as a constant of the residual standard deviation $\sigma_t(\mathbf{x})$:

$$\sigma_t(\mathbf{x}) = \sqrt{\mathbb{V}[Y(t) \mid T = t, \mathbf{X} = \mathbf{x}]} \qquad \forall t \in \{0, 1\}$$

The biases can then be parameterized as follows:

$$\eta_t(\mathbf{x}) = \lambda_t \sigma_t(\mathbf{x}) \qquad \forall t \in \{0, 1\} \tag{2.7}$$

This parameterization follows a similar logic to Variance Reduction from Section 2.1. After adjusting for the observed confounders, any remaining variability in Y can be explained as a combination of noise and effects from the unobserved confounders. By parameterizing on the remaining standard deviation, this approach is making the pessimistic assumption that all remaining variability in the outcome is due to unknown confounders.

The parameterization of the biases can now be used to define the conditional average treatment effect $\tau(\mathbf{x})$ and the average treatment effect τ in terms of observable values, λ_1 and λ_0:

$$\tau(\mathbf{x}) = \tau^*(\mathbf{x}) - \lambda_1\sigma_1(\mathbf{x})P(T = 0 \mid \mathbf{X} = \mathbf{x}) - \lambda_0\sigma_0(\mathbf{x})P(T = 1 \mid \mathbf{X} = \mathbf{x})$$

$$\tau = \tau^* - \lambda_1\mathbb{E}[\sigma_1(\mathbf{x})P(T = 0 \mid \mathbf{X} = \mathbf{x})] - \lambda_0\mathbb{E}[\sigma_0(\mathbf{x})P(T = 1 \mid \mathbf{X} = \mathbf{x}))]$$

In practice, it is useful to work with a **sensitivity interval** which informs the user of the impact of the bias, similar to the confidence interval. The above definitions can be used to compute the sensitivity interval for the average treatment effect for various levels of λ_1 and λ_0:

$$SI = \left[\inf_{\lambda_1,\lambda_0} \tau, \sup_{\lambda_1,\lambda_0} \tau\right]$$

For example, one question of interest is: for what values of λ_1 and λ_0 does the sensitivity interval include potential values for the average treatment effect of both signs? That is, how large do the potential unobserved confounding effects need to be in order for one to draw the wrong conclusion from the data? This information can then be paired with other evidence in order to make better decisions, such as the Cornfield et al. [36] study linking smoking cigarettes with lung cancer.

2.5.2 Marginal Sensitivity Model

Another approach to sensitivity analysis is the Marginal Sensitivity Model, which instead of looking at the effect of the unobserved confounders on the outcome, uses the propensity score to explore the effect of unobserved confounders on the treatment assignment. Inverse Probability Weighting (Section 2.3) can be used to re-weight the observed outcomes based on the propensity score to eliminate confounding effects assuming that $\mathbf{X}$ is an adjustment set:

$$\mathbb{E}[Y(t)] = \mathbb{E}\left[\frac{P(T = t)}{e(\mathbf{x})}Y \mid T = t\right]$$

which in turn leads to the following estimator:

$$\mathbb{E}[Y(t)] \approx \frac{1}{n_t} \sum_{i:t_i=t} w_i y_i, \qquad w_i = \frac{P(T = t_i)}{e(\mathbf{x}_i)}$$

The key question here is: how does this estimate change in the presence of unknown confounders $\mathbf{U}$ (when X is not a valid adjustment set)? Let $e(\mathbf{x}) = P(T = 1 \mid \mathbf{X} = \mathbf{x})$ be the nominal propensity score which only includes observed confounders. Let $\bar{e}(\mathbf{x}, \mathbf{u}) = P(T = 1 \mid \mathbf{X} = \mathbf{x}, \mathbf{U} = \mathbf{u})$ be the complete propensity score which includes both observed and unobserved confounders. Assuming that $\mathbf{X} \cup \mathbf{U}$ forms a valid adjustment set, then IPW still holds for the complete propensity:

$$\mathbb{E}[Y(1)] = \mathbb{E}\left[\frac{P(T = 1)}{\bar{e}(\mathbf{x}, \mathbf{u})} Y \mid T = 1\right] = \mathbb{E}\left[\frac{P(T = 1)}{e(\mathbf{x})} \frac{e(\mathbf{x})}{\bar{e}(\mathbf{x}, \mathbf{u})} Y \mid T = 1\right]$$

In the case when $\mathbf{X}$ forms a valid adjustment set, then $e(\mathbf{x}) = \bar{e}(\mathbf{x}, \mathbf{u})$ and their ratio goes to 1. The impact of the unobserved confounders can be seen as the difference between the true $\mathbb{E}[Y(1)]$ and $\mathbb{E}^*[Y(1)]$ when $\mathbf{X}$ is assumed to be a valid adjustment set:

$$\mathbb{E}[Y(t)] - \mathbb{E}^*[Y(1)] = \mathbb{E}\left[\frac{P(T = 1)}{e(\mathbf{x})} \frac{e(\mathbf{x})}{\bar{e}(\mathbf{x}, \mathbf{u})} Y \mid T = 1\right] - \mathbb{E}\left[\frac{P(T = 1)}{e(\mathbf{x})} Y \mid T = 1\right]$$

Similarly to the additive bias model, the complete propensity $\bar{e}(\mathbf{x}, \mathbf{u})$ cannot be directly estimated since $\mathbf{U}$ is unobserved (by definition). Instead, one can bound the ratio $\frac{e(\mathbf{x})}{\bar{e}(\mathbf{x}, \mathbf{u})}$ by some parameter:

$$\frac{1}{\Gamma} \leq w(\mathbf{x}) = \frac{e(\mathbf{x})}{\bar{e}(\mathbf{x}, \mathbf{u})} \leq \Gamma$$

Like in the previous method, this parameter can be used to compute a sensitivity interval for the treatment effect estimate to explore its robustness to unobserved confounders:

$$\min_{w(\mathbf{x})} \mathbb{E}\left[\frac{P(T = 1)}{e(\mathbf{x})} w(\mathbf{x}) Y \mid T = 1\right] \leq E[Y(1)] \leq \max_{w(\mathbf{x})} \mathbb{E}\left[\frac{P(T = 1)}{e(\mathbf{x})} w(\mathbf{x}) Y \mid T = 1\right]$$

2.6 Interpretability

The objectives of the real-world tasks that machine learning models are created to solve are often at odds with the formal objectives used to design and train the model. These differences could occur when the offline training data distribution is not the same as the data distribution of the real-world task, for example when one attempts to predict events far in the future. Alternatively, the real-world task

might be difficult to formalize in the first place, leading to incomplete models. The real-world task might also introduce other more informal objectives, such as the ability of citizens to contest a decision that the model made or ethical choices in the model's decision making process, which become particularly important in contexts such as courtrooms or hospitals.

Lipton writes in "The Mythos of Model Interpretability" [10] that interpretability can mitigate this gap: if we can understand how the model works, we can begin to answer these difficult questions, which in turn lets us trust the model more. They propose to divide properties of interpretable methods into two classes: Transparency (how does the model work?) and Post-Hoc Explanations (what else can the model tell me?). Notions of transparency include simulatability, decomposability and algorithmic transparency. Sometimes, these be at odds with what we want from AI in the first place: to solve problems that are too difficult for humans. Notions of post-hoc explanation include text explanations, visual explanations, local explanations and explanation by example.

Doshi-Velez proposes a 3-tier taxonomy for evaluation of the interpretability of machine learning models [9]:

1. Application-grounded - human experiments with domain experts in a real application. The golden standard. These types of experiments are great to directly evaluate how a model will perform, but can be costly or outright impossible to perform.
2. Human-grounded - human experiments with laypeople in simpler applications. Good for more general notions of interpretability.
3. Functionally-grounded - use formal definition of interpretability (e.g. the size of a network) as a proxy for explanation quality. The most imprecise of the classes, but the least costly. Works best with model classes that have already been human-verified, such as decision trees or rule-based models.

3

Methods

The aim of the project is to develop methods to estimate causal treatment effects from observational data which result in an interpretable model. This presents two main challenges: to create interpretable models and to estimate treatment effects in the presence of confounders. Interpretability generally requires models to be small and simple enough for humans to understand, or to be able to effectively explain how they work. This adds an extra constraint to estimating causal effects that has not been explored by many existing models.

In this work, we focused on creating small and simple models that are relatively transparent to humans. To this effect, we base our methods on decision trees, a model class considered to be fairly interpretable [11]. Decision trees rely on the notion of model simulatability as discussed in [10], where a model is interpretable if a human can make inferences from the model and input data in reasonable time. They can be intuitively presented to human users in a visual manner, the decision path for making inferences is usually much shorter than the total model size, they often only use a subset of the features and leaves are mutually exclusive which reduces confusion. The inbuilt hierarchy can also give the user some idea of the relative importance of features, although it is argued that a better metric for feature relevance is to see how many units pass through a particular node instead [11]. For decision tree models to be interpretable, however, it is important that the user is able to understand the meaning behind the input features, which discounts certain types of feature pre-processing. Additionally, as noted in [10], decision trees are not inherently interpretable as sufficiently large or deep trees are not simulatable by humans in reasonable time. We will therefore use a functionally-grounded evaluation of the interpretability of our model based on the size and depth of the tree. Although it provides the weakest form of evaluation for interpretability, application-grounded and human-grounded evaluation methods were not considered in the interest of resources and time.

For our purposes, decision trees are used to identify balanced partitions of the covariate space from training data. Then, causal effects are estimated based on the balanced partitions and estimation data. From another perspective, the trees are used to automatically identify balanced strata of the population from which treatment effects can be estimated. The trees can also be understood as piece-wise constant balancing scores which map regions of covariate space to constant values. Larger trees will partition the covariate space more finely, and therefore have more

expressive power. However, this is balanced by our need for models to be interpretable: models have to be large enough to find balanced partitions, but small enough to still be interpretable.

Given the limited expressive power of the models and (potentially) limited amounts of data, the trees might not be able to find perfectly balanced partitions, leaving some residual confounding bias in the leaves. To take this bias into account, the final output of our methods includes a sensitivity interval as well as a point estimate for the global ATE. The sensitivity interval gives the user an idea of how much the confounding bias might be impacting the estimate of the causal effect.

In this work, we developed three separate methods for estimating treatment effects from observational data. All three methods do this by first training a decision tree and then estimating treatment effects and the sensitivity interval from the balanced partitions. However, the methods differ in their assumptions, approach to training the tree and in their sensitivity analysis. Section 3.1 introduces the common structure shared by the three approaches. Section 3.2 describes our first method which attempts to directly measure and minimize dependence in the leaves by reducing the correlation between the treatment assignment and the covariates. Section 3.3 discusses our second method which focuses on minimizing the outcome variance and uses the Additive Bias Model described in Section 2.5.1. Section 3.4 talks about our third method, which focuses on using the tree to estimate the propensity and uses the Marginal Sensitivity Model from Section 2.5.2. Finally, Section 3.5 discusses constructing confidence intervals from decision trees by applying the large-sample and bootstrapping techniques from Section 2.4.

3.1 Decision Trees for Causal Estimation

All three methods considered share a common structure: they are all variations of regression trees with post-pruning. However, the way regression trees are used differs slightly from the usual applications. The goal of most regression trees is to approximate a function from data (see Section 2.1). Once trained, the typical use-case is then to give a prediction of the function output for single new units. The regression tree model does this by categorizing each unit into a leaf and returning the associated value for the leaf learned during training. In our case, we don't use the trees to make predictions for unseen individual units. Rather, we use the model to group units from the available data into balanced partitions, which can then be used to estimate treatment effects. All three algorithms are therefore split into two stages, as represented in Figure 3.1:

1. **Learn Balanced Partitions** - Minimize the impact of confounders on treatment effect estimates by creating balanced partitions using decision trees and the training data.

2. **Estimate Treatment Effects** - Estimate a confidence and sensitivity interval for the population ATE by aggregating estimates of the ATE from the balanced partitions.

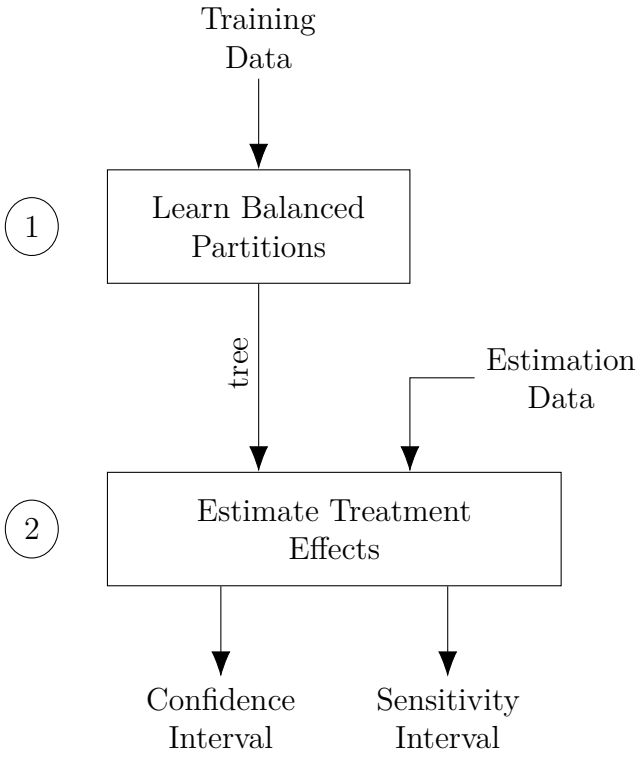


Figure 3.1: Flowchart of the top-level algorithm. The first phase takes training data as input and produces a tree which splits the data into balanced partitions to reduce the effect of confounders, The second phase estimates the Average Treatment Effect (ATE) and produces a confidence interval (CI) and a sensitivity interval (SI) from the balanced partitions and evaluation data.

This difference is also reflected in how the available data is used. In typical applications of regression trees, data is split into training used to learn the regression tree model and test data used to evaluate the performance of the model. In our application, however, we split the data into **training data** and **estimation data**. The training data is still used to learn the regression tree model. However, rather than providing an idea of how the model will perform on new data, the estimation data is used directly to estimate the causal treatment effects. It should be noted that it is also possible to use the training data directly to estimate the treatment effect. However, as Athey argues in [12], maintaining honesty facilitates analysis and eliminates some forms of bias leading to better coverage.

Finding balanced partitions by learning trees from training data is done with a greedy recursive algorithm in the same way regular regression trees are trained, as described in Section 2.1. When growing the tree, the only difference between the three methods is the **splitting criterion** used to decide which split is best for each branch. A simple early stopping criterion is used: a branch is not considered valid if it reduces the number of treatment groups in one of the children, or if the number of samples in a treatment group in one of the children goes below a pre-specified `min_group_size`. After the growing phase, cost-complexity pruning is used to shrink the tree. The total number of nodes in the tree is used as a proxy for model

complexity. Again, the only difference between the three methods in the pruning phase is the **pruning criterion** used to evaluate the performance of a particular node.

The second phase of all three methods consists of estimating the global average treatment effect from the tree and some estimation data. A tree can be described as a partitioning Π of a covariate space $\mathbb{X}$, where $\#(\Pi)$ is the number of partitions in the partitioning (the number of leaves in the tree):

$$\Pi = \{l_1, l_2, ..., l_{\#(\Pi)}\}, \quad \text{where} \quad \bigcup_{i=1}^{\#(\Pi)} l_i = \mathbb{X}$$

A leaf (or partition) is balanced if conditional ignorability holds given only that $\mathbf{X}$ is in the leaf. In other words, a leaf l is balanced if the following holds:

$$Y(t) \perp T \mid \mathbf{X} \in l \qquad \forall t \in \{0, 1\}$$

Given a partitioning Π, we can define the conditional average treatment effect in a particular leaf l as the expectation of the potential outcomes over the covariate partition described by l:

$$\tau(l) = CATE(l) = \mathbb{E}_{X,Y}[Y(1) \mid \mathbf{X} \in l] - \mathbb{E}_{X,Y}[Y(0) \mid \mathbf{X} \in l], \quad l \in \Pi$$

The global average treatment effect is given by the expectation of the conditional average treatment effect over all the leaves $l \in \Pi$:

$$\tau = ATE = \mathbb{E}_{\Pi}[\tau(l)]$$

Given estimation data and assuming conditional ignorability in the leaves, we can estimate the conditional average treatment effect $\tau(i)$, where $N_{l,t}$ is the number of samples in group $t \in \{0, 1\}$ in leaf $l \in \Pi$:

$$\hat{\tau}(l) = \frac{1}{N_{l,1}} \sum_{\substack{i:\mathbf{x}_i \in l \\ t_i = 1}} y_i - \frac{1}{N_{l,0}} \sum_{\substack{i:\mathbf{x}_i \in l \\ t_i = 0}} y_i$$

The average treatment effect τ can be estimated as follows, where N_l is the number of samples in leaf l and N is the total number of samples in the estimation data:

$$\tau = \sum_{l \in \Pi} \hat{\tau}(l) P(L = l) = \sum_{l \in \Pi} \hat{\tau}(l) \frac{N_l}{N}$$

However, in order to compute this estimate we need each leaf to have at least one sample in both the treatment and the control group. This is not guaranteed, as we only have a finite amount of samples in the estimation data. This is also

not guaranteed by the `min_group_size` hyper-parameter in the tree learning phase, since that only ensures that there are `min_group_size` samples in each group in each leaf for the training data, not the estimation data. To solve this problem, we further prune the tree. For each leaf in the tree, if the leaf does not contain enough samples in each group, then its parent is pruned. This effectively groups the samples from the leaf with the samples from its sibling leaf. This process continues until there are enough samples in each group in each leaf. A visual example of this algorithm is shown in Figure 3.2.

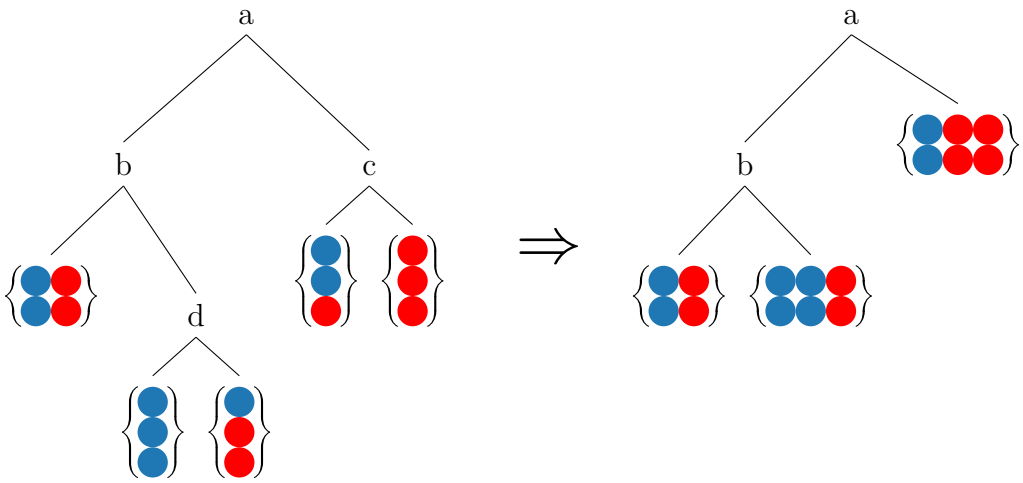


Figure 3.2: An example of pruning the estimation data set in order to have at least one sample per group per leaf. In this case, the estimation data is made up of 8 samples in the treatment group (blue) and 8 samples in the control group (red). Node d is pruned from the original tree (left) since there are no samples in the control group. Similarly, node c is pruned since there are no samples in the treatment group. The resulting pruned tree is depicted on the right.

3.2 Linear Dependence Trees

The goal of our methods is to find an interpretable tree that splits the covariate space into balanced partitions where $Y(t) \perp T \mid \mathbf{X} \in l$. The central idea behind our first method is to directly measure the dependence between the covariates and the treatment assignment in the partitions created by the tree, and then use the dependence measurement to grow and prune the tree.

We define a new splitting criterion, the **direct dependence criterion**, which computes the decrease in dependence between a covariate and the treatment assignment for a particular split s. Let $\mathbf{x}_c$ and $\mathbf{t}$ be the covariate c and the treatment assignment of the units in the parent node respectively. Let $\mathbf{x}_{c,L}$ and $\mathbf{t}_L$ be the covariate and treatment assignment of units in the left child node, and $\mathbf{x}_{c,R}$ and $\mathbf{t}_R$ be the covariate and treatment assignment in the right child node. Then, the direct dependence $DD(\mathbf{x}_c, \mathbf{t}; s)$ for a particular split s and a dependence measure $Dep : \mathbb{X}_c \times \mathbb{T} \to \mathbb{R}$ is given by:

$$DD(\mathbf{x}_c, \mathbf{t}; s) = Dep(\mathbf{x}_c, \mathbf{t}) - \frac{\#(\mathbf{t}_L)}{\#(\mathbf{t})} Dep(\mathbf{x}_{c,L}, \mathbf{t}_L) - \frac{\#(\mathbf{t}_R)}{\#(\mathbf{t})} Dep(\mathbf{x}_{c,R}, \mathbf{t}_R) \tag{3.1}$$

Note that here the dependence measure only measures the dependence between a single covariate and the treatment assignment. To find the best split, the dependence decrease is computed for all possible thresholds for each covariate at a node, and the split with the highest dependence decrease is chosen.

The pruning criterion is also defined in terms of the dependence measure. In this case, we want a measure that summarizes the relationship between all covariates and the treatment assignment. We define our pruning criterion $PC : \mathbb{X} \times \mathbb{T} \to \mathbb{R}$ as the average dependence in the node between covariates and the treatment assignment, where C is the total number of covariates and the average is taken with a uniform prior:

$$PC(\mathbf{x}, \mathbf{t}) = \frac{1}{C} \sum_{c=1}^{C} Dep(\mathbf{x}_c, \mathbf{t})$$

As for the dependence measure, there are many possible options. We opted for a measure based on the Spearman Rank Correlation Coefficient r_s, and defined our dependence measure as:

$$Dep(\mathbf{x}, \mathbf{t}) = |r_s| = |Spearman(\mathbf{x}, \mathbf{t})|$$

We chose to use the Spearman Rank Correlation Coefficient because it is simple and intuitive to understand, but has more expressive power than the Pearson Correlation Coefficient as it encompasses correlations between all monotonic functions rather than just linear functions. This measure of dependence however assumes that the dependence in the partitions is monotonic, which might not always be the case.

Another potential measure for dependence between covariates and treatment assignment is the Maximum Mean Discrepancy (MMD). For a particular leaf l, let $p_l(\mathbf{x}) = P(\mathbf{X} = \mathbf{x} \mid \mathbf{x} \in L)$ and $q_l(\mathbf{x}, t) = P(\mathbf{X} = \mathbf{x} \mid T = t, \mathbf{x} \in L)$. Assuming that the mean outcome belongs to a certain function class $\mathcal{F}$, $\mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}, T = t] \in \mathcal{F}$, then it can be shown that the MMD is connected to an upper bound based on the mean outcome:

$$\mathbb{E}[Y(t) \mid \mathbf{X} \in l] \leq \mathbb{E}[Y \mid T = t, \mathbf{X} \in l] + \mathrm{MMD}_{\mathcal{F}}(p_l(\mathbf{x}), q_l(\mathbf{x}, t))$$

See Appendix A.2 for the full derivation.

The MMD is a more powerful dependence measure than the Spearman Rank Correlation Coefficient, since one is not limited to define $\mathcal{F}$ as the class of monotonic functions. However, we didn't have enough time in our project to fully explore this avenue, so we decided to use the Spearman based measure of dependence instead.

As this method focuses on directly measuring violations of ignorability, we used the Additive Bias Model as the sensitivity model for this method.

3.3 Outcome Variance Trees

Rather than directly measuring the dependence between the outcome and the treatment assignment, outcome variance trees use the outcome variance in order to rank the performance of nodes by using variance reduction as the splitting criteria. Variance reduction also serves to optimize the sensitivity interval constructed by the additive bias model discussed in Section 2.5.1 since it seeks to minimize the variance in each leaf.

The additive bias model looks at the repercussions of violations of ignorability in terms of covariates $\mathbf{X}$. The central result was a decomposition of the true average treatment effect τ into the observed treatment effect under ignorability in terms of $\mathbf{X}$, $\tau^*(x)$, and a bias term dependent on the residual variance of the outcome. In other words, when $Y(t) \not\perp T \mid \mathbf{X}$:

$$\tau = \tau^* - \lambda_1 \mathbb{E}[\sigma_1(\mathbf{x}) P(T = 0 \mid \mathbf{X} = \mathbf{x})] - \lambda_0 \mathbb{E}[\sigma_0(\mathbf{x}) P(T = 1 \mid \mathbf{X} = \mathbf{x}))]$$

In our case, rather than violations of ignorability in terms of $\mathbf{X}$, we are interested in violations of ignorability in terms of the leaves of the tree: $Y(t) \not\perp T \mid \mathbf{X} \in l$. Therefore, we adapt the additive bias model to condition on the partitions rather than the individual $\mathbf{x}$s, effectively using a coarser balancing score. We define the additive bias in terms of the leaves:

$$E[Y(t) \mid T = 1, \mathbf{X} \in l] = \mathbb{E}[Y(t) \mid T = 0, \mathbf{X} \in l] - \eta_t(l) \qquad \forall t \in \{0, 1\}$$

Assuming consistency, this leads to the following decomposition of the average treatment effect:

$$\tau = \tau^* - \lambda_1 \mathbb{E}\Big[\sigma_1(l) P(T = 0 \mid \mathbf{X} \in l)\Big] - \lambda_0 \mathbb{E}\Big[\sigma_0(l) P(T = 1 \mid \mathbf{X} \in l))\Big]$$

where in this case τ^*, the average treatment effect under ignorability, is defined as an expectation over the leaves of a particular partitioning Π:

$$\tau^* = \mathbb{E}_\Pi[\tau^*(l)] = \mathbb{E}_\Pi\Big[\mathbb{E}[Y \mid T = 1, \mathbf{X} \in l] - \mathbb{E}[Y \mid T = 0, \mathbf{X} \in l]\Big]$$

and the variances $\sigma_t(l)$ are also defined in terms of the leaves:

$$\sigma_t(l) = \sqrt{\mathbb{V}[Y(t) \mid T = t, \mathbf{X} \in l]} \qquad \forall t \in \{0, 1\}$$

We can then compute estimates for τ from the estimation data:

$$\hat{\tau}^*(l) = \frac{1}{N_{l,1}} \sum_{\substack{i:\mathbf{x}_i \in l \\ t_i=1}} y_i - \frac{1}{N_{l,0}} \sum_{\substack{i:\mathbf{x}_i \in L \\ t_i=0}} y_i$$

$$\hat{\tau}^* = \sum_{l \in \Pi} \hat{\tau}^*(l) P(L = l) = \sum_{l \in \Pi} \hat{\tau}^*(l) \frac{N_L}{N}$$

$$\hat{\sigma}_t^2(l) = \frac{1}{N_{l,t}} \sum_{\substack{i:\mathbf{x}_i \in l \\ t_i=t}} y_i^2 - \frac{1}{N_{l,t}^2} \Big(\sum_{\substack{i:\mathbf{x}_i \in l \\ t_i=t}} y_i \Big)^2$$

$$\hat{\tau} = \hat{\tau}^* - \lambda_1 \sum_{l \in \Pi} \sigma_1(l) \frac{N_{l,0}}{N} - \lambda_0 \sum_{l \in \Pi} \sigma_0(l) \frac{N_{l,1}}{N}$$

Since $0 \leq \sigma_t(l)$ and $0 \leq P(T = t, L = l) \leq 1$ for all values of t and l, then the sign of the two bias terms is fully determined by their λ_t. If we let $\lambda_1, \lambda_0 \in [-\lambda, \lambda]$, where λ is a free parameter informed by domain knowledge, then we can easily compute the sensitivity interval as:

$$\overline{\tau} = \sup_{\lambda_1, \lambda_0} \hat{\tau} = \hat{\tau}^* + \frac{\lambda}{N} \sum_{l \in \Pi} \Big[\sigma_1(l) N_{l,0} + \sigma_0(l) N_{l,1} \Big]$$

$$\underline{\tau} = \inf_{\lambda_1, \lambda_0} \hat{\tau} = \hat{\tau}^* - \frac{\lambda}{N} \sum_{l \in \Pi} \Big[\sigma_1(l) N_{l,0} + \sigma_0(l) N_{l,1} \Big]$$

Then, the sensitivity interval can be computed as:

$$SI = \Big[\, \underline{\tau}, \; \overline{\tau} \, \Big]$$

To construct the tree that minimizes the width of the sensitivity interval we need to choose appropriate splitting and pruning criteria. This is simple in this case: since the bias is entirely described by the variance in the outcome, we can just use the variance reduction on the outcome variable as described in Section 2.1.

Algorithm 2 shows a summary of how to compute the sensitivity interval using the outcome variance method. It is worth noting that this method focuses solely on the outcome variable and does not consider the relationship between the covariates $\mathbf{X}$ and the treatment T. This approach also requires the same assumptions of smoothness and large-enough sample size as the Additive Bias Model from Section 2.5.1.

Algorithm 2: Outcome Variance Method

Inputs :

- D_{train} - Training data
- D_{estim} - Estimation data
- λ - Sensitivity parameter

Outputs : Upper and lower bound of SI for the ATE

tree $\leftarrow$ `OutcomeVarianceTree().fit(`D_{train}`)`
$L \leftarrow$ `tree.partition(`D_{estim}`)`
$N \leftarrow$ `size(`D_{estim}`)`
$\hat{\tau}^* \leftarrow$ `sum(`$\hat{\tau}^*(l)N_l/N$ `for` $l \in L$`)`
bias $\leftarrow$ `sum(`$\sigma_1(l)N_{l,0}/N + \sigma_0(l)N_{l,1}/N$ `for` $l \in L$`)`
lower $\leftarrow \hat{\tau}^* - \lambda$ bias
upper $\leftarrow \hat{\tau}^* + \lambda$ bias
return lower, upper

3.4 Propensity Trees

Rather than focusing on the relationship between the covariates and the outcome variance like the outcome variance trees, propensity trees take inspiration from Inverse Propensity Weighting (Section 2.3) and the Marginal Sensitivity Model (Section 2.5.2) and focus on the relationship between the covariates and the treatment assignment.

First, two definitions:

Definition 1 (Nominal propensity)**.** The true propensity including only observed confounders, defined as $e(\mathbf{x}) = P(T = 1 \mid \mathbf{X} = \mathbf{x})$. Reweighting by this propensity will not yield a balanced pseudo-population due to the remaining confounding bias coming from the unobserved confounders. However, this propensity can be estimated from data since it only relies on observed quantities.

Definition 2 (Leaf propensity)**.** The estimate of the propensity from the leaf of the tree, defined as $e_l = P(T = 1 \mid \mathbf{X} \in l)$. Given estimation data partitioned by a tree, the leaf propensity for leaf l can be estimated as $e_l = N_{l,1}/N_l$, where $N_{l,1}$ is the number of samples in the treatment group in leaf l and N_l is the total number of samples in leaf l. Although less accurate than the nominal propensity $e(\mathbf{x})$ since it relies on leaves rather than the full covariate space, this propensity remains more interpretable because it can be estimated directly from the tree.

The nominal propensity $e(\mathbf{x})$ and the leaf propensity e_l can be visualized on a spectrum represented in Figure 3.3. The most fine-grained balancing score is the identity function on the far left, which follows from the fact that $\mathbf{x}$ is a valid adjustment set. The coarsest balancing score is the nominal propensity score $e(\mathbf{x})$. Everything to the left of the nominal propensity score is an *approximation* of the nominal propensity. These functions only approximately balance the data, and the coarser they are the worse the approximation. The leaf propensity is a piece-wise constant approx-

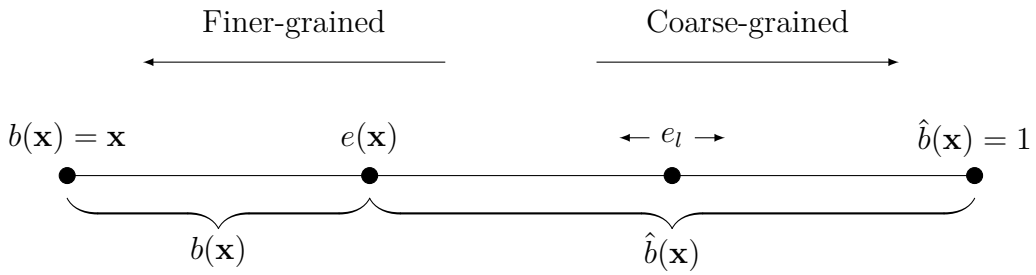


Figure 3.3: Spectrum of balancing scores in terms of how fine-grained they are.

imation of the nominal propensity and therefore lives in this space. A larger tree will produce a more detailed and expressive piece-wise approximation, and the leaf propensity will move to the left in the diagram. However, interpretability becomes a concern here: the tree cannot become too large. Therefore, the estimate e_l cannot just approximate $e(\mathbf{x})$ as closely as possible, but must take into account the size of the tree as well. The coarsest approximation of the nominal propensity is the constant function which uses no information from the covariates. In the case where ignorability holds, $e(\mathbf{x})$ is a constant function (since treatment assignment is not influenced by the covariates) and the gap between $e(\mathbf{x})$ and $\hat{b}(\mathbf{x})$ disappears.

As seen in Section 2.3, classical IPW re-weights the samples according to their nominal propensity to create a balanced pseudo-population from which treatment effects can be estimated:

$$\mathbb{E}[Y(t) \mid \mathbf{X}] = \mathbb{E}\left[\frac{P(T=t)}{e(\mathbf{x})} Y \mid T = t, \mathbf{X} = \mathbf{x}\right]$$

In our case, rather than a re-weighting of the whole population we are interested in a per-leaf re-weighting, effectively creating per-leaf pseudo-populations. Assuming consistency and that $\mathbf{X}$ is a valid adjustment set, per-leaf re-weighting can be done as follows:

$$\mathbb{E}[Y(t) \mid \mathbf{X} \in l] = \mathbb{E}\left[\frac{e_l}{e(\mathbf{x})} Y \mid T = t, \mathbf{X} \in l\right] \tag{3.2}$$

For a full derivation of the result above see Appendix A.3.

Taking inspiration from the Marginal Sensitivity Model, we can start looking at what would happen if $\mathbf{X}$ is *not* a valid adjustment set. We introduce a set of unknown confounders $\mathbf{U}$ such that $Y(t) \not\perp T \mid \mathbf{X}$ but $Y(t) \perp T \mid \mathbf{X}, \mathbf{U}$. Then, we define the following:

Definition 3 (Complete propensity)**.** The true propensity when including both the observed and unobserved confounders, defined as $\bar{e}(\mathbf{x}, \mathbf{u}) = P(T = 1 \mid \mathbf{X} = \mathbf{x}, \mathbf{U} = \mathbf{u})$. Reweighting the population by the inverse of the complete propensity would yield a balanced pseudo-population since all relevant confounders are taken into

account. However, this function cannot be estimated from data since we don't have access to the unobserved confounders $\mathbf{U}$ (by definition).

The complete propensity can be used to correct Equation 3.2 for unobserved confounders:

$$\mathbb{E}[Y(1) \mid \mathbf{X} \in l] = \mathbb{E}\left[\frac{e_l}{e(\mathbf{x})}\frac{e(\mathbf{x})}{\bar{e}(\mathbf{x},\mathbf{u})}Y \mid T = 1, \mathbf{X} \in l\right]$$

This decomposition presents two potential sources of error:

1. The coarsening of the leaf propensity relative to the nominal propensity implies that $\frac{e_l}{e(\mathbf{x})} \neq 1$. This might be due to the tree model being unable to fully model the propensity score due to sample size limitations or interpretability constraints. This effect can be corrected for since both $\hat{e}_l \approx e_l$ and $\hat{e}(\mathbf{x}) \approx e(\mathbf{x})$ can be estimated.
2. The bias due to unknown confounders implies that $\frac{e(\mathbf{x})}{\bar{e}(\mathbf{x},\mathbf{u})} \neq 1$. This effect cannot be corrected for since $\bar{e}(\mathbf{x},\mathbf{u})$ cannot be estimated from the available data. Instead, we can perform a sensitivity analysis by bounding $\frac{e(\mathbf{x})}{\bar{e}(\mathbf{x},\mathbf{u})}$ by some Γ and observing how changing the bounds affects the final sensitivity interval.

This leads to the following formulation:

$$\mathbb{E}[Y(1) \mid \mathbf{X} \in l] = \mathbb{E}\left[\frac{e_l}{e(\mathbf{x})}WY \mid T = 1, \mathbf{X} \in l\right], \qquad \frac{1}{\Gamma} \leq W = \frac{e(\mathbf{x})}{\bar{e}(\mathbf{x},\mathbf{u})} \leq \Gamma \quad (3.3)$$

The upper and lower bounds of $\mathbb{E}[Y(1) \mid \mathbf{X} \in l]$ for a particular Γ can be computed by taking the supremum and infimum over the weights:

$$\overline{\mathbb{E}[Y(1) \mid \mathbf{X} \in l]} = \sup_{\frac{1}{\Gamma} \leq W \leq \Gamma} \mathbb{E}\left[\frac{e_l}{e(\mathbf{x})}WY \mid T = 1, \mathbf{X} \in l\right]$$

$$\underline{\mathbb{E}[Y(1) \mid \mathbf{X} \in l]} = \inf_{\frac{1}{\Gamma} \leq W \leq \Gamma} \mathbb{E}\left[\frac{e_l}{e(\mathbf{x})}WY \mid T = 1, \mathbf{X} \in l\right]$$

These can be computed efficiently by splitting the outcomes Y in the leaf into two groups, one with positive outcomes and the other with the negative outcomes, taking advantage of linearity of expectations. For the upper bound, we maximize the positive values by setting $W = \Gamma$ and minimize the negative values with $W = \frac{1}{\Gamma}$. For the lower bound we do the opposite: we minimize the weights for positive values and maximize the weights for negative values.

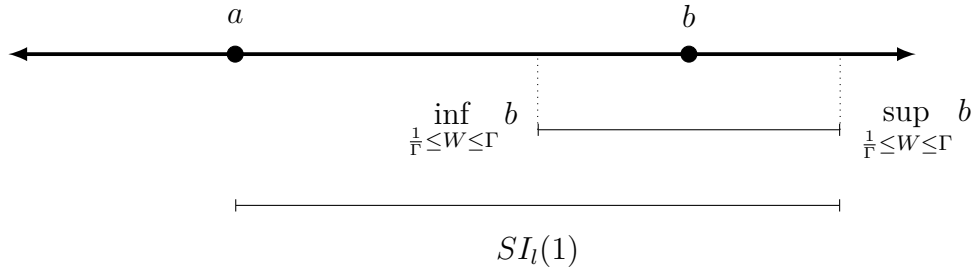


Figure 3.4: An illustration of the marginal sensitivity interval for estimating the average $Y(1)$ in leaf l. Point a represents the estimate of $Y(1)$ assuming ignorability, $\mathbb{E}[Y \mid T = 1, \mathbf{X} \in l]$. Point b represents the estimate when making the $e_l/e(\mathbf{x})$ correction and assuming that $\mathbf{X}$ is a valid adjustment set, $\mathbb{E}[\frac{e_l}{e(\mathbf{x})}WY \mid T = 1, \mathbf{X} \in l]$. The interval around b shows the range of possible values the estimate b could take if the weight $W = \frac{e(\mathbf{x})}{\bar{e}(\mathbf{x},\mathbf{u})}$ was bounded by some Γ. Notice that in this case point a lies *outside* of the interval around b, and therefore the interval is extended to the left to include point a.

$$\overline{\mathbb{E}[Y(1) \mid \mathbf{X} \in l]} = \Gamma\mathbb{E}\left[\frac{e_l}{e(\mathbf{x})}Y \mid T = 1, \mathbf{X} \in l, Y > 0\right] P(Y > 0) + \frac{1}{\Gamma}\mathbb{E}\left[\frac{e_l}{e(\mathbf{x})}Y \mid T = 1, \mathbf{X} \in l, Y < 0\right] P(Y < 0)$$

$$\underline{\mathbb{E}[Y(1) \mid \mathbf{X} \in l]} = \frac{1}{\Gamma}\mathbb{E}\left[\frac{e_l}{e(\mathbf{x})}Y \mid T = 1, \mathbf{X} \in l, Y > 0\right] P(Y > 0) + \Gamma\mathbb{E}\left[\frac{e_l}{e(\mathbf{x})}Y \mid T = 1, \mathbf{X} \in l, Y < 0\right] P(Y < 0)$$

The sensitivity interval for the estimate of the potential outcome $Y(1)$ can be computed as the interval between the upper and lower bounds:

$$SI_l(1) = \left[\ \overline{\mathbb{E}[Y(1) \mid \mathbf{X} \in l]}, \underline{\mathbb{E}[Y(1) \mid \mathbf{X} \in l]}\ \right]$$

However, there is a problem in this approach. As shown in Figure 3.4, there are two potential ways to estimate the average treatment effect with this model: point $a = \mathbb{E}[Y \mid T = 1, \mathbf{X} \in l]$ which assumes that the leaves are balanced and point $b = \mathbb{E}[\frac{e_l}{e(\mathbf{x})}Y \mid T = 1, \mathbf{X} \in l]$ which weights the outcome based on the propensity in the leaves and the estimated propensity. The former is simpler and more interpretable than the latter since it does not rely on the opaque estimate for the nominal propensity $\hat{e}(\mathbf{x})$, but it might not be included in the interval $[\ \overline{\mathbb{E}[Y(1) \mid \mathbf{X} \in l]}, \underline{\mathbb{E}[Y(1) \mid \mathbf{X} \in l]}\ ]$ if enough confounding bias is left in the leaf. For this reason, we chose to use the smallest interval which both contains the estimate assuming ignorability *and* the upper and lower bounds:

$$\overline{SI_l(1)} = \max\Big(\mathbb{E}[Y \mid T = 1, \mathbf{X} \in l],\ \overline{\mathbb{E}[Y(1) \mid \mathbf{X} \in l]} \Big)$$

$$\underline{SI_l(1)} = \min\Big(\mathbb{E}[Y \mid T = 1, \mathbf{X} \in l],\ \underline{\mathbb{E}[Y(1) \mid \mathbf{X} \in l]} \Big)$$

The upper and lower bounds for the expected value of the potential outcome where the treatment was *not* given in leaf l, $SI_l(0)$, can be computed similarly by noting that $P(T = 0 \mid X) = 1 - P(T = 1 \mid X)$. Then, the sensitivity for the average treatment effect in leaf l is computed as:

$$\underline{SI_l} = \underline{SI_l(1)} - \overline{SI_l(0)}$$

$$\overline{SI_l} = \overline{SI_l(1)} - \underline{SI_l(0)}$$

$$SI_l = \Big[\underline{SI_l}, \overline{SI_l} \Big]$$

Finally, given a tree partitioning Π, the sensitivity interval for the global average treatment effect is given by the weighted average of the sensitivity intervals in the leaves:

$$SI = \sum_{l \in \Pi} P(L = l) SI_i$$

As for the splitting and pruning criteria, similarly for the outcome variance method the goal is to optimize the tree in order to minimize the sensitivity interval. In turn, this involves both minimizing the upper and lower bounds for the potential outcomes, $\overline{\mathbb{E}[Y(t)]}$ and $\underline{\mathbb{E}[Y(t)]}$, as well as the distance between $\mathbb{E}[Y \mid T = t, \mathbf{X} \in l]$ and $\mathbb{E}[\frac{\hat{e}_l}{\hat{e}(\mathbf{x})} Y \mid T = t, \mathbf{X} \in l]$. To minimize these quantities directly, we opted to base the splitting and pruning criteria on the width of the sensitivity interval, as it reflects both quantities. Specifically, the best split was chosen as the split that most decreased the width of the sensitivity interval in the children nodes on average. The pruning criteria was set to the width of the sensitivity interval in each node.

3.5 Confidence Intervals for Trees

In this section we describe two methods for constructing confidence intervals for the tree methods described above.

3.5.1 Large-Sample Confidence Intervals

The general structure of the large-sample method for generating confidence intervals from a tree is to first find the confidence intervals for the average treatment effect in the leaves of the tree, and then aggregating those intervals into a single interval for the global average. From the Central Limit Theorem, the per-leaf average treatment effects are assumed to be normally distributed. One can then compute then the confidence interval for the global average treatment effect as follows:

1. In each leaf, estimate the standard error of the average treatment effect.
2. Aggregate the per-leaf standard error to find the global sample standard error.
3. Use the global standard error to find a confidence interval for the global ATE.

When constructing the confidence intervals in the leaf partitions, we assumed that the tree produces balanced partitions and therefore that the data is unconfounded. Let $\mu_{l,t}$ and $\sigma_{l,t}$ be the mean and standard deviation of the outcome Y in group $t \in \{0, 1\}$ in leaf l. Given balanced leaves, the average treatment effect in leaf l is given by $\tau(t) = \mu_{l,1} - \mu_{l,0}$. The standard error of the average treatment effect in leaf l, $SE(\tau(l))$, is defined as the standard deviation of the sampling distribution of $\tau(l)$. A common technique to compute the sample standard deviation of the difference of means if the Student's t test. However, we cannot assume that $\sigma l, 1 = \sigma_{l,0}$, since the distribution of the treatment and control groups might differ. Therefore, the Welch t test is used instead of the Student's t test, which defines $SE(\tau(l))$ as:

$$SE(\tau(l)) = \sqrt{\frac{\sigma_{l,1}^2}{N_{l,1}} + \frac{\sigma_{l,0}^2}{N_{l,0}}}$$

where $N_{L,1}$ is the number of samples in the treatment group in leaf l and $N_{l,0}$ is the number of samples in the control group in leaf l. When $\sigma_{l,1}$ and $\sigma_{l,0}$ are not known, these can be approximated by the sample standard deviations $s_{l,1}$ and $s_{l,0}$ computed from the treatment and the control group. The standard error can then be approximated by:

$$SE(\tau(l)) \approx \sqrt{\frac{s_{l,1}^2}{N_{l,1}} + \frac{s_{l,0}^2}{N_{l,0}}}$$

Given a tree partitioning Π, the point estimate for the global average treatment effect is estimated as the weighted average of the point estimates for the ATE in each leaf:

$$\tau = \mathbb{E}_\Pi[\tau(l)] \approx \sum_{l \in \Pi} \tau(l) \frac{N_l}{N}$$

where N_l is the number of samples in leaf l and N is the total number of samples. The standard error of the global average treatment effect can be derived as follows:

$$SE(\tau) = \sum_{l \in \Pi} SE\left(\frac{N_l}{N}\tau(l)\right) = \sum_{l \in \Pi} \frac{N_l^2}{N^2} SE\left(\tau(l)\right)$$

This derivation assumes that $\tau(l)$s are independent from each other. This is a plausible assumption since we assume that the units are i.i.d., and the growing of the tree which creates the leaf partitions is done on a separate data set (the training data). This is related to confidence intervals in stratified sampling, where in our case the stratifications are the partitions given by the leaves.

Finally, given a large enough sample size we can say that the distribution of τ is well-approximated by a normal distribution. Therefore, the $(1 - \alpha)$ confidence interval can be computed as:

$$CI(\tau) = \tau \pm z_{\alpha/2} SE(\tau)$$

where $z_{\alpha/2}$ is the critical value computed from the z-distribution (the standard normal distribution). The full algorithm for computing the large-sample confidence interval is summarized in Figure 3.

Algorithm 3: Large-Sample Confidence Interval

Inputs :

- `tree` - Trained tree
- D_{eval} - Evaluation data set
- CL - Confidence Level

Outputs : Lower and upper bounds for ATE

leaf_means $\leftarrow$ [ ]
leaf_vars $\leftarrow$ [ ]
$X, T, Y \leftarrow D_{eval}$
for $p \in$ `tree.partition(`X`)` **do**
    $Y_{p1}, Y_{p0} \leftarrow$ `SplitTreatmentGroups(`Y_p, T_p`)`
    $m_{p1}, s_{p1}, n_{p1} \leftarrow$ `Mean(`Y_{p1}`)`, `Std(`Y_{p1}`)`, `Len(`Y_{p1}`)`
    $m_{p0}, s_{p0}, n_{p0} \leftarrow$ `Mean(`Y_{p0}`)`, `Std(`Y_{p0}`)`, `Len(`Y_{p0}`)`
    $m_p \leftarrow m_{p1} - m_{p0}$
    $s_p \leftarrow$ `UnpooledVariance(`s_{p1}, s_{p0}, n_{p1}, n_{p0}`)`
    `leaf_means.append(`m_p`)`
    `leaf_vars.append(`s_p`)`
end
$m, s^2 \leftarrow$ `WeightedAverage(`*leaf_means, leaf_vars*`)`
$I \leftarrow$ `GaussianInterval(`m, s, CL`)`
lower $\leftarrow m - I$
upper $\leftarrow m + I$
return lower, upper

3.5.2 Bootstrap Confidence Intervals

Algorithm 4 shows the algorithm used to compute the bootstrap confidence intervals. The evaluation data set has the same shape as the training data set: $D_{eval} = (X_{eval}, T_{eval}, Y_{eval})$. In order to compute the mean, there must be at least one sample in both the treatment and the control group (i.e. $set(T_{eval}) = \{0, 1\}$. Sampling is done with replacement and per treatment group since they may have different distributions. The proportion of samples in each group is kept the same: if the original sample has 70 units in the treatment group and 30 units in the control group then the bootstrapped samples will also have 70 units in the treatment group and 30 units in the control group. Percentiles are computed by returning the value in μs at the index corresponding to the confidence level. For example, if we had 1000 bootstrap samples and a two-tailed confidence level $CL = 0.975$, then the lower bound corresponds to the value in μs at index 25 and the upper bound would be the value in μs at index 975.

Algorithm 4: Bootstrap Confidence Interval

Inputs :

- `tree`- Trained tree
- D_{eval} - Evaluation data set
- CL - Confidence Level
- B - Number of bootstrap samples

Outputs : Lower and upper bounds for ATE

Function `Sample`(D)**:**

    $D_1, D_0 \leftarrow$ `SplitTreatmentGroups`(D)
    $S_1 \leftarrow$ `SampleWithReplacement`(D_1)
    $S_0 \leftarrow$ `SampleWithReplacement`(D_0)
    return $S_1 + S_0$

$\mu s = [\,]$
for $b \in B$ **do**

    $S_b \leftarrow$ `Sample`(D_{eval})
    $\mu \leftarrow$ `tree.predict_ATE` (S_b)
    μs`.append` (μ)

end
$\mu s \leftarrow$ `Sort`(μs)
lower = `Percentile`(μs, $(1 - CL)/2$)
upper = `Percentile`(μs, $(1 + CL)/2$)
return lower, upper

4

Experiments

In order to evaluate the linear dependence tree, outcome variance tree and propensity tree we ran three experiments. Each experiment is executed on two data sets: a synthetic data set with a simple, well-defined data generating process and IHDP, a semi-synthetic data set which is closer to what one might encounter in the real world. The trees are compared to benchmark models which use similar assumptions where appropriate. We begin by providing a description of the data sets in Section 4.1. Next, we examine the relationship between tree size (our proxy value for interpretability) and model performance in Section 4.2. Section 4.3 looks at how the SI parameters affect the interval width and coverage probability for the different models. Finally, Section 4.4 compares the confidence intervals constructed by the various models.

4.1 Data Sets

In this section we describe the two data sets used in the experiments: a synthetic data set generated from a simple linear propensity model and IHDP, a semi-synthetic data set with more realistic properties that is commonly used for benchmarking in the causal inference community.

4.1.1 Synthetic Data Set

The aim of the synthetic data experiments is to explore the methods while having more control over the data. We can use a model that is simple to reason about and remove the effects of noise in the outcome. Using synthetic data also lets us have an exact equation for the propensity which we can use as an estimate in the Propensity Method for best-case scenario performance.

The data for this experiment is generated randomly from the simple model described in Figure 4.1. The model has a single continuous uniformly distributed covariate $X \in [0, 1]$. The propensity is a linear function of X, and it ensures positivity as the line never crosses 0 or 1. The outcome Y is generated as a linear function of X and T. The outcome is therefore confounded by covariate X since it affects both the treatment effect and the outcome. The true average treatment effect in this model is 4. The average treatment effect assuming ignorability is $5\frac{1}{15}$, meaning there is a confounding effect of $5\frac{1}{15} - 4 = 1\frac{1}{15}$.

$$X \sim Uniform(0, 1)$$
$$e(x) = 0.1 + 0.8x$$
$$T \sim Bernoulli(e(X))$$
$$Y = 4X - T$$

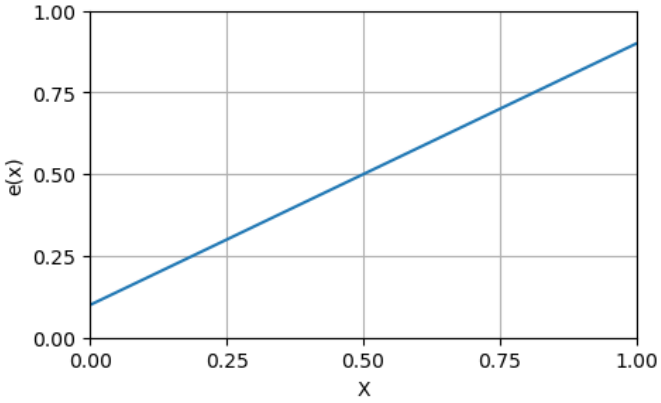


Figure 4.1: The linear propensity model

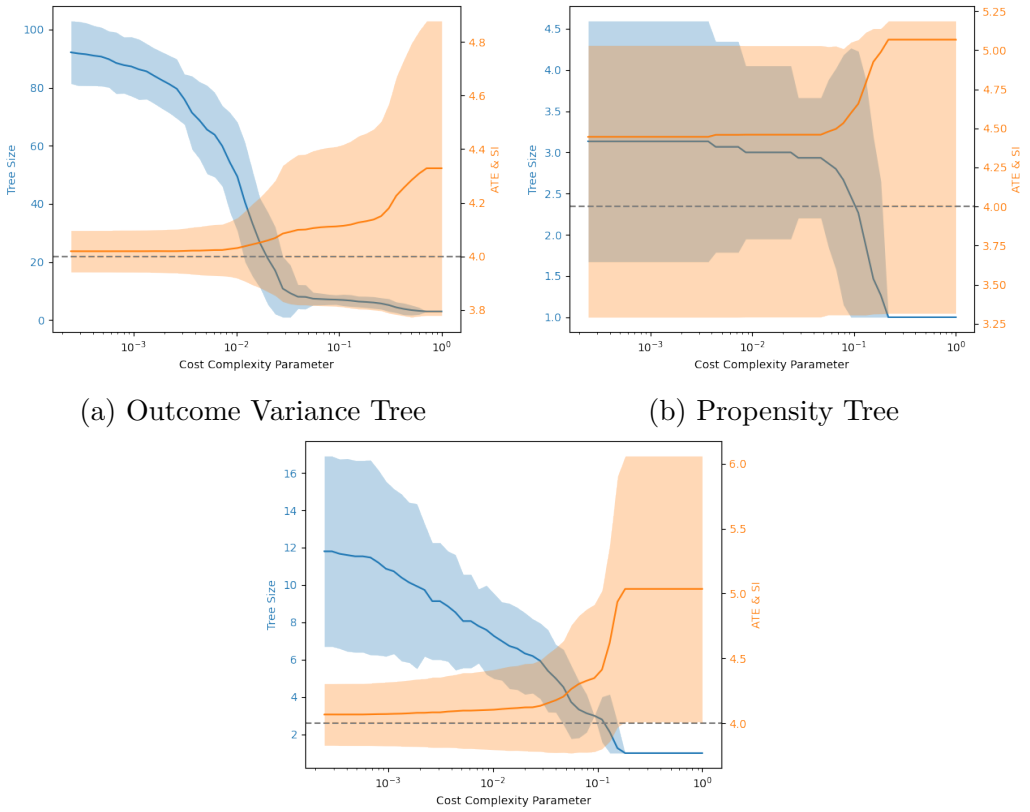
4.1.2 IHDP Data Set

The goal of the semi-synthetic experiment is to provide insights into the performance of the methods in a more realistic setting. For this purpose, we utilize the IHDP data set [15]. This data set comprises of 6 continuous and 19 binary covariates, with 139 units in the treatment group and 608 units in the control group. While the covariate and treatment assignment data originate from a real-world randomized experiment called the Infant Health Development Program, the outcome is generated based on these covariates (hence the semi-synthetic label). This ensures ignorability, as the outcome is only dependent on the measured covariates. To introduce confounding bias, a non-random portion of the treatment group is removed, simulating data from an observational study. The advantage of semi-synthetic data set over a real world data set is that we still know the true treatment effect while still being more realistic that the simple model from Section 4.1.1. The true average treatment effect in this model is 4, whilst the average treatment effect assuming ignorability is unknown.

4.2 Experiment: Cost-Complexity Parameter

This experiment focuses on the relationship between interpretability and model performance. The size of the tree is used as a proxy for interpretability: the smaller the tree, the more interpretable it is assumed to be. Note that this is a strong assumption: there may be many other factors that contribute to the interpretability of trees other than the tree size, as outlined in Section 2.6. Model performance is measured by the point-estimate for the ATE and the sensitivity interval. A performant model would have ATE estimates close to 4 (the true treatment effect set in the model) and small sensitivity intervals with good coverage. As the tree size decreases we expect the models to become less expressive and therefore less able to effectively split the data into balanced leaves. The ATE estimate would be expected to drift upwards towards the estimate for the ATE which assumes ignorability. Similarly, the sensitivity interval is expected to increase in width for the same sensitivity interval parameter.

The experiment involves varying the cost-complexity parameter, which controls the post-pruning of the trees. The higher the cost-complexity parameter, the more the pruning stage will give importance to the tree size and will tend to select smaller trees. First, the tree is trained on the data without post-pruning. Then, the same tree is copied and post-pruned for $N = 50$ different values of the cost-complexity parameter, ranging from close to 0 to 1. After pruning, the model is used to estimate the ATE, sensitivity interval and tree size. This entire process is repeated for $M = 30$ different trees to get an idea of the variance in the results. The sensitivity intervals for the outcome variance and linear dependence trees are computed with $\lambda = 1$. $\Gamma = 1.1$ is used for training and estimating the sensitivity interval for the propensity tree.



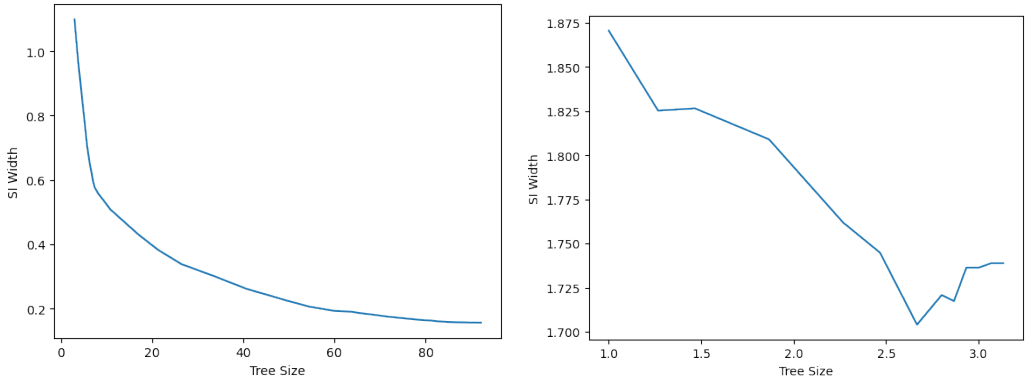
(a) Outcome Variance Tree

(b) Propensity Tree

(c) Linear Dependence Tree

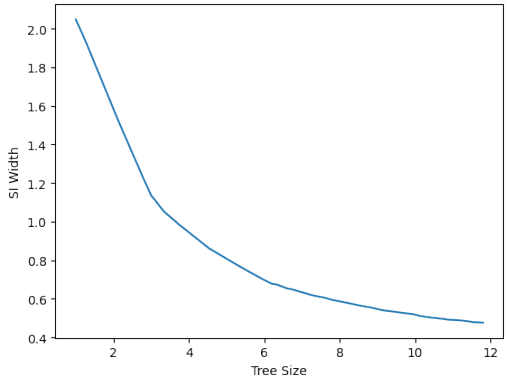
Figure 4.2: Relationship between the cost-complexity parameter, the tree size and the sensitivity interval for the synthetic data

Figure 4.2 shows the results for the synthetic data set. As predicted, in all three cases the tree size (represented in blue) decreases as the cost-complexity parameter increases: the more importance is given to the size in the pruning, the more the pruning is likely to select smaller trees. All trees eventually reach size 1, where the selection pressure for smaller trees outmatches the performance gains from more nodes due to larger cost-complexity parameters. However, for small values the trees do not start at the same size: the outcome variance tree starts with around 90 nodes on average, the linear dependence tree with around 12 and the propensity tree with just under 4. The outcome variance tree is therefore much more expressive at lower



(a) Outcome Variance Tree

(b) Propensity Tree



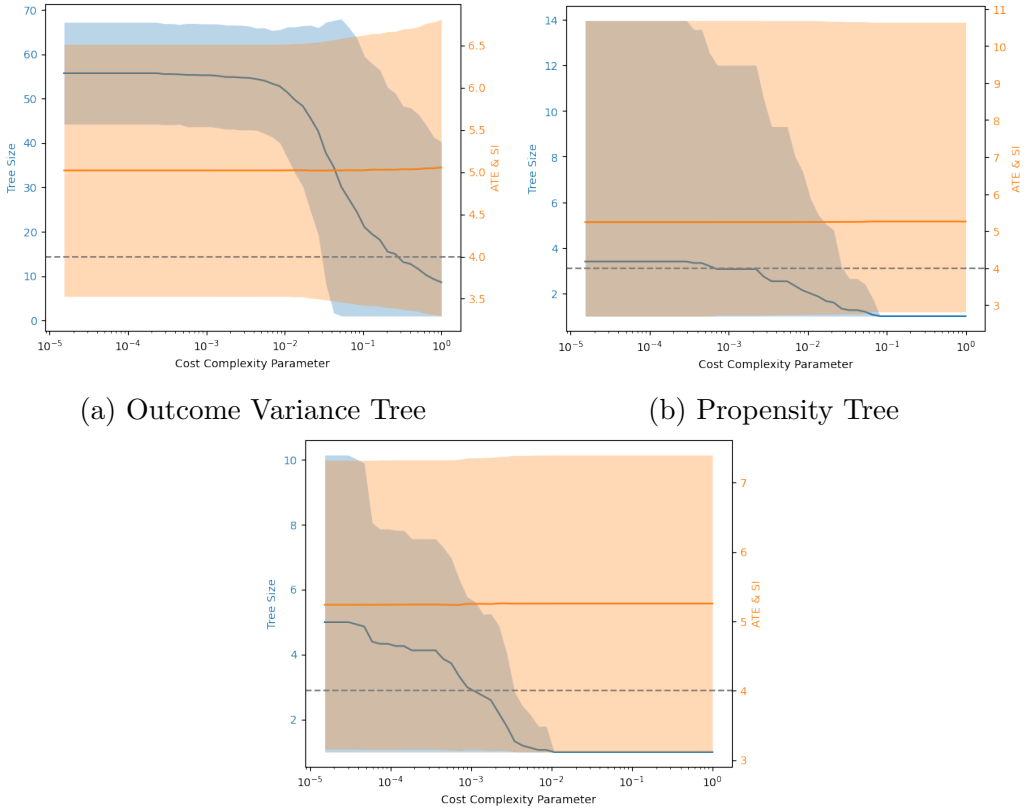
(c) Linear Dependence Tree

Figure 4.3: Relationship between the tree size and the sensitivity interval for the synthetic data

values of the cost-complexity parameter, which is reflected in it's better ATE point-estimates and smaller SI width. The propensity and linear dependence trees reach tree size 1 between 10^{-1} and 10^{0}, at which point the ATE point-estimate jumps to the estimate assuming no confounding bias ($5\frac{1}{15}$). This is because trees of size 1 assume that all the data belongs to the same leaf and therefore assume the data is already unconfounded.

In all three cases, the performance of the model in terms of the sensitivity interval and ATE estimate also worsens as the cost-complexity parameter increases. The ATE estimate moves further from the true value of 4 (the dashed line) and towards the estimate assuming ignorability. The interval width also steadily increases as the tree size decreases, leaving to a right-facing trumpet-like shape. This pattern of the interval width increasing as the tree size decreases is also demonstrated by the negative shape of the curves in Figure 4.3. The large jump in the propensity tree is also reflected here as a sharper angle in its relationship between tree size and interval width.

Figures 4.4 and 4.5 show the results for the IHDP data experiment. As with the synthetic data, the tree size decreases as the cost-complexity parameter increases for all three trees. However, likely due to the higher amounts of noise in the data,



(a) Outcome Variance Tree

(b) Propensity Tree

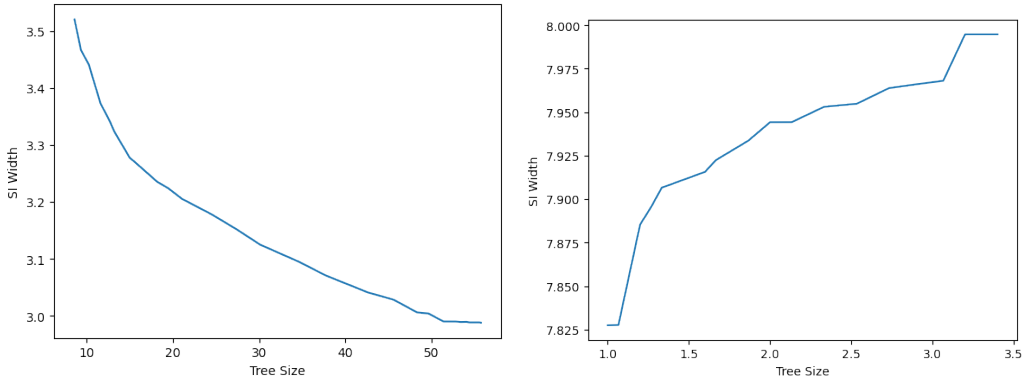
(c) Linear Dependence Tree

Figure 4.4: Relationship between the cost-complexity parameter, the tree size and the sensitivity interval for the IHDP data

both the tree size variance and the sensitivity intervals are significantly wider than before.

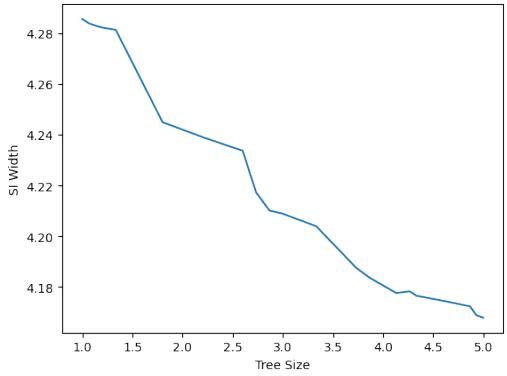
The outcome variance tree starts smaller than for the synthetic data at around 55 nodes, and also stays larger for much longer than before, with the drop-off occurring at $\sim 10^{-2}$ instead of $\sim 10^{-3}$. The outcome variance tree also still has the expected negative correlation between interval width and tree size.

The linear dependence tree also starts smaller at ~ 5 nodes in the tree on average. However, contrary to the outcome variance tree, the tree size drop off occurs sooner than its synthetic counterpart, with the tree reaching size 1 already at 10^{-2}. In fact, we had to extend the range of the cost-complexity parameter for the IHDP data to include values of the order of 10^{-5} in order to capture some of the drop-off to tree size 1. The negative correlation between tree size and interval width also holds for the linear dependence tree, although the interval width varies much less than before. These differences could be due to the fact that the values computed by the direct dependence criterion from Equation 3.1 are all very close to each other for the IHDP data, possibly due to the increased noise. In turn, this implies that improvements to the sensitivity interval due to growing the tree are only slight and easily affected by the cost-complexity parameter.



(a) Outcome Variance Tree

(b) Propensity Tree



(c) Linear Dependence Tree

Figure 4.5: Relationship between the tree size and the sensitivity interval for the IHDP data

Contrary to the other two trees, the propensity tree started out at a similar size as in the synthetic data experiment. Similarly to before, it still has a slow drop-off to tree size 1 from 10^{-3} to 10^{-1}. However, the variance for the tree size is much more pronounced in this experiment. The propensity tree also breaks the pattern in terms of the interval width. Whereas all the models trained on synthetic data and the other two trees trained on IHDP data all have a negative curve in the tree size vs. interval width graph, the propensity tree has a positive curve in Figure 4.5. This is strange, as it implies that increasing the size of the tree *decreases* the performance of the sensitivity interval. Our hypothesis is that this behaviour comes from the $\frac{e_l}{e(\mathbf{x})}$ ratio in Equation 3.3. Similarly to IPW, dividing by the propensity $e(\mathbf{x})$ exacerbates the impact of outliers with propensities close to 0. At the same time, subdividing the population has a chance of increasing some of the leaf propensities e_l since there are fewer samples in each leaf. These two effects sometimes interact multiplicatively with each other, leading to very large sensitivity intervals for some of the leaves, which negatively affect the final sensitivity interval. This effect becomes more pronounced when Γ is set to 1 for both training and estimation (as show in Figure 4.6) which effectively reduces Equation 3.3 to:

$$\mathbb{E}[Y(1) \mid \mathbf{X} \in l] = \mathbb{E}\left[\frac{e_l}{e(\mathbf{x})} Y \mid T = 1, \mathbf{X} \in l\right] \tag{4.1}$$

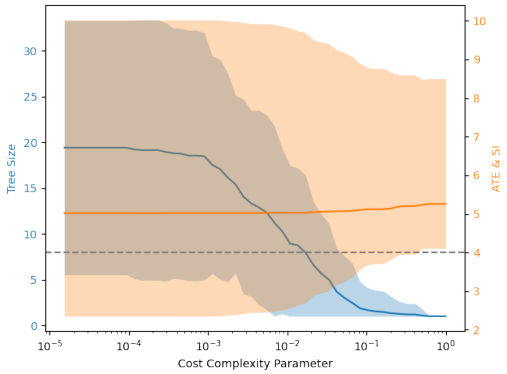


Figure 4.6: Relationship between the cost-complexity parameter, the tree size and the sensitivity interval for the propensity tree trained on IHDP data

4.3 Experiment: Sensitivity Parameter

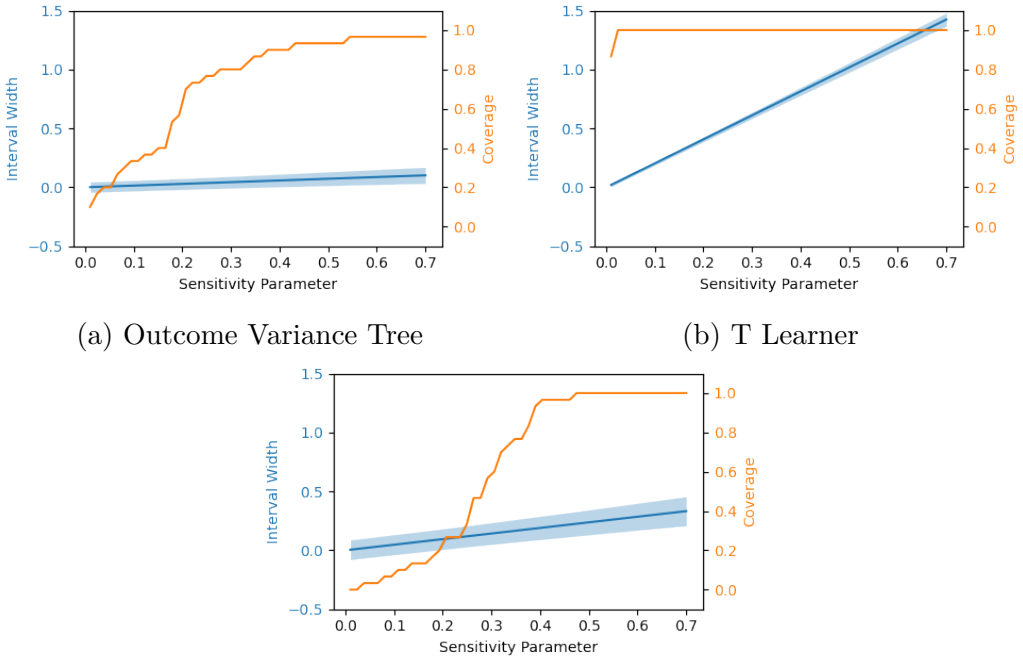
The goal of the sensitivity parameter experiment is to evaluate the quality of the sensitivity intervals computed by the tree models. A good sensitivity interval is the smallest interval that has the wanted coverage properties. For example, for a confidence level of 95%, the ideal interval is the smallest interval which contains the true value 95% of the time. To this end, intervals are often evaluated on two metrics: the interval width and the coverage. The interval width is simply the difference between the upper bound and the lower bound. The coverage is the probability that the interval will contain the true value of the measurement.

In our context, the sensitivity interval directly depends on a free parameter we (creatively) call the sensitivity parameter. In the Additive Bias Model (Section 2.5.1) the sensitivity parameter is λ, the multiplier in front of the bias computed from the outcome variance. In the Marginal Sensitivity Model (Section 2.5.2) the sensitivity parameter is Γ, which defines the bounds for the weight associated with the unknown confounders.

The central idea of the experiment is to vary the sensitivity parameter and compare the width and coverage of the sensitivity intervals produced by the models for the same parameter. However, since the function of the parameter varies significantly for the different sensitivity models, they are difficult to compare directly. Instead, we limited ourselves to comparisons between models which use the same sensitivity analysis method. In the case of the Linear Dependence Tree, which is not associated with a particular sensitivity analysis method, the sensitivity interval was computed for both sensitivity analysis methods. In the experiment we also compared the tree models to two benchmark models: a T Learner with regression trees for the additive bias model class and a classical IPW with a regression tree estimate of the propensity score for the marginal sensitivity model class. The benchmark models

were chosen to be simple, not constrained to be interpretable and to rely on similar base assumptions as the tree models.

The experiment was conducted as follows. For each model type, $M = 30$ models were trained and evaluated on different realizations of the data sets. For the synthetic data, new data was generated to train each model (500 samples). For the IHDP data, each model was trained on a different realization of IHDP with a 50/50 split between training and evaluation sets, maintaining the same proportions of treated and control units. For each trained model, the sensitivity interval was computed for $N = 50$ sensitivity parameters. The λs were varied between 0.01 and 0.7, whereas the Γs were varied between 1 and 1.3. The training Γ for the propensity tree was set to the same value as the estimation Γ. The trees were given a small cost-complexity parameter of 0.00001 in order to produce larger and more expressive trees. The only exception was the IPW regression tree, which was given a cost-complexity parameter of 0.01 in order to avoid extreme propensity predictions (predictions of 0 or 1). Finally, the interval width and coverage metrics were computed from the sensitivity intervals for each sensitivity parameter. Coverage was computed as the percentage of sensitivity intervals which contained the true treatment effect (4 in each case). Interval width was computed as the unweighted average interval width.



(a) Outcome Variance Tree

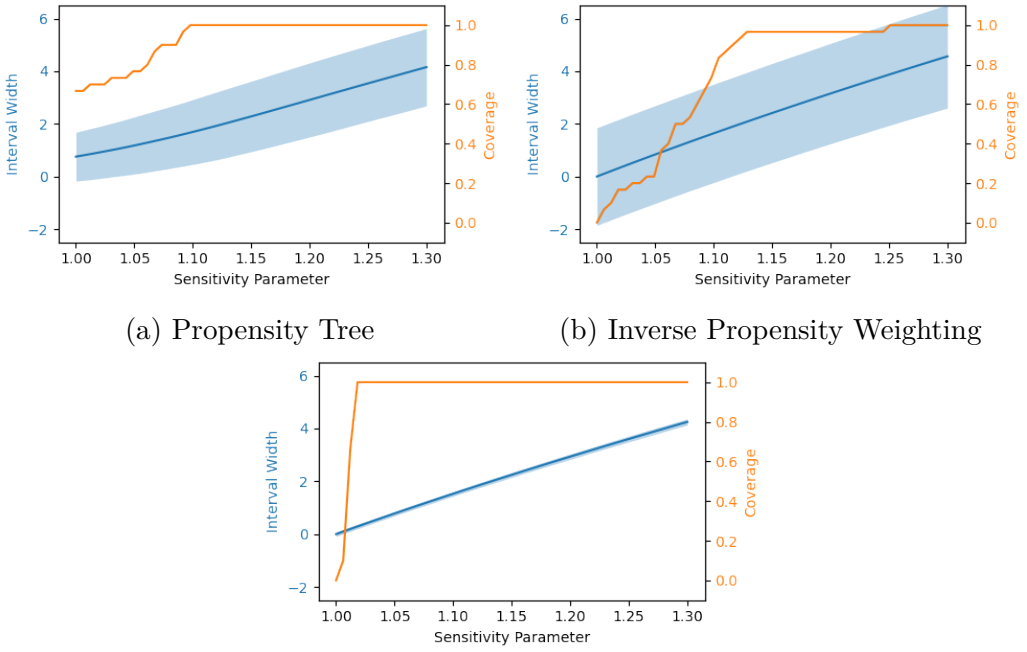
(b) T Learner

(c) Linear Dependence Tree

Figure 4.7: Relationship between the SI parameter and the sensitivity interval width and coverage probability for the synthetic data for models using the Additive Bias Model

Figure 4.7 shows the results for the additive bias sensitivity models run on the synthetic data set. The interval width for all three models scales linearly with the sensitivity parameter, which follows directly from Equation 2.7. The outcome variance tree has a smaller gradient and variance for the interval width, which indicates

slightly better performance. This is likely due to the outcome variance tree optimizing directly for minimal sensitivity intervals constructed by the additive bias method. The two tree models were comparable in terms of coverage. The T Learner however has wider average interval width and higher coverage than the other two models. One reason for this could be that T Learners train two models, one on the data from the control group and the other on the data from the test group, and then uses the output of the models to estimate the treatment effect. Splitting the data in this manner gives less data to each individual model, which tends to introduce more variance in the output and increases the size of the sensitivity interval.



(a) Propensity Tree

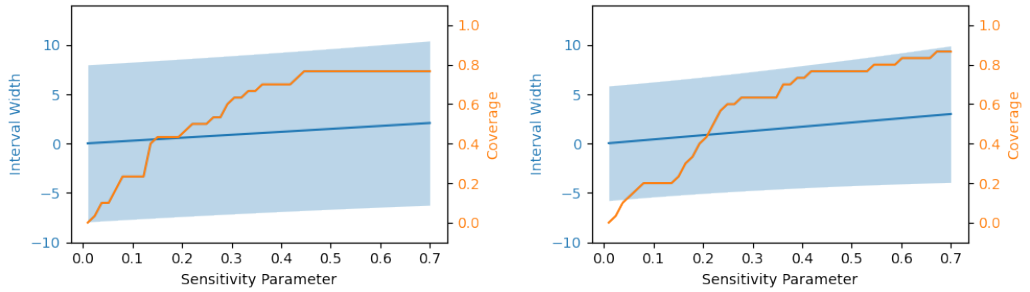
(b) Inverse Propensity Weighting

(c) Linear Dependence Tree

Figure 4.8: Relationship between the SI parameter and the sensitivity interval width and coverage probability for the synthetic data for models using the Marginal Sensitivity Model

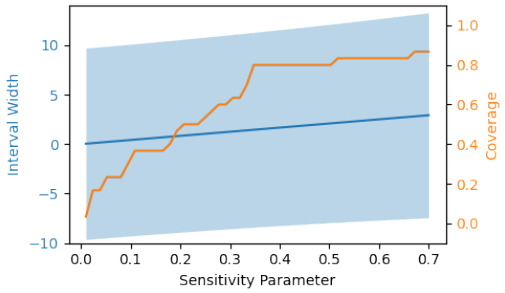
The results for the models using the marginal sensitivity analysis are displayed in Figure 4.8. Here, as in Figure 4.7, the sensitivity interval width seems to grow linearly with the sensitivity parameter Γ. In the marginal sensitivity analysis, Γ defines bounds for the weights representing the bias from the unobserved confounders. The upper bound is defined as Γ and the lower bound as $\frac{1}{\Gamma}$. The upper bound is linear with the sensitivity parameter and the lower bound is approximately linear (since for the restricted domain of $[1, 1.3]$ $y = \frac{1}{x}$ is approximately linear, which explains the apparent linearity of the interval width. In terms of variance of the interval width, the linear dependence tree has a remarkably lower variance than the other two models. The coverage is also markedly better for the linear dependence tree than for the other two models, converging to 1 much faster than the rest. The propensity tree and inverse propensity weighting behave similarly, with the coverage plateauing at 1. However, the propensity tree is the only model that has an above-zero coverage and interval width for $\Gamma = 1$. This is due to the fact that the sensitivity interval

for the propensity tree also takes into account the unbiased estimate for the average treatment effect, which widens the interval and increases coverage for small values of Γ.



(a) Outcome Variance Tree

(b) T Learner

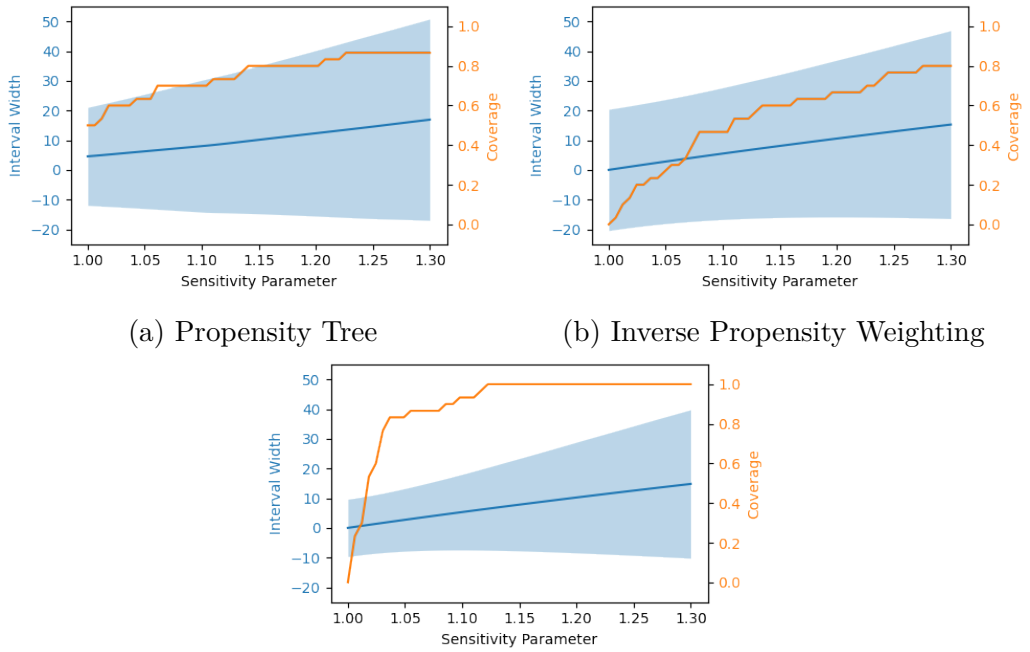


(c) Linear Dependence Tree

Figure 4.9: Relationship between the SI parameter and the sensitivity interval width and coverage probability for the IHDP data for models using the Additive Bias Model

Figure 4.9 shows the results of the experiment for the additive bias sensitivity models on the IHDP data. All three models performed similarly. The interval widths are linearly related to the sensitivity parameter as before, but their variance has significantly increased. The coverage is also poorer overall, only reaching ~ 0.85 at $\lambda = 0.7$. These effects could be explained by the fact that the IHDP data is much noisier than the synthetic data, which interferes with the algorithms' ability to identify and remove confounding bias.

Finally, the results of the experiment for the marginal sensitivity analysis models are shown in Figure 4.10. Similarly to the synthetic data, the interval widths seem to grow approximately linearly with the sensitivity parameter, although the non-linearity becomes more apparent in the interval width variance. The interval width variance and coverage are also worse than before, likely due to the noisiness of the IHDP data relative to the synthetic data. The interval width variance however also seems to grow considerably with the sensitivity parameter which was not the case before. Another similarity with the models trained on synthetic data is that the linear dependence tree is the clear best performer, with much better coverage and smaller interval width than the other two.



(a) Propensity Tree

(b) Inverse Propensity Weighting

(c) Linear Dependence Tree

Figure 4.10: Relationship between the SI parameter and the sensitivity interval width and coverage probability for the IHDP data for models using the Marginal Sensitivity Model

4.4 Experiment: Confidence Intervals

The goal of this experiment is to compare the confidence intervals generated by the three methods over 500 realizations of each data set. Confidence intervals give a different perspective from sensitivity intervals, as they concern themselves with all sources of noise rather than just confounding bias. We chose to evaluate the methods over two types of confidence intervals: large-sample (Section 3.5.1) and bootstrap (Section 3.5.2). We also compared the methods to the two benchmarks from the sensitivity parameter experiment from Section 4.3, as well as the estimation of the ATE without any correction for confounding bias (which we call the "Basic" algorithm). In training, we used a cost-complexity parameter of 0.01 for the tree models. A training gamma of 1.2 was used for the propensity tree. Similarly to Experiment 4.3, the quality of the confidence intervals was evaluated by recording the coverage and the average interval width. We used a 95% confidence level when evaluating both confidence intervals, and a bootstrap sample size of 1000 for the bootstrap confidence intervals.

The results of the experiments can be found in Table 4.1 (synthetic data) and Table 4.2 (IHDP data). Overall, the bootstrap confidence intervals had a tendency to be much more confident than the large-sample confidence intervals with much smaller average interval widths across the board. However, coverage tended to go in favor of large-sample confidence intervals, especially for the IHDP data.

	Large-Sample		Bootstrap	
Algorithm	Width ↓	Coverage ↑	Width ↓	Coverage ↑
Basic	0.285	0.000	0.284	0.000
IPW	0.668	0.998	0.209	0.932
T Learner	0.313	1.000	0.075	0.190
Outcome Variance Tree	0.048	0.424	0.100	0.674
Propensity Tree	0.406	0.000	0.400	0.000
Linear Dependence Tree	0.109	0.054	0.158	0.106

Table 4.1: The large-sample and bootstrap confidence intervals for models trained on the synthetic data set

	Large-Sample		Bootstrap	
Algorithm	Width ↓	Coverage ↑	Width ↓	Coverage ↑
Basic	2.106	0.616	1.105	0.416
IPW	2.106	0.616	1.164	0.452
T Learner	2.047	0.694	1.122	0.540
Outcome Variance Tree	15.759	0.884	1.496	0.600
Propensity Tree	21.748	0.936	1.562	0.522
Linear Dependence Tree	21.749	0.936	1.554	0.528

Table 4.2: The large-sample and bootstrap confidence intervals for models trained on the IHDP data set

In the synthetic data set, the coverage for the basic model was zero for both confidence intervals, meaning that the confidence intervals never contained the true value. This is because there is a distance of $1\frac{1}{15}$ between the true treatment effect and the estimated treatment effect assuming ignorability, but the average interval width is much smaller. Since there is relatively little noise in the data and the model is unaware of the confounding bias, the confidence intervals are overconfident. When trained on the noisier IHDP data, the basic model is more conservative and produces larger confidence intervals and with better coverage.

In contrast, the Inverse Propensity Weighting algorithm performed significantly better for the synthetic data than for the IHDP data. Here, the confounding bias is adjusted using an estimate of the propensity score, and therefore becomes much less of a factor. The noise in the IHDP data set likely decreased the performance of the regression tree estimate of the propensity score, which in turn worsened the performance of the model. A similar trend can also be observed for the T Learner: its large-sample confidence interval performs very well for the synthetic data set, whereas the two confidence intervals for the IHDP data set perform less well. However, the bootstrap confidence interval in the synthetic data set is the worst performer with very small average interval width and a coverage of 0.19.

The tree models had quite different performance on the synthetic data. The outcome variance tree produced confidence intervals with small average interval width

and lower coverage than the IPW model. The linear dependence tree also had very low interval widths, but with very low coverage compared to the outcome variance tree. The propensity tree obtained a coverage of 0 for both confidence intervals. One reason for this behaviour could be that the outcome variance tree is the most effective of the three at reducing confounding bias in the data, whilst the propensity tree struggled the most (just like the basic model). However, the tree models had very similar performance for the IHDP data. The large-sample confidence intervals had large average interval widths with high coverage, with the outcome variance tree being a little smaller and with lower coverage than the other two. This could be due to the fact that the large-sample intervals are explicitly based on the outcome variance which the outcome variance tree optimizes for directly. Note that the interval widths for the trees increased much more than for the benchmarks, indicating that the trees might not be as confident in the presence of noisier data. The bootstrap confidence intervals for the large-sample confidence intervals are all very comparable.

5

Conclusion

This work presents three methods for estimating causal treatment effects from observational data using interpretable decision trees. The three methods share a common structure. First, a decision tree is learned from training data with a greedy split-then-prune algorithm. The goal of the tree is to split the covariate space into balanced partitions, where units from the treatment and control groups in the same partition share the same distribution. Then, the per-leaf populations are used to estimate the average treatment effect as well as confidence and sensitivity intervals. Finally, these measures are aggregated into global estimates by a weighted average over the per-leaf populations, where the weights reflect the population size.

Given this structure, each method proposed a different approach to the splitting and pruning criteria used to train the tree, as well as the sensitivity models used to compute the sensitivity intervals. The outcome variance model uses the relationship between the covariates and the outcome and splits the tree by minimizing the outcome variance. It is associated with the additive bias sensitivity model. The propensity method instead uses the relationship between the covariates and the treatment assignment, focusing on approximating a black-box estimate of the propensity. Propensity trees are associated with the marginal sensitivity model. In both the outcome variance and propensity trees, the splitting and pruning criteria are explicitly optimized in order to minimize their associated sensitivity model. The linear dependence method however explicitly measures the dependence between the covariates and the treatment assignment and splits the tree in order to minimize the measured dependence in the leaves.

The three methods were tested on a synthetic and a semi-synthetic data set. The results of the methods are discussed in Chapter 4 alongside three benchmarks: the average treatment effect estimate assuming ignorability, Inverse Propensity Weighting with a regression tree estimate of the propensity and a T Learner with a regression tree estimate of the outcome.

A common thread observed throughout the experiments was that the amount of noise not related to confounding interferes with the models' ability to remove confounding bias. Sensitivity intervals and model size for models trained on noisier IHDP data were always larger and more conservative than for the synthetic data. The tree models also struggled to reduce the sensitivity interval as they grew larger in noisier environments, especially the linear dependence tree. However, the lack of noise in synthetic data lead many of the models to produce overconfident confidence

intervals with low interval width and low coverage, whereas the noise in the IHDP data forced the models to produce more conservative intervals with higher coverage. This was particularly apparent for the basic model and the propensity tree, which produced intervals with a coverage of 0 for the synthetic data. Also, whereas the tree methods outperformed the benchmarks in terms of sensitivity intervals, the benchmarks outperformed in terms of the confidence intervals, especially for the synthetic data.

In terms of sensitivity intervals, the outcome variance tree was successfully able to optimize the intervals constructed by the additive bias method, outperforming the other models. It also seemed to produce the largest and most reliable trees in the cost-complexity experiment. The linear dependence tree also performed well, achieving comparable results with the outcome variance tree and significantly outperforming the other models for the marginal sensitivity intervals. However, the propensity tree did not manage to outperform the other models even though its splitting and pruning criteria explicitly optimize for the sensitivity interval width computed via the marginal sensitivity method. Instead, it performed comparably to the IPW benchmark in the sensitivity parameter experiment and exhibited a surprising behaviour in the cost-complexity parameter experiment where increasing the size of the tree actually increased the size of the sensitivity interval. Our hypothesis is that this occurs due to the imperfections in the estimates for e_l and $e(\mathbf{x})$. However, more work needs to be done to fully understand the problem.

Finally, the cost-complexity experiment showed that, for the outcome variance and linear dependence trees, decreasing the cost-complexity parameter increases tree size and also increases the performance of the sensitivity interval and average treatment effect prediction. Therefore, the cost-complexity parameter can be thought of as a knob that allows a potential user to decide how performant vs. interpretable they want the model to be. Sweeping the parameter also reveals interesting behaviour. By observing how the sensitivity interval width changes with tree size, one can get an idea about how much confounding bias there is relative to other noise in the data. For relatively noiseless data, like the synthetic data set, the interval widths shrunk significantly as the trees grew larger. For noisy data, such as the IHDP data set, the sensitivity intervals stayed relatively constant.

5.1 Improvements and Future Work

There are many potential avenues for improvements to the methods. In this work, we used relatively simple methods for constructing confidence intervals, and a large literature exists for improving both large-sample methods and bootstrap methods. For the large-sample methods one could investigate as to why the intervals were so much larger for the trees than for the benchmarks in the IHDP data set. For the bootstrap methods, one could consider implementing bias corrections and acceleration [37], which should help alleviate bias and skewness between the bootstrap distribution and the sample distribution.

Another methodological improvement for the propensity method in particular is

to improve the propensity estimate $\hat{e}(\mathbf{x})$. In our work we used a Logistic Regression model, which although popular and powerful assumes a linear relationship between the covariates and the logit of the treatment assignment. Several alternatives have been proposed in the literature such as neural networks, random forests or meta-learners [38]. These alternatives could help the performance of our propensity method, particularly in cases where $e(\mathbf{x})$ is more complex or when the propensity approaches 0 or 1 where small errors in the estimation are greatly magnified in the weights.

Improvements could also be made in the evaluation of the methods. For the synthetic data experiments, with would be interesting to characterize how the size of the tree scales with the complexity of the relations between the covariates and the treatment assignment and outcome. Intuitively, more complex confounding relationships necessitate that the trees be more expressive in order to capture enough detail to balance the leaves, which in turn implies bigger and potentially less interpretable models. To answer the question, one could imagine creating synthetic data generating processes with increasing confounder complexity, for example by increasing the order of the polynomial which describes the propensity or by increasing the dimensionality of the covariates in different ways. Then, one could compare the performance of the models for the different synthetic data sets and try to identify scaling relations between confounder complexity and model performance.

Another avenue for improvement for the synthetic data experiments is to look at how noise in the outcome affects model performance. Outcome noise is present in most real-world scenarios. How does increasing the amount noise affect the performance of the methods? How will it affect the scaling relationships between the cost complexity parameter and the model performance and the sensitivity parameter and model performance? How do these relationships change for different kinds of noise (Gaussian, Uniform, etc.)?

As for the semi-synthetic data experiment, it would be interesting to apply the methods to other benchmark data sets with different properties. For example, how would the outcome variance and propensity methods perform on the Jobs data set [39] where the outcome is binary rather than continuous? Several other benchmark data sets are listed in [5]. One could also look at the sample efficiency of the methods. How well do they perform for smaller data sets? This could be tested on the IHDP data set by randomly removing units to make the data set smaller.

Improvements could also be made to our evaluation of interpretability. We used a functionally-grounded evaluation, where we used the tree size as a proxy for the interpretability of the whole method and the tree depth as a proxy for the interpretability of making predictions using the models. It would be interesting however to evaluate the models using an application-grounded evaluation. For example, given a particular problem for which the assumptions for all three models are applicable, once could form a blind experiment where test subjects are asked to use and rate the models produced by the three methods. Then, the interpretability of the model would be given by the average score of the test subjects for the model.

There are several potential avenues for extending the models. A straight-forward

extension would be for data with more than two groups. This would necessitate a generalization of the splitting criteria and for how the propensity function is used. Finally, another avenue that was hinted at in Section 3.2 is to explore another method based on direct dependence as measured by the Maximum Mean Difference.

Bibliography

[1] A. Yazdani and E. Boerwinkle, "Causal inference in the age of decision medicine," *Journal of data mining in genomics & proteomics*, vol. 6, no. 1, 2015.

[2] T. A. Glass, S. N. Goodman, M. A. Hernán, and J. M. Samet, "Causal inference in public health," *Annual review of public health*, vol. 34, pp. 61–75, 2013.

[3] H. R. Varian, "Causal inference in economics and marketing," *Proceedings of the National Academy of Sciences*, vol. 113, no. 27, pp. 7310–7315, 2016.

[4] D. Liang, L. Charlin, and D. M. Blei, "Causal inference for recommendation," in *Causation: Foundation to Application, Workshop at UAI. AUAI*, 2016.

[5] L. Yao, Z. Chu, S. Li, Y. Li, J. Gao, and A. Zhang, "A survey on causal inference," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 5, pp. 1–46, 2021.

[6] S. Vansteelandt and R. M. Daniel, "On regression adjustment for the propensity score," *Statistics in medicine*, vol. 33, no. 23, pp. 4053–4072, 2014.

[7] P. R. Rosenbaum and D. B. Rubin, "The central role of the propensity score in observational studies for causal effects," *Biometrika*, vol. 70, no. 1, pp. 41–55, 1983.

[8] E. A. Stuart, "Matching methods for causal inference: A review and a look forward," *Statistical science: a review journal of the Institute of Mathematical Statistics*, vol. 25, no. 1, p. 1, 2010.

[9] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[10] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[11] A. A. Freitas, "Comprehensible classification models: A position paper," *ACM SIGKDD explorations newsletter*, vol. 15, no. 1, pp. 1–10, 2014.

[12] S. Athey and G. Imbens, "Recursive partitioning for heterogeneous causal effects," *Proceedings of the National Academy of Sciences*, vol. 113, no. 27, pp. 7353–7360, 2016.

[13] S. Wager and S. Athey, "Estimation and inference of heterogeneous treatment effects using random forests," *Journal of the American Statistical Association*, vol. 113, no. 523, pp. 1228–1242, 2018.

[14] S. Athey and S. Wager, "Estimating treatment effects with causal forests: An application," *Observational Studies*, vol. 5, no. 2, pp. 37–51, 2019.

[15] J. L. Hill, "Bayesian nonparametric modeling for causal inference," *Journal of Computational and Graphical Statistics*, vol. 20, no. 1, pp. 217–240, 2011.
[16] H. A. Chipman, E. I. George, and R. E. McCulloch, "Bart: Bayesian additive regression trees," *The Annals of Applied Statistics*, 2010.
[17] P. Wang, W. Sun, D. Yin, J. Yang, and Y. Chang, "Robust tree-based causal inference for complex ad effectiveness analysis," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 2015, pp. 67–76.
[18] J. Li, S. Ma, T. Le, L. Liu, and J. Liu, "Causal decision trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 2, 257–2y71, 2016.
[19] H. Laurent and R. L. Rivest, "Constructing optimal binary decision trees is np-complete," *Information processing letters*, vol. 5, no. 1, pp. 15–17, 1976.
[20] L. Breiman, *Classification and regression trees*. Routledge, 2017.
[21] Y. Mansour, "Pessimistic decision tree pruning based on tree size," in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, Citeseer, 1997, pp. 195–201.
[22] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers-a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 4, pp. 476–487, 2005.
[23] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.
[24] P. W. Holland, "Statistics and causal inference," *Journal of the American statistical Association*, vol. 81, no. 396, pp. 945–960, 1986.
[25] D. B. Rubin, "Causal inference using potential outcomes: Design, modeling, decisions," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 322–331, 2005.
[26] E. L. Hannan, "Randomized clinical trials and observational studies: Guidelines for assessing respective strengths and limitations," *JACC: Cardiovascular Interventions*, vol. 1, no. 3, pp. 211–217, 2008.
[27] M. A. Hernán and J. M. Robins, *Causal inference*. CRC Boca Raton, FL, 2010.
[28] A. DAmour, P. Ding, A. Feller, L. Lei, and J. Sekhon, "Overlap in observational studies with high-dimensional covariates," *Journal of Econometrics*, vol. 221, no. 2, pp. 644–654, 2021.
[29] B. Neal, "Introduction to causal inference," 2015.
[30] E. J. Williamson and A. Forbes, "Introduction to propensity scores," *Respirology*, vol. 19, no. 5, pp. 625–635, 2014.
[31] P. C. Austin and E. A. Stuart, "Moving towards best practice when using inverse probability of treatment weighting (iptw) using the propensity score to estimate causal treatment effects in observational studies," *Statistics in medicine*, vol. 34, no. 28, pp. 3661–3679, 2015.
[32] B. Efron, "Bootstrap methods: Another look at the jackknife," *The Annals of Statistics*, vol. 7, no. 1, pp. 1–26, 1979. DOI: 10.1214/aos/1176344552.
[33] J. Fox, *Applied regression analysis and generalized linear models*. Sage Publications, 2015.
[34] T. J. DiCiccio and B. Efron, "Bootstrap confidence intervals," *Statistical science*, vol. 11, no. 3, pp. 189–228, 1996.

[35] W. Liu, S. J. Kuramoto, and E. A. Stuart, "An introduction to sensitivity analysis for unobserved confounding in nonexperimental prevention research," *Prevention science*, vol. 14, pp. 570–580, 2013.

[36] J. Cornfield, W. Haenszel, E. C. Hammond, A. M. Lilienfeld, M. B. Shimkin, and E. L. Wynder, "Smoking and lung cancer: Recent evidence and a discussion of some questions," *Journal of the National Cancer institute*, vol. 22, no. 1, pp. 173–203, 1959.

[37] B. Efron, "Better bootstrap confidence intervals," *Journal of the American statistical Association*, vol. 82, no. 397, pp. 171–185, 1987.

[38] D. Westreich, J. Lessler, and M. J. Funk, "Propensity score estimation: Neural networks, support vector machines, decision trees (cart), and meta-classifiers as alternatives to logistic regression," *Journal of clinical epidemiology*, vol. 63, no. 8, pp. 826–833, 2010.

[39] J. Robins, "A new approach to causal inference in mortality studies with a sustained exposure periodapplication to control of the healthy worker survivor effect," *Mathematical modelling*, vol. 7, no. 9-12, pp. 1393–1512, 1986.

A

Appendix

A.1 Derivations for the Additive Bias Model

$$\mathbb{E}[Y(1) \mid X] \tag{A.1}$$

$$= \mathbb{E}[Y(1) \mid T = 1, X]P(T = 1 \mid X) + \mathbb{E}[Y(1) \mid T = 0, X]P(T = 0 \mid X) \tag{A.2}$$

$$= \mathbb{E}[Y(1) \mid T = 1, X]P(T = 1 \mid X) + (\mathbb{E}[Y(1) \mid T = 1, X] - \eta_1(X))P(T = 0 \mid X) \tag{A.3}$$

$$= \mathbb{E}[Y(1) \mid T = 1, X] - \eta_1(X)P(T = 0 \mid X) \tag{A.4}$$

$$= \mathbb{E}[Y \mid T = 1, X] - \eta_1(X)P(T = 0 \mid X) \tag{A.5}$$

$$\mathbb{E}[Y(0) \mid X] \tag{A.6}$$

$$= \mathbb{E}[Y(0) \mid T = 1, X]P(T = 1 \mid X) + \mathbb{E}[Y(0) \mid T = 0, X]P(T = 0 \mid X) \tag{A.7}$$

$$= (\mathbb{E}[Y(0) \mid T = 0, X] + \eta_0(X))P(T = 1 \mid X) + \mathbb{E}[Y(0) \mid T = 0, X]P(T = 0 \mid X) \tag{A.8}$$

$$= \mathbb{E}[Y(0) \mid T = 0, X] + \eta_0(X)P(T = 1 \mid X) \tag{A.9}$$

$$= \mathbb{E}[Y \mid T = 0, X] + \eta_0(X)P(T = 1 \mid X) \tag{A.10}$$

$$\begin{aligned} CATE(X) &= \mathbb{E}[Y(1) \mid X] - \mathbb{E}[Y(0) \mid X] \\ &= \mathbb{E}[Y(1) \mid T = 1, X] - \eta_1(X)P(T = 0 \mid X) \\ &\quad - \mathbb{E}[Y(0) \mid T = 0, X] - \eta_0(X)P(T = 1 \mid X) \\ &= CATE^*(X) - \eta_1(X)P(T = 0 \mid X) - \eta_0(X)P(T = 1 \mid X) \end{aligned}$$

$$
\begin{aligned}
ATE &= \mathbb{E}[CATE(X)] \\
&= \mathbb{E}[CATE^*(X)] - \mathbb{E}[\eta_1(X)P(T=0 \mid X) + \eta_0(X)P(T=1 \mid X)] \\
&= ATE^* - \mathbb{E}[\eta_1(X)P(T=0 \mid X) + \eta_0(X)P(T=1 \mid X)]
\end{aligned}
$$

A.2 Derivations for the Direct Dependence Method

$$
\begin{aligned}
&\mathbb{E}[Y(t) \mid \mathbf{X} \in l] \\
&= \mathbb{E}_X[\mathbb{E}_Y[Y(t) \mid \mathbf{X} \in l, X] \mid \mathbf{X} \in l] \\
&= \mathbb{E}_X[\mathbb{E}_Y[Y \mid \mathbf{X} \in l, X, T=t] \mid \mathbf{X} \in l] \\
&= \mathbb{E}_X[\mathbb{E}_Y[Y \mid \mathbf{X} \in l, X, T=t] \mid \mathbf{X} \in l] + \mathbb{E}[Y \mid \mathbf{X} \in l, T=t] - \mathbb{E}[Y \mid \mathbf{X} \in l, T=t] \\
&= \mathbb{E}[Y \mid \mathbf{X} \in l, t] + \mathbb{E}_X[\mathbb{E}_Y[Y \mid \mathbf{X} \in l, X, t] \mid \mathbf{X} \in l] - \mathbb{E}_X[\mathbb{E}[Y \mid \mathbf{X} \in l, X, t] \mid \mathbf{X} \in l, t] \\
&= \mathbb{E}[Y \mid \mathbf{X} \in l, t] + \sum_x \underbrace{\mathbb{E}_Y[Y \mid x, t]}_{\text{unknown}} (p(x \mid \mathbf{X} \in l) - p(x \mid \mathbf{X} \in l, t)) \\
&\leq \mathbb{E}[Y \mid \mathbf{X} \in l, t] + \left| \sum_x \mathbb{E}_Y[Y \mid x, t] \, (p(x \mid \mathbf{X} \in l) - p(x \mid \mathbf{X} \in l, t)) \right| \\
&\leq \mathbb{E}[Y \mid \mathbf{X} \in l, t] + \sup_{f \in \mathcal{F}} \left| \sum_x f(x) \, (p(x \mid \mathbf{X} \in l) - p(x \mid \mathbf{X} \in l, t)) \right| \\
&= \mathbb{E}[Y \mid \mathbf{X} \in l, t] + \mathrm{MMD}_{\mathcal{F}}(p(x \mid \mathbf{X} \in l), p(x \mid \mathbf{X} \in l, t))
\end{aligned}
$$

A.3 Derivations for the Propensity Method

$$
\begin{aligned}
&\mathbb{E}_Y[Y(t) \mid \mathbf{X} \in l] \\
&= \sum_y y P(Y(t) = y \mid \mathbf{X} \in l) \\
&= \sum_x \sum_y y P(X = x, Y(t) = y \mid \mathbf{X} \in l) \\
&= \sum_x \sum_y y \frac{P(X = x, Y(t) = y \mid \mathbf{X} \in l, T = t)}{P(X = x, Y(t) = y \mid \mathbf{X} \in l, T = t)} P(X = x, Y(t) = y \mid \mathbf{X} \in l) \\
&= \sum_x \sum_y y \frac{P(X = x, Y(t) = y \mid \mathbf{X} \in l)}{P(X = x, Y(t) = y \mid \mathbf{X} \in l, T = t)} P(X = x, Y(t) = y \mid \mathbf{X} \in l, T = t) \\
&= \mathbb{E}_{X,Y}\left[\frac{P(X, Y(t) \mid \mathbf{X} \in l)}{P(X, Y(t) \mid \mathbf{X} \in l, T = t)} Y(t) \mid \mathbf{X} \in l, T = t \right] \\
&= \mathbb{E}_{X,Y}\left[\frac{P(X \mid \mathbf{X} \in l) P(Y(t) \mid X, \mathbf{X} \in l)}{P(X \mid \mathbf{X} \in l, T = t) P(Y(t) \mid X, \mathbf{X} \in l, T = t)} Y(t) \mid \mathbf{X} \in l, T = t \right] \\
&= \mathbb{E}_{X,Y}\left[\frac{P(X \mid \mathbf{X} \in l)}{P(X \mid \mathbf{X} \in l, T = t)} Y(t) \mid \mathbf{X} \in l, T = t \right] \\
&= \mathbb{E}_{X,Y}\left[\frac{P(T = t \mid \mathbf{X} \in l)}{P(T = t \mid X)} Y(t) \mid \mathbf{X} \in l, T = t \right] \\
&= \mathbb{E}_{X,Y}\left[\frac{P(T = t \mid \mathbf{X} \in l)}{P(T = t \mid X)} Y \mid \mathbf{X} \in l, T = t \right]
\end{aligned}
$$