Causal Sensitivity Analysis for Decision Trees

by

Chengbo Li

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Ventilator assignments in the pediatric intensive care unit (PICU) are made by medical experts; however, for some patients the relationship between ventilator assignment and patient health status is not well understood. Using observational data collected by Virtual PICU Systems (VPS) (58,772 PICU visits with covariates and different ventilator assignments conducted by clinicians), we attempt to identify which patients would derive the greatest clinical benefit from ventilators by providing a concise model to help clinicians estimate a ventilator's potential effect on individual patients, in the event that patients need to be prioritized due to limited ventilator availability.

Effectively allocating ventilators requires estimating the effect of ventilation on different patients; this is known as individual treatment effect estimation. However, we only have access to non-randomized data, which is confounded by the fact that sicker patients are more likely to be ventilated. In order to reduce bias due to potential confounding to estimate the average treatment effect, propensity score matching has been widely studied and applied to estimate the average treatment effect, which matches patients from treated group with patients from control group based on similar conditional probability of ventilator assignment given an individual patient's features. This matching process assumes no unmeasured confounding, meaning there must be no unobserved covariates influencing both treatment assignment and patient's outcome. However, this is not guaranteed to be true, and if it is not, the average treatment effect estimation using propensity score matching approach can be fragile given an unmeasured confounder with strong influences.

Rosenbaum and Dual Sensitivity Analysis is specifically designed for potential unmeasured confounder problems in propensity score matching, assuming confounder's existence it evaluates how "sensitive" the treatment effect estimation after matching can be. This sensitivity analysis method has been well-studied to evaluate the estimated average treatment effect based on propensity score matching, specifically, using generalized linear models as the propensity score model.

However, both estimating treatment effect via propensity score matching and its sensitivity analysis have their limitations: first, propensity score matching only helps in estimating the average treatment effect, while it does not provide much information about individual treatment effect on each patient; second, Rosenbaum and Dual Sensitivity Analysis only evaluates the robustness of estimated average treatment effect from propensity score matching, while it cannot evaluate the robustness of a complex model estimating the individual treatment effect, such as a decision tree model.

To solve this problem, we attempt to estimate the individual treatment effect from observational study, by proposing the treatment effect tree (TET) model. TET can be estimated through learning a Node-Level-Stabilizing decision tree based on matched pairs from potential outcome matching, which is a matching approach inspired by propensity score matching. With synthetic data generated to mimic the real-world clinical setting, we show that TET performs very well in estimating individual treatment effect, and the structure of TET can be estimated by conducting potential outcome matching in observational data.

There is a matching process in TET estimation, and to evaluate the robustness of the estimated TET learned through potential outcome matching in observational data, we propose an empirical sensitivity analysis method to show how sensitive the estimated TET's structure and predictive power can be in situations with strong levels of confounding described by Rosenbaum and Dual Sensitivity Analysis. We use the same synthetic dataset with different levels of confounding encoded as boolean confounders to experiment with this sensitivity analysis method. We show the experimental results of estimating TET from observational data, as well as their performances in sensitivity analysis. The experimental results show that with strong covariates setting, the estimated TET from observational data can be very stable against strong levels of confounding described by Rosenbaum and Dual Sensitivity Analysis encoded as boolean confounders.

In this work, we propose TET model for individual treatment effect estimation with observational data, we show that TET can be learned from matching individuals based on potential outcome. We designed an empirical sensitivity analysis method to evaluate the robustness of TET with different levels of confounding described by Rosenbaum and Dual Sensitivity Analysis, and the experimental results show the learned TET can be stable against strong levels of confounding.

Acknowledgements

First, I would like to thank my supervisor Professor Daniel Lizotte, you have been the greatest mentor to me ever since my first day in the field of computer science, and you helped and guided me through all these obstacles and difficulties during my studies. This work would never have been possible without you. It is a great honor to be your student, and I thank you for being a teacher with so much patience and a professor being so cool from the bottom of my heart.

I would like to thank Professor Jesse Hoey and Professor Peter van Beek for being my reviewers, thank you for your valuable feedbacks and suggestions to this work.

I would like to thank Dr. Randall Wetzel at Children's Hospital Los Angeles and Virtual PICU Systems for granting us access to the real-world VPS data for our research, I can only imagine the difficulties estimating treatment effect without any data, and I feel glad that we have access to this valuable VPS data so we can fully focus on developing theories without worry much about lacking of real world data.

Finally, I would like to thank my family and my very close friends for your support, you made me believe it is such a wonderful thing to live in this world.

Dedication

To the ones that I love.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis is motivated by a *triage problem*, that is, given a population of patients, identify the patients who would benefit the most from treatment. In cases where treatment is a scarce resource, solving the triage problem can help decision-makers decide how to best allocate available resources. The details of the specific triage problem in which we are interested are given below in Section 1.1.

Before addressing the triage problem, we begin by briefly reviewing fundamental concepts and methods for estimating treatment effects. In the field of biostatistics, the *treatment effect*, defined as the "comparisons of the outcome with treatment and the outcome without treatment" [20], is the key to evaluate how a treatment influences health outcomes. Here, we define treatment as a boolean variable assigned to individuals in a population, and in general, the outcome refers to a scalar describing the health status at some point after the treatment assignment. The treatment effect tells us if the treatment, such as a new medicine or a new operation, causally influences the outcome, such as patient's health status. Thus, the treatment effect is one of the most important quantities to estimate when designing new treatments or evaluating existing ones.

However, estimating the treatment effect of a specific treatment on an individual is not an easy task, from both the medical and statistical points of view, because we can never observe the *counterfactual* outcome; that is, we can never observe "what would have happened" if a treated patient had not been treated, or vice-versa. Instead, researchers often estimate an *averaged* treatment effect by comparing "similar" populations. Ideally these populations are generated by randomization, which means treatment is randomly assigned to patients in the population before observing their outcomes. Doing so ensures that the population of treated patients and the population of non-treated or *control* patients are similar if the sample sizes are large enough. However, randomly giving treatments can be expensive or even unethical, especially when the outcome can be a patient's mortality status that might be negatively influenced by treatment.

In cases where randomized experiments are not possible, we may use *observational* data to estimate treatment effects. Here the term "observational" indicates that the experimenters do not assign treatment; assignment is controlled by clinicians, such that clinicians can assign different treatments to different patients with different health status according to their judgement. We refer to this as the "treatment policy." Normally, a clinician, or more precisely the treatment policy, tends to assign treatment to a sub-group of patients that is believed to benefit most from the treatment. To describe this mathematically, if we collect a group of covariates ("features" in machine learning) from each patient to describe this patient's characteristics that can influence the clinician's decision of assigning a treatment, the observational setting essentially means that treatments are assigned in an unknown (but potentially learnable) fashion based on patient's covariates, instead of according to a known policy as in the randomized experiment setting. But, from the statistical point of view, this observational setting can make even the average treatment effect extremely difficult to estimate because the group of patients who received treatment and the group of patients who did not may be dramatically different. Thus, difference between average outcomes in the two groups might not be caused by treatment itself, but by the selection process that determines whether a patient is treated.

One of the most widely used methods to estimate the average treatment effect from an observational study is *matching* [22], which in general uses a set of observational data to construct treated and control groups using individuals that have been "matched" with each other by their covariates so that the generated groups are similar to one another. One can then estimate average treatment effects in the same way as one would for a randomized trials. There are many ways to achieve matching, among them *propensity score matching* [20], which is widely studied and used. The *propensity score* of an individual is that individual's probability of receiving treatment according to the treatment policy that generated the data. Based on the *strongly ignorable assumption* [20], it selects a subset of control population through matching on propensity score, which is learned from the data. Propensity score matching and its applications are used in a wide variety of fields including medicine [2], marketing strategy design [9], political evaluation, and many other [8].

Propensity score matching can only produce unbiased estimates of the average treatment effect if there are *no unmeasured confounders*, that is, if in our observational study we have all of the covariates we need about each patient to correctly estimate their probability of receiving treatment. Given an estimated average treatment effect provided by propensity score matching, we typically want to know "How much we can trust this estimation?" To answer this question, one can conduct *sensitivity analysis* [19]. Such analyses consider how our estimates would change if in fact the assumption of no unmeasured confounding were false. In other words, they attempt to evaluate and show how "sensitive" the estimated treatment effect can be to any potential unmeasured confounders. Rosenbaum Sensitivity Analysis and its variation, Dual Sensitivity Analysis, are used to evaluate the sensitivity of the estimated treatment effect provided by propensity score matching to unmeasured confounding by assuming that the treatment policy is estimated using a Generalized Linear Model (GLM) [18].

Propensity score matching and related sensitivity analyses allow us to estimate *average* treatment effects using *observational data* and assess *sensitivity* to unmeasured confounding. In order to solve our problem of interest, the *triage problem*, we wish to estimate *individual* treatment effects using *observational data* and still assess *sensitivity* to unmeasured confounding. To this end, we make four main contributions:

1. We define the *Treatment Effect Tree* (TET) which we use to define and learn individual treatment effects.
2. We show that the TET *cannot* be estimated using propensity score matching, and present an alternative procedure for estimation.
3. We develop a new method for Dual Sensitivity Analysis of the estimated TET that is based on previous work using Generalized Linear Models.
4. We illustrate the use of our approach on synthetic data and on observational data collected by Virtual PICU Systems (VPS) concerning the ventilator triage problem.

1.1 Motivation

Our work is motivated by the need for evidence-based decision support for the ventilator triage problem. Mechanical ventilators are widely used in Intensive Care Units (ICU), and are a critical treatment that can prevent severely sick ICU patients from dying. However, mechanical ventilators can also cause side-effects, such as barotrauma and lung injury to patients, thus a ventilation decision should be carefully made based on a patient's specific status, in order to maximize the ventilator's positive effect and minimize its side-effects on the patient.

The ventilator triage problem refers to the situation of a limited number of ventilators in ICU, such that in extreme scenarios (e.g. natural disasters) an ICU might not have enough mechanical ventilators to give to all patients who would, under ordinary circumstances, receive them. In this setting we wish to allocate ventilators to patients for whom the treatment effect is greatest; that is, we wish to allocate them to the individual patients who would benefit the most. In our case, we focus on patients potentially having higher chances of surviving with ventilations. This is essentially the individual treatment effect estimation problem we described above.

Given data collected by Virtual PICU Systems (VPS) from different paediatric hospitals ICUs, we have access to 58,772 individual patients with each patient's status recorded as 142 covariates of boolean, categorical, and continuous-valued types; each patient's ventilation decision made by clinicians was also recorded as a boolean variable in the data, and each patient's mortality status after ventilation was also recorded and included as a boolean indicator in the data. We present an analysis of these data in Chapter 3.

1.2 Thesis Structure

In Chapter 2, we review the essential components from both statistical and computer science fields upon which our work is based. These include logistic regression, basic causal inference terminology specific to our setting, matching, sensitivity analyses, and decision trees.

In Chapter 3, we introduce the concept of Individual Treatment Effect (ITE) and define the category of problems that require estimating ITE. After that, we introduce one method to estimate ITE from observational data, which is getting the Treatment Effect Tree (TET) by training on matched pairs of individuals. As mentioned earlier, propensity score matching is one of the most popular methods for estimating average treatment effects; however we show that the TET cannot be estimated from propensity score matching alone, with both theoretical analysis and experimental results, and we explain why matching on estimated outcome can provide reasonable ITE estimation. We also propose our empirical method of conducting sensitivity analysis on TET with different levels of confounding with respect to the confounding concept defined by Rosenbaum and Dual sensitivity analysis. We use an experiment to show how this new mechanism works on data collected from an observational study.

In Chapter 4, we provide the experimental results using the methods in Chapter 3, showing that the TET can be estimated from matching based on estimated outcomes, and by showing how the true TET and the estimated TET differ with different levels of confounding, we show our sensitivity analysis on decision trees can provide a straightforward description of how confounding may influence the estimated TET structures. We also show experiment results of applying TET estimation and sensitivity analysis on the estimated TET with real-world clinical data collected by VPS.

Finally, we conclude our experiment results and include our thoughts during the experiment, with the future plan of this work in Chapter 6.

Chapter 2

Background

In this Chapter, we review the fundamental concepts of logistic regression, which is crucially important in estimating the average treatment effect via propensity score matching. We briefly introduce what logistic regression tries to estimate and how it works.

2.1 Logistic Regression Models

Being categorized as a Generalized Linear Model (GLM) [10], logistic regression models [7] are designed to solve classification problems. Similar to any regression problem, a classification problem also requires the prediction of target y given inputs x, but the target value y can take only a few discrete values instead of a continuous value as in linear regression. In this section we focus on binary classification problem, whose target value y can be either 1 or 0, namely, positive $\oplus$ and negative $\ominus$ labels of data instances.

Instead of directly returning the final prediction of labels, logistic regression produces an estimated conditional probability of label being positive based on input x and a hypothesis h. This is computed by multiplying a parameter vector $\theta = \langle \theta_0, \theta_1, ..., \theta_n \rangle$ by an input vector x and transforming the result to the range $(0, 1)$ as follows:

$$h_\theta(x) = S(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{2.1}$$

in which,

$$S(z) = \frac{1}{1 + e^{-z}} \tag{2.2}$$

is the sigmoid function, also known as the logistic function.

Logistic regression training aims to find the optimal hypothesis h that can predict correct labels as well as possible, using the criterion of conditional likelihood. Given the setting that the hypothesis returns the estimated conditional probability of being labeled positive, we have

$$\begin{aligned} Pr(y=1|x,\theta) &= h_\theta(x) \\ Pr(y=0|x,\theta) &= 1-h_\theta(x) \end{aligned} \tag{2.3}$$

or equivalently,

$$Pr(y|x,\theta) = (h_\theta(x))^y (1-h_\theta(x))^{1-y}. \tag{2.4}$$

Assuming that the m instances in the training data set are all drawn independently from each other, the likelihood of hypothesis parameter vector θ can be expressed as

$$\begin{aligned} \mathcal{L}(\theta) &= Pr(\vec{y}|X,\theta) \\ &= \prod_{i=1}^{m} Pr(y^{(i)}|x^{(i)},\theta) \\ &= \prod_{i=1}^{m} \Big((h_\theta(x^{(i)}))^{y^{(i)}} (1-h_\theta(x^{(i)}))^{1-y^{(i)}} \Big) \end{aligned} \tag{2.5}$$

such that the optimal hypothesis parameters θ maximizes the likelihood above. Similar to the likelihood analysis in linear regression [12], instead of directly maximizing the likelihood, we maximize the log likelihood:

$$\begin{aligned} \ell(\theta) &= \log \mathcal{L}(\theta) \\ &= \sum_{i=1}^{m} \big(y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)})) \big) \end{aligned} \tag{2.6}$$

This function may then be maximized over θ using any algorithm appropriate for smooth functions. (E.g. Newton-Raphson.)

2.2 Tools for Observational Data

In this section, we introduce basic terminology used in the study of estimating treatment effects from observational data, and we define some basic symbols that we will use throughout the thesis.

2.2.1 Definitions and Terms

X

The covariates (or features) describing a patient. In our case X describes a patient's health status and characteristics. x_i refers to the covariate vector of observation/patient i in the data set. In general, x_i can have different types of elements, e.g. boolean type, integer type, continuous value type, category type, etc. However, the type of each element in X must be consistent across all observations.

Z

The treatment assignment. For our application, Z describes the ventilation decision made by clinicians in the ICU. z_i refers to the specific treatment assignment of observation/patient i. In general, Z contains only boolean values, since for each observation i either receives a treatment (z_i encoded as *True* or 1, observation i is "treated"), or not (z_i as *False* or 0, observation i is "control"). We let N denote the total number of observations we have. We also let N_1 and N_0 be the number of treated and control observations, respectively.

R^0 and R^1

The potential outcomes. These contain, for each patient, the outcome of the patient *under both the treated and the control condition.* Note that for any given patient, only one of these is ever observed. r_i^0 refers to observation/patient i's outcome under the control condition, and r_i^1 refers to observation/patient i's outcome under the treatment condition. In our setting, both R^0 and R^1 are boolean.

R

The *observed* outcome. For a treated patient, this is R^1. For a control patient, this is R^0. (The meaning of R depends on the value of Z.)

U

A hypothesized unmeasured confounder. It is never observed, though we may make assumptions about its relationships to Z, R^1, and R^0. It describes a hidden covariate outside of X that influences both the treatment assignment Z and the potential outcomes R^1 and R^0. U is assumed to be independent of X, that is, one cannot estimate U based on X. In our setting, we assume the potential confounder is boolean.

2.2.2 Treatment Effects

The treatment effect is a scalar quantity describing how much the outcome, such as company's earning or patient's health condition, is affected by the treatment, such as a marketing action or a medical treatment. Given an individual i, the treatment assignment of i, which is the treatment i received, is denoted as z_i, and the outcome with and without treatment is denoted as as r_i^1 and r_i^0. The effect of treatment z_i on individual i is defined as the difference between two outcomes with different treatments:

$$\begin{aligned} \tau_i &= (r_i|z_i = 1) - (r_i|z_i = 0) && (2.7) \\ &= r_i^1 - r_i^0. && (2.8) \end{aligned}$$

The *average treatment effect* (averaged over individuals) can then be defined as the average difference between the two estimated outcomes with different treatments:

$$\begin{aligned} \text{ATE} &= \frac{1}{N}\sum_{i=1}^{N}\tau_i && (2.9) \\ &= \frac{1}{N}\sum_{i=1}^{N} r_i^1 - r_i^0 && (2.10) \\ &= \frac{1}{N}\sum_{i=1}^{N} r_i^1 - \frac{1}{N}\sum_{j=1}^{N} r_j^0. && (2.11) \end{aligned}$$

By averaging each individual's treatment effect across all individuals, we can estimate the average treatment effect over the population. Note that this can be interpreted also as subtracting the averaged control outcomes from the averaged treatment outcomes.

Estimating ATE

Clearly, the treatment effect τ_i on individual i is impossible to identify in the real world, no matter whether in a randomized or non-randomized experiment, for each individual i only one of the potential outcomes, either r_i^0 or r_i^1 can be observed, but never both of them. One of them has to be counterfactual. To solve this problem, we may try to estimate one of the potential outcomes. The estimated treatment effect on individual i can then be expressed as the difference between one observed and one estimated potential outcome:

$$\hat{\tau}_i = r_i^1 - \hat{r}_i^0 \tag{2.12}$$

or

$$\hat{\tau}_i = \hat{r}_i^1 - r_i^0 \tag{2.13}$$

Now consider the population in our setting: given a sample of $N = N_1 + N_0$ individuals, treatment z_i and the observed outcome r_i which may be r_i^0 or r_i^1 depending on z_i, we can estimate the ATE by

$$\widehat{\text{ATE}} = \frac{1}{N_1} \sum_{z_i=1} r_i^1 - \frac{1}{N_0} \sum_{z_i=0} r_i^0 \tag{2.14}$$

By averaging values from two subgroups, we no longer require counterfactual outcomes for each individual. Also note that the treated and control subgroups might be of different sizes. If the treatment and control groups come from the same distribution, as in a randomized trial, $\widehat{\text{ATE}}$ will be unbiased.

The *Average Treatment Effect on the Treated* (ATT) is similar to the ATE, but instead of looking at the whole population, this time we only focus on the comparisons on the treated subgroup

$$\frac{1}{N_1} \sum_{i \in P_t} (r_i^1 - \hat{r}_i^0) \tag{2.15}$$

in which $\hat{r}_i^0$ is typically estimated using an observed outcome r_j^0 from the control group. When the treated and control groups are the same size and have the same distribution, ATT = ATE. Otherwise, estimating the ATT requires a reliable estimation of the average counterfactual outcome, specifically, the expected outcome without treatment, for individuals i in the treated group. One way to estimate such a counterfactual outcome is to match individuals in the treated group with individuals in the control group, as long as the matched pair are believed to share a similar r^0. The ATT is of interest because it estimates the averaged effect of treatment among the *treated* sub-population; this is particularly relevant for triage if we are making decisions about patients who would ordinarily be treated.

2.3 Matching

As we have seen, estimating the ATT requires that we estimate counterfactual outcomes for individuals in the treated group. One of the most popular methods for this is called *matching.* One view of matching is that it attempts to take observational data and produce a new dataset that would have been created by a randomized experiment. The main idea of matching is that estimating counterfactuals can be achieved by matching up individuals with similar covariates from the different treated and control groups in observational data as shown in Figure 2.1. If the well-matched individuals from different groups share the same or similar covariates, it is reasonable to consider their treatments as randomly assigned in this matched group. Several methods of finding well-matched pairs from treated and control groups are all called "matching." This idea can also be viewed simply as trying

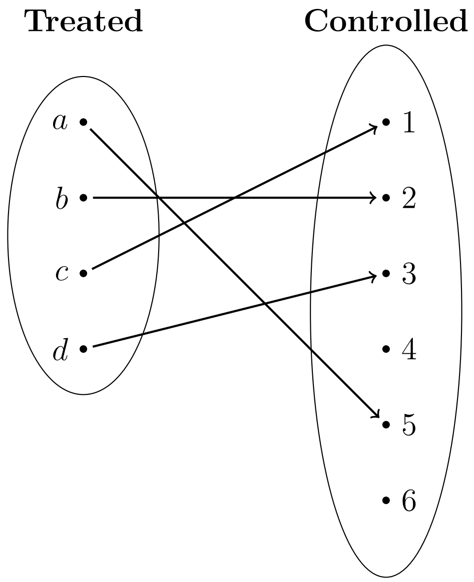


Figure 2.1: Matching between treated and controlled groups

to re-balance the covariate distribution bias in the treated and control groups due to the treatment selection to simulate a randomized experiment, in order to finally compute causal effects on the rebalanced-observational data. Matching has been widely used in the two following scenarios based on different types of observational data:

1. The first scenario is the one in which the data contains each individual's treatment assignment and covariates, but the outcome values are not available or still remain unknown. In this scenario, matching is primarily used to select appropriate individuals for follow-up studies, which may or may not involve the future outcomes. Though this is not relevant to the purpose of using matching for causal effect estimation, it was the setting for most original works on matching [15].

2. The second scenario is the one in which the data contains treatment assignments, covariates, as well as outcome values. In this scenario, matching is widely used to balance the treated and control groups for treatment effect estimation by reducing covariate bias in these two groups.

2.3.1 Strongly Ignorable Assumption

One of the important assumptions required for matching to be effective is the Strongly Ignorable Treatment Assignment Assumption [20]. When data are properly collected from a randomized experiment, x is known to include all covariates that are both used to assign treatment z, and possibly related to the control and treated outcomes (r^0, r^1). Which implies the treated and control outcomes (r^0, r^1) and treatment z are conditionally independent given covariates x:

$$(r^0, r^1) \perp\!\!\!\perp z | x.$$

However, this is often not the case for non-randomized experiments. For a non-randomized setting, given a vector x containing some covariates, the treatment assignment is considered strongly ignorable if for all possible instances of x the following statement holds true:

$$(r^0, r^1) \perp\!\!\!\perp z | x \text{ and } 0 < Pr(z = 1|x) < 1 \text{ for all } x$$

Briefly, if the statement above still holds, then we say the treatment assignment is strongly ignorable, which requires that (1) each individual in the population has a *non-zero* chance of being treated, and (2) treatment z and potential outcomes (r^0, r^1) are independent given a set of covariates x. This assumption allows us to get consistent estimates of a treatment effect in observational studies by adjusting for the observed covariates, for example by matching[16]. With this assumption, it becomes reasonable to match individuals from different treatment assignment groups based on their similar covariates x, such that the matched pairs can be regarded as receiving treatments through a randomized experiment.

2.3.2 Covariate-based Matching

Considering the fact that τ_i of a treated individual i is the difference between observed outcome r_i^1 of i being treated and the unobserved outcome r_i^0 of i not being treated, the last missing piece that remains to be the estimated is the outcome $\hat{r}_i^0$ of i not being treated, for which we can never truly observe. Generally, all matching methods tend to estimate this unknown expected outcome by looking at the control group and finding each treated individual i with at least one control "paired" individual j from the control group, such that we generate two sub-population using i and j, and these sub-populations share the maximum of "similarity" of covariates distribution. In this section, we will discuss how to measure "similarity" in order to find reasonable matches for all treated individuals.

Definition of distances

Rather than define the "similarity between individuals," it is typically easier to define the difference between individuals and use it to find the "least different" individuals for matching. There is a variety of distance definitions, and each definition of distance will lead to a different matching procedure. Here we review four commonly-used definitions of distance $D_{i,j}$ between individuals i and j.

Exact Distance Exact distance is an intuitive definition based on the concept of covariate similarity; in fact, instead of finding "similar individuals with different treatment assignment," it directly finds differently treated individuals with exactly the same covariates:

$$D_{i,j} = \begin{cases} 0 & \text{if } x_i = x_j \\ \infty & \text{if } x_i \neq x_j. \end{cases}$$

Though we can imagine the pairs matched according to this exact distance must be most reliable, since there is no other "similarity" better than being exactly the same, it is only applicable to very special situations such that we have enough control individuals sharing the exact same covariates as treated patients, which is not applicable in general. In fact, if we push this property to the extreme, every treated patient can be matched with a controlled patient using an exact distance match. These matched pairs together can be regarded as drawn from a perfectly randomized experiment, if the covariates include all factors influencing treatment assignment.

Mahalanobis Distance Mahalanobis distance is a more complex definition which makes use of distribution of sample covariates in a population:

$$D_{i,j} = (x_i - x_j)^T \text{cov}^{-1}(X_c)(x_i - x_j)$$

When Mahalanobis distance is used to estimate the ATT, $cov(X_c)$ refers to the sample variance-covariance matrix of X estimated from the control group. The difficulty of applying this definition in matching is that it does not work very well in situations where X is high-dimensional and each covariate is not normally distributed, which is unfortunately the case in most observational studies. Gu and Rosenbaum [6] point out the reason Mahalanobis does not perform well in such cases is due to its assumption of regarding all interactions among covariates in X as equally important, but this assumption is weak since they are very likely to have different weights for different interactions among them.

2.3.3 Propensity Score Matching

Perhaps the most widely-used matching criterion, propensity score distances are defined based on conditional treatment probabilities,

$$D_{i,j} = dist(Pr(z_i = 1|x_i), Pr(z_j = 1|x_j))$$

where *dist* is some function describing the distance between two probabilities. Recall that $Pr(z_i = 1|x_i)$ is the conditional probability of an individual being treated given the observed covariates. We will discuss the use of propensity scores in the next section.

The idea of propensity scores was first introduced by Rosenbaum and Rubin [20]. Instead of focusing on the differences and interactions among all covariates in X for scaling by pushing the statistics into high-dimensional spaces (e.g. Mahalanobis distance), propensity score takes the opposite direction of summarizing all covariates into a single quantitative scalar, which is an individual's conditional probability of being treated given the observed covariates.

Consider the fact that for each specific propensity score value, these two sub-populations of individuals with the same propensity score from treated and control groups should have the "same distribution", in this case it does not mean these two individuals are exactly the same, but in a sense that tahey all share the same conditional probability of being treated. From this point of view, matching with propensity score can also be interpreted as conducting several mini-scale randomized experiments by resampling new treated and control groups from the original treated and control populations.

Compared with exact distance, the advantage of propensity score is obvious: finding pairs with the same conditional probability is a more tolerant condition than exact covariates, so it is more likely to produce a greater amount of meaningful matched-pairs, as long as we have the conditional probability estimated properly.

However, it would be impossible to get access to the true conditional probabilities (propensity score) without a randomized experiment, therefore such probabilities must be estimated from observational data. Generally, we may utilize any statistical model that takes covariates as inputs and gives predicted binary values as outputs as a propensity score model, and the most widely used model is logistic regression to regress treatment on covariates

$$\log\left(\frac{P_i}{1 - P_i}\right) = \beta_0 + \sum_{x_k \in x_i} \beta_k x_k$$

which is based on the assumption that x_i captures all factors that influence the treatment of individual i. Notice that we do not involve the outcome of i into the propensity score model, due to the Strongly Ignorable Treatment Assignment Assumption we mentioned before, which assumes the two potential outcomes (R_0, R_1) and the treatment assignment z_i should be conditionally independent given covariates x_i.

Matching individuals with similar propensity score from different treatment assignment groups performs well in selecting observations to construct "new" treated and control sub-populations, such that comparing the average outcome between these two sub-populations can be a good estimation of ATT. However, *matching individuals on propensity score does not guarantee that the outcome differences of each matched pair (i,j) is a good estimation of the counterfactual outcome of either individual i or j*, as we will see in our experiment in Chapter 4.

2.3.4 Matching Methods

With a distance metric chosen, the next step to conduct matching is the matching process itself, which in general refers to matching each treated individual with one or several control individuals, and there is a great variety of methods to achieve this matching process. In this section, we introduce one major popular group of methods for conducting matching. One intuitive matching method is to match each treated to individual k control individuals with the minimal distances according to the chosen distance measurement. Known as the $k : 1$ nearest neighbor matching, this is probably the most commonly used approach to find matched pairs.

A particularly variation of the $k : 1$ nearest neighbor matching is when k is set to 1, which means each treated individual will be matched with only one control individual. This one-to-one nearest matching approach is popular for its effective reduction of the bias between treated and control groups, as well as its ease of implementation. Though there are complaints regarding its only picking one being essentially equivalent to disregarding a great number of individuals in controlled group. However, one-to-one matching can also be seen as an ordinary $k : 1$ matching method with a even stricter condition — only the most similar individuals will be matched as a pair. Considering the fact that in most cases we do have a larger controlled group, one-to-one nearest neighbor matching does provide a more precise match result with higher balance quality. When choosing one-to-one nearest neighbor matching, one must consider in what order shall the treated individuals be matched, for the change of order might vary the overall matching quality. Many modifications of the above "greedy" one-to-one matching method can solve this problem well, such as finding match pairs minimizing the overall distances between treated and matched control groups, known as the optimal one-to-one matching.

Another way to solve the problem is to allow matching with replacement; that is, each controlled individual can be matched to more than one treated individual if necessary. This approach works especially well in situations with insufficient controlled individuals, and we no longer need to worry about greedy matching, since each matched individual is still available to be matched, hence, previous matches do not influence future potential matches. However, we need to pay extra attention when using one-to-one matching with replacement: due to the possibility that one controlled individual can be matched multiple times, the causal inference results, such as the estimate of ATT, might be heavily influenced by only a few controlled individuals matched many times. To avoid this, we must take the match frequency of each matched individual into account when conducting one-to-one matching with replacement.

2.4 Sensitivity Analysis

As an important statistical tool, sensitivity analysis is the study of how the uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input [19]. In this chapter, we will discuss sensitivity analysis in detail, as well as its application to propensity score matching.

2.4.1 Unobserved Covariate

As we have discussed in Section 2.3.1, no matter which specific model is applied, all matching methods share the same goal of balancing the treatment bias caused by the non-randomized treatment assignment process. And in order to reduce this bias, it will match individuals according to their covariates. In other words, the general idea of matching is primarily based on the assumption that covariates X are sufficient to capture all information of the treatment assignment process, hence we are able to represent the treatment assignment process by building the treatment policy from X and Z. The assumption above guarantees that if individuals are matched based on similar X or other reasonable statistics of X, the differences of potential outcomes R among these matched individuals are only due to the remaining difference between these matched individuals—the treatment assignment Z itself. With all of these causalities above, we may draw a Bayesian Network to illustrate this assumption.

More precisely, the assumption requires that the Bayesian Network (See Figure 2.2) is a complete representation of all the causalities between X, Z, and R. There is no other

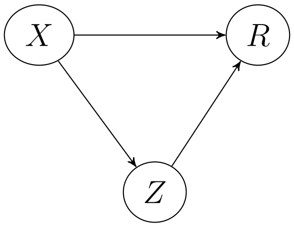


Figure 2.2: Bayesian Network of Matching Assumption

factor except X to influence Z, and R is only influenced by X and Z, namely, there is no hidden covariate which is not included in X that influences Z and R. If this assumption of no hidden covariate holds true, then we are confident to say matching with a model learned from X is a valid approach for causal inference in observational settings.

However, this very strong assumption is often not guaranteed in an observational setting. For one reason, the treatment assignment process might not be fully recorded in the data, which will leave the possible existence of one or even several hidden covariates out of X. Another reason could be the involvement of another hidden covariate that remains invisible during the treatment assignment process, such as assigning Z whose distribution cannot be captured by X alone.

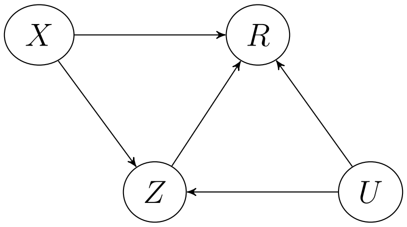


Figure 2.3: Bayesian Network with Confounder U

The possible existence of one or more hidden covariates, known as the confounders U, will invalidate the previous assumption. Shown by another Bayesian Network with hidden covariate (See Figure 2.3), if one or several confounders U is not included in X, and at the same time U is marginally independent of X so we cannot represent/predict U from X alone, then the information X contains by itself is not sufficient to represent the treatment assignment process. As a result of the hidden U, any matched pair based on X alone can be actually very different with respect to this confounder U. This causal uncertainty makes it impossible to determine if the difference in outcomes R is due to the different treatment assignments Z, or to different values of the confounder U, since U also influences the outcome R.

Considering the possible existence of confounder U, a common concern regarding matching's reliability is to question whether the readjustment of observational data by matching solely based on X alone will fail to account for hidden confounder U, which was not measured and included during matching process, and if such a confounder U actually exists, how much might it vary this readjustment to harm the original matching's performance.

2.4.2 Rosenbaum Sensitivity Analysis

One method that attempts to answer the previous question is the Rosenbaum Sensitivity Analysis [17] [18]. As illustrated by its name, Rosenbaum Sensitivity Analysis is used to estimate a quantifiable increase of uncertainty among all matched groups that one or several hidden confounders U will bring. This can also be interpreted as how "sensitive" the matching is to potential hidden confounders.

The idea behind Rosenbaum Sensitivity Analysis is straightforward and empirical: assume we have a set of data X, Z, R, with these data we learn a matching model h_1 solely based on X alone, then we get all matched groups according to h_1, such that individuals in the same matched group have the same estimated conditional probability of being treated given covariates X. After matching, we suspect the existence of a confounder U; however we have no access to U's specific values.

If we consider the matching model h_1 as a readjustment of individuals data points between treated and control groups, then we know that with a new matching model h_2 based on X and U together[1], the readjustment from h_2 is supposed to be different from and more precise and reliable than h_1. Or, U will change some of the readjustments in h_1, such as changing individual data points into different matched groups, and in turn, produce different estimates of counterfactuals.

However, it is impossible to actually learn h_2 and compare it to h_1, since we do not know the values of U. Drawing back to the very original goal of analyzing a matching model's reliability, Rosenbaum Sensitivity Analysis says the specific h_2 model is not necessary for evaluating h_1's sensitivity to U. As a matter of fact, what we want to know is how different the matching results from h_1 can be [18], if we manually change the matching readjustment of h_1, assuming a confounder U is swapping individual data points in different matched pairs.

[1] In this case, there is no other hidden covariates besides U.

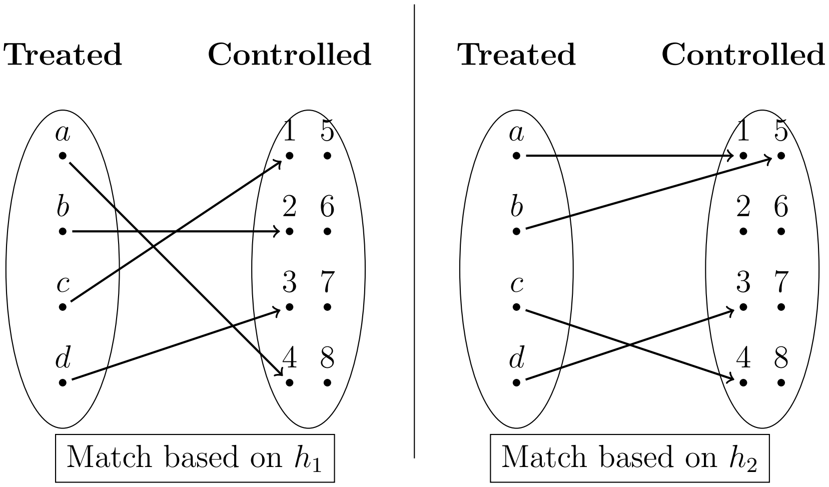


Figure 2.4: Rosenbaum sensitivity analysis as readjustment of matching

Firstly, assume there is no confounder at all, so the treatment assignment of Z is only based on covariates X. We can calculate the following ratio $\rho_{i,j}$:

$$\rho_{i,j} = \frac{\pi_i(1-\pi_j)}{(1-\pi_i)\pi_j} \tag{2.16}$$

in which,

$$\pi_k = Pr(z_k = 1|X = x_k) = \sum_u Pr(z_k = 1|X = x_k, U = u)Pr(U = u) \tag{2.17}$$

So $\rho_{i,j}$ is actually the ratio of two individual i and j's odds ratios of being treated given their covariates. Besides, in a simple randomized experiment[2], this ratio should be exactly 1.0, since every individual is equally likely to be assigned with treatment.

Now, consider the value of $\rho_{i,j}$ with individuals i and j and an extra condition that i and j share the same covariate values $x_i = x_j$, it is obvious that $\rho_{i,j}$ is now 1.0, since i and j have the same odds ratio of being treated given their covariates:

$$\rho_{i,j} = \frac{\pi_i(1-\pi_j)}{(1-\pi_i)\pi_j} = 1.0, \forall (i,j) \in \{(a,b) : X_a = X_b\}$$

[2] In this case, it refers experiments with the setting that individuals are randomly selected, according to a uniform distribution, to be treated.

At the same time, if we calculate the same ratio between any matched pair of individuals i and j, the statement of $\rho_{i,j}$ being 1.0 still holds true, for the case of matching, matched pair (i, j) might not share the exact covariates, but they do share the same conditional probability of being treated give their covariates. Though the condition is a bit relaxed, the ratio will not change, because the ratio's value directly relies on these conditional probabilities:

$$\rho_{i,j} = \frac{\pi_i(1-\pi_j)}{(1-\pi_i)\pi_j} = 1.0, \forall(i,j) \in \{(a,b) : \pi_a = \pi_b\} \tag{2.18}$$

The equation above further demonstrates the goal of matching is to mimic a smaller scale randomized experiment in each matched group, by matching individuals with different treatment assignments based on their conditional probability of being treated given their covariates. Note that the conditional probability π_i in the equation above is calculated from a propensity score model based on covariates x_i, and that the statement above only holds true when there is no other hidden covariate not captured by x_i.

Now, if we introduce a confounder U into our setting, which means though all the matching procedures are done without U, this confounder does influence the conditional probability of being treated given X and U, and it does influence the outcome R. So if we use $\pi_k^{x,u}$ to denote the true conditional probability of individual k being treated given covariates x_k and u_k, we have:

$$\pi_k^{x,u} = Pr(z_k = 1 | X = x_k, U = u_k) \tag{2.19}$$

Note the fact in our previous setting of no confounder, $\pi_k^{x,u} = \pi_k$. However with confounder u_k, now $\pi_k^{x,u} \neq \pi_k$. Due to this change, we also need to update the previous definition of ratio $\rho_{i,j}$ 2.18:

$$\rho_{i,j} = \frac{\pi_i^{x,u}(1-\pi_j^{x,u})}{(1-\pi_i^{x,u})\pi_j^{x,u}}$$

for which $\rho_{i,j}$ no longer relies on π_i and π_j, but now $\pi_i^{x,u}$ and $\pi_j^{x,u}$.

Notice that this ratio is always calculated ***after*** matching, and during the matching process we do not know any information of U, even its existence. The matching model averaging over U is still solely based on X; in another words, the matching model is still trying to generate matched groups (i, j) with individuals i and j having different treatments, but the same $\pi_i = \pi_j$.

Now we calculate the ratio between matched pair individuals again, and not to our surprise, it is no longer guaranteed to be 1.0:

$$\rho_{i,j} = \frac{\pi_i^{x,u}(1 - \pi_j^{x,u})}{(1 - \pi_i^{x,u})\pi_j^{x,u}}, \forall(i,j) \in \{(a,b) : \pi_a = \pi_b\} \tag{2.20}$$

Let us assume that if we randomly pick matched individuals i and j to record their ratio $\rho_{i,j}$, its value will be within certain boundaries:

$$\frac{1}{\Gamma} \leq \rho_{i,j} \leq \Gamma, \forall(i,j) \in \{(a,b) : \pi_a = \pi_b\} \tag{2.21}$$

How much this boundary departs from 1.0 reflects how much bias the confounder U brings into the matching performance. As the influence of confounder U on treatment assignment Z is increased, the ratio will be further from 1.0, hence, Γ will become larger.

This ratio is an essential component in Rosenbaum Sensitivity Analysis, and its boundary Γ can then be seen as a boundary of the degree of departure from random assignment of treatment, which is what a reasonable matching process is trying to achieve by matching individuals with similar covariates but different treatments. With hidden confounders U, two matched individuals with exactly the same observed covariates X may still differ in the odds of receiving the treatment, one of these individuals true odds of being treated can be at most Γ times larger than this individuals matched pair.

Instead of trying to calculate the boundary Γ of this ratio, Rosenbaum Sensitivity Analysis attempts to test how much the original matching's performance (e.g. the estimated ATT and its associated p-value) will change, if the matching individuals are changed and swapped within a specific boundary of Γ, which now serves as a parameter to describe the level of departure from a randomized setting [19].

2.4.3 Dual Sensitivity Analysis

Rosenbaum Sensitivity Analysis has investigated the potential imbalance or association between the treatment assignment Z and the confounder U, and has been widely applied in many fields to evaluate the robustness of matching.

However, Rosenbaum Sensitivity Analysis only focuses on using Γ to directly evaluate and analyze treatment assignment Z's departure from a randomized setting after introducing U. Recall the previous Bayes network 2.3, the confounder U has direct causal effects on both treatment assignment Z and outcome R, though swapping individuals in the original matched pairs with certain level of randomness does introduce uncertainty when comparing their outcomes, but essentially Rosenbaum Sensitivity Analysis does not include a straightforward way to evaluate how much randomness the confounder U brings to outcome R.

As an extension of the original Rosenbaum Sensitivity Analysis, Dual Sensitivity Analysis [5] includes of the confounder U's effects on both the treatment Z and outcome R simultaneously, by controlling different levels of uncertainty in these two effects. Dual Sensitivity Analysis can be regarded as a more comprehensive analysis on matching's sensitivity to the potential unobserved covariate [5]. With exactly the same settings of Γ in Rosenbaum Sensitivity Analysis, Dual Sensitivity Analysis evaluates the randomness U introduces to R by asking a straightforward question: given treatment and covariates, how inbalanced will the outcome distribution become after introducing U? For simplicity, we assume the treatment Z, and outcome R are all binary.

For the scenario without any confounder at all, the outcome R is only determined by covariates X and treatment Z as illustrated in Figure 2.2. Similar to the previous approach, for two individuals i and j, let us calculate the following ratio:

$$\rho'_{i,j} = \frac{\eta_i(1-\eta_j)}{(1-\eta_i)\eta_j} \tag{2.22}$$

in which

$$\eta_k = Pr(r_k = 1|X = x_k, Z = z_k). \tag{2.23}$$

So this ρ' essentially represents the odds ratio of outcomes between two individuals given their covariates X and treatment assignments Z. Though the ratio ρ' here does not have any direct links to randomized experiment as ρ in Equation 2.16, in the following we show it to be very helpful when analyzing the confounder's effect on outcomes.

For two individuals i and j having the identical covariates X and treatment Z, it is intuitive that their expected outcomes are identical, hence $\rho'_{i,j}$ is 1.0, as long as our assumption that outcome R is fully determined by X and R holds true:

$$\rho'_{i,j} = \frac{\eta_i(1-\eta_j)}{(1-\eta_i)\eta_j} = 1.0, \forall (i,j) \in \{(a,b) : X_a = X_b, Z_a = Z_b\}.$$

Now let us include confounder U into our analysis. Note that besides Z and X, U is also influencing outcome R. So if we use P'_k to denote the true conditional probability of individual k having outcome as 1 given covariates x_k, treatment z_k and confounder u_k, we have:

$$P'_k = Pr(r_k = 1 | X = x_k, Z = z_k, U = u_k). \tag{2.24}$$

Because, $P'_k \neq \eta_k$ with confounder U, we need to update our definition of $\rho'_{i,j}$ in 2.22:

$$\rho'_{i,j} = \frac{P'_i(1 - P'_j)}{(1 - P'_i)P'_j}. \tag{2.25}$$

Now this ratio ρ' relies on the true conditional probabilities P' instead of η.

To estimate the treatment effect, the ideal method is to compare outcomes from two individuals with identical covariates but different treatments. This is based on the Strongly Ignorable Assumption that individual's two potential outcomes r_0 and r_1 are conditionally independent of treatment Z given covariates X. In this case, it is reasonable to say the outcome difference between individuals having the same covariates is a result of different treatment assignments. However, the existence of the unmeasured confounder U breaks this Strongly Ignorable Assumption, that is, potential outcomes are no longer independent of treatment Z given covariates X, moreover, their values now also have dependencies on U.

Given two individuals i and j having identical covariates, their outcomes differences are not only due to different treatment assignments. Even if two individuals have identical covariates ***and*** identical treatment, their outcomes may still vary within certain boundaries:

$$\frac{1}{\Delta} \leq \rho'_{i,j} \leq \Delta, \forall (i,j) \in \{(a,b) : X_a = X_b, Z_a = Z_b\} \tag{2.26}$$

To further explain the boundaries above: the ratio $\rho'_{i,j}$ departs from being 1.0 only in cases that $U_i \neq U_j$, and how much it will depart is bounded by Δ, for example, Δ being 2.0 means the odds of getting a positive outcome can be doubled if we only change the value of U. A Δ being 1.0 simply says the odds of getting positive outcome will always stay the same no matter how U changes, in another words, given X and Z, the unobserved covariate U would introduce no imbalance to R at all.

2.4.4 Using Logistic Regression Propensity Score Models

Rosenbaum Sensitivity Analysis is widely used to evaluate a matching's performance in scenarios where treatments are supposed to be assigned from a model which can be approximated by a logistic regression model taking covariates X and confounder U as inputs. As will be shown in this section, the main reason for choosing Rosenbaum Sensitivity Analysis to evaluate the reliability of matching with logistic regression models relies heavily on the treatment model's log-linear structure, with which the degree of departure parameter Γ and confounder U will have a more direct link with each other. With this advantage, the specific values of U are *no longer necessary* for analyzing a model's sensitivity. Instead, one can set different levels of effect from U to Z by directly setting different values of Γ.

Assume a regression model is learned from data to predict the treatment assignment, in which the covariates X and the confounder U are independent from each other and there is no interaction between them (See Figure 2.3), as appeared in the following expression:

$$\log(\frac{P_i}{1-P_i}) = \kappa(x_i) + \gamma u_i. \tag{2.27}$$

Here, P_i is the conditional probability of being treated given covariates X and U as in Equation 2.19, κ is a function taking a vector of X as inputs, and γ is a factor denoting the confounder U's effect on treatment assignment.

Then we have the odds ratio of i being treated as:

$$\frac{P_i}{1-P_i} = \exp(\kappa(x_i) + \gamma u_i)$$

For two perfectly matched individuals i and j, we have $x_i = x_j$, such that we can rewrite the ratio of $\rho_{i,j}$ between these two individuals as:

$$\begin{aligned} \rho_{i,j} = \frac{P_i(1-P_j)}{(1-P_i)P_j} &= \exp(\kappa(x_i) - \kappa(x_j) + \gamma u_i - \gamma u_j) \\ &= \exp\left(\gamma(u_i - u_j)\right) \end{aligned} \tag{2.28}$$

When U's constraint has been set as binary $u_i = \{0, 1\}, \forall i$, then $\rho_{i,j}$ can be further simplified as

$$\rho_{i,j} = \begin{cases} 1 & \text{if } u_i = u_j \\ \exp(\gamma) & \text{if } u_i > u_j \\ \frac{1}{\exp(\gamma)} & \text{if } u_i < u_j \end{cases}$$

and we can directly map the degree of departure parameter Γ to the confounder effect γ as:

$$\Gamma = \exp(\gamma)$$

This allows us to avoid having to produce specific values of the confounder U when conducting experiments for sensitivity analysis, but this is only possible if a generalized linear relation is assumed between the treatment odds to X, as well as the treatment odds to U, while there is no interaction between X and U. Subsequently, we will see that when using decision trees for analysis, we cannot use this "trick."

2.5 Decision Trees

Decision trees are predictive models constructed by supervised machine learning based on the labeled training data. Decision tree learning algorithms work by finding the most informative covariate, partitioning the data based on the covariate, then recursively processing each partition. After the training process, covariates selected to split the data can be represented as a tree-structure model, hence the term "decision tree". Decision tree models are widely used due to good predictive performance, as well as ease of implementation and interpretation. Decision trees can be used for both regression analysis, whose predicted outcome is more often a real value number, and classification analysis, whose predicted outcome is the class that data belongs to. Here we introduce the classification decision tree model.

2.5.1 Decision Tree Representation

The learned decision tree model consists of a group of ⟨*covariate*, *partitions*⟩ pairs arranged in the structure of a tree; in each ⟨*covariate*, *partitions*⟩ pair the *covariate* is represented as a "tree node" and each *partition* from *partitions* is represented as a "branch" or a directed edge from the covariate to the child node.

Each tree node can have a number of "branches" that link to other child tree nodes, a "branch" is normally defined as a constraint of the node covariate, such that instances satisfying this branch's constraint will be sent to the child node the branch links to. When a tree node has more than one branch, we say the data instances *split* on this tree node, while a tree node with no branches is a "leaf" node. Instead of further splitting data instances, a leaf node returns the final prediction. If a tree node has no incoming edges or branches, it is the root node of the tree.

With a learned decision tree model, an instance is classified by starting at the root node of the tree, testing the covariate specified by this node [11], then moving down the tree branch whose corresponding constraint is satisfied by the value of covariate. This process is then repeated for the subtree rooted at the new node, until it reaches a leaf node.

In 2.5, we show a typical learned decision tree [11] which classifies Saturday mornings according to whether they are suitable for playing tennis. The node at the top with covariate *Outlook* and no incoming branches is the root node. Nodes with rounded corners without any child node are the leaf nodes; each leaf node stores the final predictions. Other nodes are all splitting nodes.

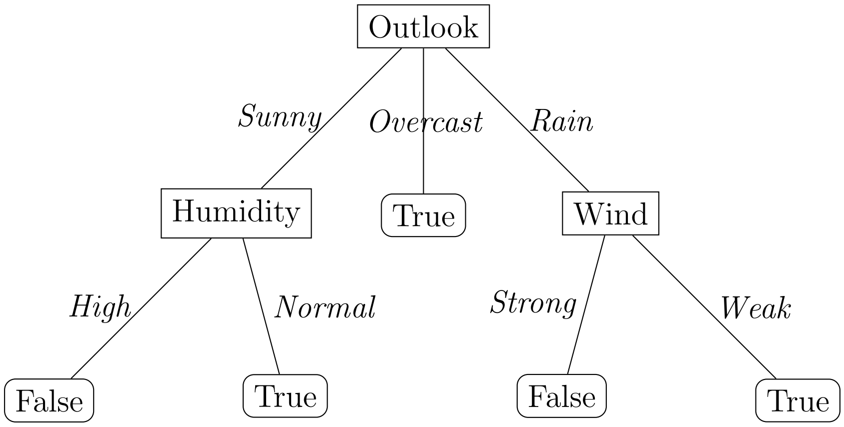


Figure 2.5: A decision tree to classify if a day is suitable for playing tennis

2.5.2 Decision Tree Learning Algorithm

The process of generating tree nodes and tree branches to develop a learned decision tree requires a decision tree learning algorithm, which is essentially a searching algorithm in the space of all possible decision trees that can be built from the data. In general, most decision tree learning algorithms are variations on a core algorithm that employs a top-down, greedy search [11], that is, gradually "growing" a decision tree from the root node, generating child nodes such that each node's covariate and constraints could "mostly" split all local instances in this node into groups with homogeneous predictions. In this section, we review one basic decision tree learning algorithm, the ID3 algorithm [14].

Covariate Selection

The core idea of the ID3 algorithm is to answer the following question: which covariate and constraint should we choose to generate the next node that is most useful for classification? The ID3 algorithm answers this question with the help of information gain: recursively generate a new node with the most informative covariate and constraints that maximize the information gain, which is the amount of entropy decrease after data are partitioned by the new node. The definition of information gain is based on the concept of entropy, which in information theory measures the impurity of the label within an instance collection. One way to understand this is to connect entropy with bit representation of each instance, that is, entropy can be regarded as the averaged minimal number of bits/digits we need to represent all instance labels in a group:

We can easily distinguish two different types of instances by labeling each of them either 0 and 1, which will take only 1 bit. So with n bits, we can distinguish at maximal 2^n different instances labels. Given an instance collection with k different labels, we can easily calculate the minimal bits to distinguish all of them, which is $\log_2 k$.

However, this minimal number of bits can only reflect how many different labels we have in an instance collection, it does not reflect any information regarding the proportion of each label in the collection, while one might argue that by changing the proportion of each label, the "impurity" of collection will also change.

To include the information regarding proportion of each instance labels in the impurity measurement, we can take the use of each unique instance type's probability in a group. Assume in our instance collection S, we have n different instance labels, and the proportion of instance label i in S denoted as p_i, we have the definition of entropy in collection S:

$$Entropy(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{2.29}$$

while,

$$\sum_{i=1}^{n} p_i = 1 \tag{2.30}$$

if we only have two labels in the group: positive label with proportion of $p_\oplus$ and negative label with proportion of $p_\ominus$, then the group entropy can be represented as

$$Entropy(S) = -p_\oplus \times \log_2 p_\oplus - p_\ominus \times \log_2 p_\ominus \tag{2.31}$$

in which

$$p_\oplus + p_\ominus = 1.0 \tag{2.32}$$

such that entropy is 0.0 if all instances in the group have the same label, and for boolean types the entropy reaches the maximum of 1.0 if this group is equally mixed with two different types; hence, higher the entropy, less the purity.

Information gain captures the drop in entropy after processing instances through a node. Given a splitting node *Node A* with two partitions $Partition_1$ and $Partition_2$ as in Figure 2.6, all instances in *Node A* satisfying the condition of $Partition_1$ fall into *Node B*, others in *Node C*. Assume there are in total N_A instances in *Node A*, N_B instances in *Node B* and N_C instances in *Node C*, then we have $N_B + N_c = N_A$.

As we have shown, the entropy of all instances in *Node A* before splitting is defined as:

$$Entropy(A) = -\sum_{i \in n} p_i \log_2 p_i \tag{2.33}$$

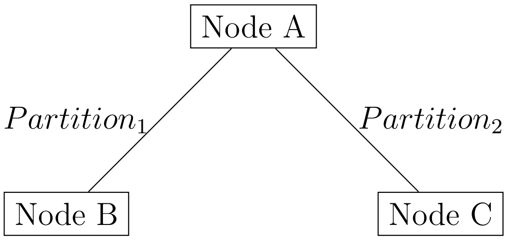


Figure 2.6: Splitting node with binary constraint

similar to *Node A*, we can also get *Entropy(B)* and *Entropy(C)* after the split, the information gain after this splitting node is formally defined as:

$$IG(A) = Entropy(A) - \frac{N_B}{N_A}Entropy(B) - \frac{N_C}{N_A}Entropy(C) \tag{2.34}$$

As we can see, with a larger information gain, the two groups of instances will get "purer" after the split, hence, the more useful this split is to our classification, which can be regarded as a series of splits aiming to generate subgroups with minimal group entropy.

Thus, when generating a new split node, ID3 tries all covariates with constraints and calculate the potential information gain for each covariate, then picks the ⟨*covariate*, *partitions*⟩ pair with the highest information gain as a new node, then recursively generate nodes for each child node.

As a greedy, top-down approach for generating new splitting nodes, one may ask when to stop growing more nodes. To solve this problem, different stopping criteria can be used, such as stopping when there is no covariate to use, or stopping when the number of instances is below a threshold. When the algorithm stops generating a child node for a branch, the current node becomes a leaf node in the tree. A leaf node returns the majority of the class among all instances in it as the prediction.

2.5.3 Tree Performance

After a decision tree is learned, we need to evaluate its performance; that is, if the learned tree can classify new instances correctly, which refers to the predictive power of the tree. And it would be helpful to qualify a learned tree by estimating its predictive power before actually putting it in usage.

Predictive Power and Cross Validation

Estimating the predictive power before new instances are obtained means we need to use the same data for both training and testing. However, we should never use the same instances for both training and testing purposes, since decision tree learning algorithm takes a greedy approach, the learned tree tends to maximize its predictive performances on instances used to train it. In another word, the learned decision tree's predictive power is biased to the data set it learns from; hence, using the training data for testing provides an overestimation of its predictive power.

One way to properly estimate the learned tree's predictive power is cross validation. The idea is to split the whole data into two sets: training set and testing set. Use only the training set data when growing a decision tree; after learning a tree, use only the testing set data to test the learned tree's predictive power. The estimated predictive power, such as learned tree's prediction error rate on testing set, is more trustworthy as long as there is no intersection between training and testing sets.

As a variation of cross validation, *k*-fold cross validation provides a more predictive power estimation by avoiding the bias in training and testing sets: instead of splitting the whole data into two sets, it equally and randomly split the whole data into *k* folds, namely, $fold_1$ to $fold_k$. Then it iterates through all k folds, in every iteration i it keeps $fold_i$ as the testing set, learns a new tree $Tree_i$ based on other $k-1$ folds, uses $fold_i$ to test the learned $Tree_i$, keeps record of the predictive power metric, such as error-rate, of $Tree_i$. After all k iterations, it calculates the average of all k predictive power metrics as the estimated predictive power of the decision tree learned from the whole data.

Pruning

One potential problem with the greedy decision tree learning algorithm we mentioned before is overfitting; that is, the learned decision tree often overfits the training data, such that it only performs the best with instances from the training data, but performs worse with other new instances. This happens for several reasons: one, there can be noise in the training data or the training data is biased, which causes the greedy algorithm to pick less optimal covariates while growing the tree; another reason can be improper selection of the threshold in the stopping criterion, thus each node near the bottom of the tree does not have enough instances to grow child nodes that improve the overall tree performance.

One way to deal with the potential overfitting problem is to use a validation set for reduced-error pruning. Before learning a decision tree from the training data set, we first split the training set into two separate groups: the new training set and the validation set. There should be no intersection between these two sets. The training set, as it literally means, is for training purposes only, while the validation set is used to prune the decision tree learned from the training set. Reduced-error pruning assumes that each node in the tree can be regarded as a potential node to be pruned. Starting from the leaf nodes in a bottom-up approach, each node i will be tested in two cases: The first case is to keep the node as it is, no matter if it is a leaf node or a splitting node, calculate the estimated classification error rate $e_{root}(i)$ of this tree instance using validation set. The second case is to remove the subtree rooted at this node, then making this node, which was originally a splitting node, a new leaf node by returning the classification result as the most common classification among all instances in this node, calculate the classification error rate $e_{leaf}(i)$ of this modified decision tree instance using validation set. We then compare the decision tree's prediction accuracy on validation set and modify the tree by only iteratively removing the node such that its removal will increase decision tree's predictive power the most, which is the node i that maximizes the value of $e_{leaf}(i) - e_{root}(i)$. This pruning process stops when any node removal will only decrease decision tree's predictive power, that is, $e_{leaf}(i) - e_{root}(i) < 0.0, \forall i$.

2.5.4 NLS-Decision Tree

The pruning approach above solves the problem of overfitting in greedy decision tree learning algorithms; however, this only improves the estimated predictive power of the pruned decision tree. Considering the fact that our final learned decision tree should be easily interpreted and used by clinicians in all situations even without the help of a computer, we do need to maintain a stable and meaningful structure of the tree. In cases of small amount of training data, the ID3 algorithm might not be the ideal approach for decision tree generation, for it produces a tree that maximizes predictive power on the training data, but the resulting tree may not be stable to small changes in the data.

The stability of a decision tree often refers to if the decision tree maintain its original structure when noise is introduced in the training data. If the decision tree keeps its original splitting nodes and edges connecting these nodes, then the decision tree is regarded as stable. The Node-Level-Stabilized learning algorithm (NLS-DT) designed by Dannegger [4] attempts to generate a decision tree based on predictive power while maintaining a simple and stable tree structure at the same time. In this section, we will introduce the general concepts of the NLS-DT algorithm.

Algorithm

The NLS-DT algorithm generates a binary classification decision tree with a more stable structure than other algorithms such as ID3. The main concept of NLS-DT is to introduce bootstrap replicates of data when choosing covariates as splitting nodes. As shown in Algorithm 1, when generating a new node using NLS-DT, instead of directly examining all potential covariates with their informative metrics such as information gain, it uses n bootstrap replicates of data at the current node to construct a node with a high predictive power that also maintains the node-level stabilization at the same time. For each node-level bootstrap replicate, the algorithm separately chooses the most informative covariate and constraint, then it generates the splitting node with covariate that voted by the most number of bootstrap replicates, and taking the median of constraint values from all bootstrap replicates voting for this covariate as the final constraint value of this new node.

```
Data: Labeled training data
Result: NLS-Decision Tree
Initialization of root node;
for every leaf node $t$ containing $N_t$ instances do
    for $n$ = 1 to $N$ do
        Generate bootstrap replicate $N_t(n)$ from $N_t$;
        Choose the most informative $\langle covariate_n, constraint_n \rangle$ from $N_t(n)$;
        Include $\langle covariate_n, constraint_n \rangle$ into $Table_t$;
    end
    Set $covariate(T)$ as the most frequent $covariate$ in $Table_t$;
    Include $Table_t$ entries with $covariate_i == covariate(T)$ into $Table'_t$;
    Set $constraint(T)$ as the median of $constraint$ in $Table'_t$;
    Update current node as $\langle covariate(T), constraint(T) \rangle$ ;
    Generate two child leaf node of $t$ with instances in $N_t$ split by $constraint(T)$;
    Reset $Table_t$ and $Table'_t$;
end
```

Algorithm 1: NLS-Decision Tree learning algorithm

Pruning

After a NLS-Decision Tree is learned from training set, we also need to prune the tree to avoid overfitting. As proved by Briand [3], the reduced-error pruning approach we introduced before performs well for NLS-Decision tree compared with other pruning method such as cross-validation pruning.

Chapter 3

The Treatment Effect Tree (TET) and its Causal Sensitivity Analysis

In this chapter, we introduce our new method to estimate the treatment effect on each individual with the help of a decision tree learned from matched pairs, and we propose our method to conduct causal sensitivity analysis for this decision tree model, as an extension of the classic Rosenbaum Sensitivity Analysis and Dual Sensitivity Analysis.

First, we discuss the ventilator triage problem and categorize all similar problems that require estimation of individual treatment effects. We argue that estimating individual treatment effects can be achieved by estimating the treatment effect of sub-populations grouped by similar covariates that influence outcome, and that this is a necessary first step to solve such problems.

We then introduce a new concept which we call the Treatment Effect Tree (TET), which is a decision tree that can be used to achieve the estimation of individual treatment effect. We also present a guide for how to learn and estimate the TET based on individuals through matching on potential outcomes.

Having introduced the idea of the TET, we then propose our empirical end-to-end sensitivity analysis for TET estimation. This can be regarded as extending the original Rosenbaum and Dual sensitivity analysis from logistic regression models into a more general group of models, including decision trees.

3.1 Individual Treatment Effect Estimation Problems

As we mentioned in Section 1.1, we attempt to solve the ventilator triage problem by prioritizing patients according to the effect of mechanical ventilation on those patients, and this requires estimating the Individual Treatment Effect, which is one of the main tasks in our work. We use the term of Individual Treatment Effect (ITE) to refer to the treatment's effect on each individual:

$$\tau_i = r_i^1 - r_i^0 \tag{3.1}$$

in which, as before, r_i^z stands for the potential outcome of individual i if they had treatment assignment z.

Given the data collected from an observational study, estimating the Average Treatment effect on the Treated (ATT) can be achieved by performing propensity score matching on the data and compare the average outcomes within the treated and control groups as discussed in Section 2.3.3. However, the ATT only evaluates the treatment's effect averaged across all treated individuals; it does not provide any information helping us to understand which subgroup of individuals would benefit the most from treatment, given the assumption that treatment might have different levels of effect on subgroups of individuals with different covariates. In fact, the assumption above is often considered as true, that is, treatment does not have a "uniform" effect for all individuals. For example, a treatment specifically designed to cure a disease might be more effective to patients with that disease, and a marketing action such as price reduction might be more effective to customers essentially care more about the price.

Instead of estimating the averaged treatment effect across treated individuals, estimating the treatment effect on each individual can provide valuable information helping us to have a thorough understanding of the treatment, and designing a better strategy of assigning treatments in the future. Though we use a specific method to estimate the ITE, we believe there are many other methods for estimation. To be more precise, in this work we focus on estimating the ITE for each treated individuals, hence, we estimated the Individual Treatment Effect on the Treated (ITT). For the convenience of notation, we use the term ITE in following sections to refer to the estimand, and the estimated ITT as the estimation of ITE.

With the idea of ITE given, the ventilator triage problem can be regarded as one example of a category of problems; here we categorize such problems as ITE Estimation Problems, for they all require a valid estimation of a boolean treatment's effect on each individuals from observational study. We call a problem an ITE estimation problem if it satisfies the following conditions:

1. Each individual i can be described by covariates x_i.
2. The treatment Z is a boolean value, and the treatment z_i of individual i is fully determined or strongly influenced[1] by covariates x_i.
3. Individual's outcome R after treatment is observed and recorded during the study. For individual i, the corresponding outcome r_i is believed to be fully-determined or strongly influenced by the covariates x_i and treatment z_i.
4. The goal is to estimate and predict the ITE for each individual i based on covariates x_i.

For the specific case of our ventilator triage problem, the goal is to provide a model taking covariates of a new patient that can predict the treatment's effect on this patient. In the following section, we propose our method to estimate ITE with data collected from observational studies.

3.2 Treatment Effect Tree (TET)

The concept of Treatment Effect Tree (TET) is motivated by the ventilator triage problem: clinicians want an easy-to-apply model predicting the treatment effect of ventilator on a new patient given the patient's covariates, and this model should be applicable offline without the help of any computer after the model is learned. The rationale for this is that the model is intended for use in potential disaster scenarios, where access to a computer may not be possible. This motivation naturally leads us to the idea of a decision tree model predicting treatment effect for each individual patient, for its ease of implementation and for the intuitive structure that can be easily understood and applied by clinicians without computer science background.

The rationale for building the Treatment Effect Tree is that treatment is supposed to have different effect on individuals described by different covariates, thus individuals can be grouped according to their covariates and the treatment should have similar amount of effect within the same group. This process can be represented by a tree model, that is, if we use a decision tree to achieve the grouping process based on covariates, then individuals falling into the same leaf node are expected to have a similar treatment effect.

[1] Considering the cases with potential confounder U, z_i is determined by x_i and u_i together

Given the definition of ITE in Equation 3.1, the real TET can be acquired by predicting the τ_i via tree nodes that split on covariates x_i. This can be achieved with decision tree learning algorithms, if two potential outcomes are accessible for each individual at the same time. However, as we discussed, this is not practical because for individual i we can only observe one of the two potential outcomes, either the outcome with treatment r_i^1 or the outcome with control r_i^0, but never both. So the real ITE data are not available; instead, we can only try to estimate the real TET by estimating counterfactuals. Then the problem now becomes what information shall we feed the decision tree learning algorithm to estimate the real TET, or, how shall we acquire the ITE from the data collected from an observational study.

A TET may be constructed for different populations of patients; in this thesis we focus on building a TET specifically for the population of treated patients, analogous to estimating the ATT. A similar strategy could be used to build a TET on the entire population, or on a different sub-population.

3.2.1 Estimating the TET using matching

Matching is designed to handle the problem of only one outcome being observed, and considering its applications, matching has been used to estimate the ATE and ATT well. Naturally, we follow the idea of matching to estimate the real TET, that is, instead of using two potential outcomes r_i^0 and r_i^1 of the same individual i, we use the actual outcome r_i that has been observed and recorded in the data, combined with another actual outcome r_j from another individual j that is matched with i, we make sure that $z_i \neq z_j$, such that the differences between r_i and r_j can be regarded as a rough estimation of ITE, given a reliable matching approach.

Propensity Score Matching

As discussed above, one of the most widely used matching methods when estimating ATE or ATT is to match individuals with different treatment assignments according to the propensity score, which is the conditional probability of being treated given covariates. However, though propensity score matching performs very well when estimating the ATE or ATT, it will not succeed when the goal is to estimate the TET: matching individuals with similar propensity scores does *not* guarantee that it is reasonable to directly compare their outcomes and using the outcome difference as the estimated ITE; in fact, the matched results are only useful if we later plan to average outcomes across all matched pairs. Given the definition that ITE refers to the treatment effect defined by covariates, we cannot simply average across the whole population, as that is equivalent to assuming that the ITE is uniform over all individuals and independent of covariates.

To further demonstrate that matching with propensity score and use matched pairs to train a decision tree does not result in an valid estimation of TET, we conduct an experiment in Section 4.4.1 showing that the decision tree learned from propensity score matching is totally different from the real TET. However, in the experiment we also point out that propensity score matching does guarantee to perform well when estimating treatment effect after averaging, either the ATE or the ATT.

Potential Outcome Matching

Now, reconsider the definition of ITE in Equation 3.1, it is true that we can not observe the two potential outcomes at the same time, thus for individual i we can only use the actual outcome r_i that is observed and recorded in the data. By matching, we hope to have the comparison between the outcome r_i and the matched outcome r_j able to provide us more information about the treatment effect to individuals similar to i. Propensity score defines the "similarity" by the conditional probability of being treated given covariates, and we showed above that using the propensity score cannot provide the information we need for estimating the TET because individuals with the same propensity score do not essentially share the same ITE.

In order to regroup individuals such that treatment effect is more consistent and uniform for individuals in the same group, we propose to match based on the potential outcome, such that individuals with different treatment assignment but the similar potential outcomes get matched together. Here we choose to match on the potential outcome without treatment, since we intend to use the covariates from treated individuals to estimate the real TET on the treated population, so the individual i and j with different treatment should be matched with each other if they share the similar potential outcome without treatment. Hence,

$$\hat{\tau}_i = r_i^1 - r_j^0 \tag{3.2}$$

in which,

$$z_i = 1 \text{ and } z_j = 0 \text{ and } |\hat{r}_i^0 - \hat{r}_j^0| < \delta \tag{3.3}$$

There are many different methods to get $\hat{r}_i^0$, however, we choose to use the decision tree model to estimate the potential outcome without treatment, this decision tree can be learned from using the control group of individuals to predict their outcomes based on their covariates. The reason for choosing potential outcome without treatment to match on is based on the definition of ITE in Equation 3.1, that is, if two individuals i and j have the same two potential outcomes $r_i^0 = r_j^0$ and $r_i^1 = r_j^1$, their ITEs are the same. Based on this idea, we can match on individuals with the similar estimated outcomes instead of the actual outcomes.

To further show that matched pairs from estimated outcome matching can be used to estimate the real TET, we conduct an experiment in 4.4.2 and show the real TET as well as estimated TET using estimated outcome matching by decision tree and by regression. We show that the estimated TET based on estimated outcome matching with decision tree outcome estimator is almost the same as the real TET, proving that estimated outcome matching can perform well in estimating the real TET from observational data.

3.3 Sensitivity Analysis for TET

For TET, there is still a matching process used during its estimation and learning. As we know, the quality of propensity score matching is always vulnerable to unmeasured confounding, and for this reason Rosenbaum and Dual Sensitivity Analysis were derived to evaluate matching's robustness in situations with different levels of unmeasured confounding. Given our approach of estimated outcome matching for TET estimation, a similar question which is reasonable to ask would be: how robust the estimated TET will be in situations with different levels of confounding? Answering this question is more complex for TET estimation than for estimating ATE from propensity score matching, since our final model gives more a complex output than regular matching, that is, instead of producing a scalar from the matching results, such as the estimated ATE, we use the matched results and learn another model based on the matched results, which is the estimated TET.

Confounding might influence the results of estimated TET at two different stages: First, the real TET with confounding might differ from the real TET without confounding, and if the real TET structure is vulnerable to a very weak confounding, then TET might not be very valuable after all; Second, the estimated TET with confouding might also differ from the estimated TET without confounding, and this information can help us evaluate whether the estimation process is reliable or not. As a result, our sensitivity analysis must be able to analyze the robustness of our matching-based TET model, and by using the term "robustness" we refer to how stable both the real TET and estimated TET structures would be given different levels of confounding controlled by Rosenbaum and Dual Sensitivity Analysis, that is, given confounders U with levels of effect on treatment assignment Z and outcome R.

However, it is not possible to conduct Rosenbaum sensitivity analysis or Dual sensitivity analysis on decision trees directly, due to the fact that both of these analyses determine the *effect* of an unmeasured confounding variable on a generalized linear model without actually constructing it. To study decision tree sensitivity to unmeasured confounders we will explicitly generate confounder observations u_i for each individual, based on the Γ and Δ parameters which describe the relationship between the confounder, treatment, and outcome. Then we compare the structures of the TET learned without the hypothesized confounder with the TET learned with the hypothesized confounder. Figure 3.1 shows a diagram to illustrate the procedures of conducting sensitivity analysis on an estimated TET.

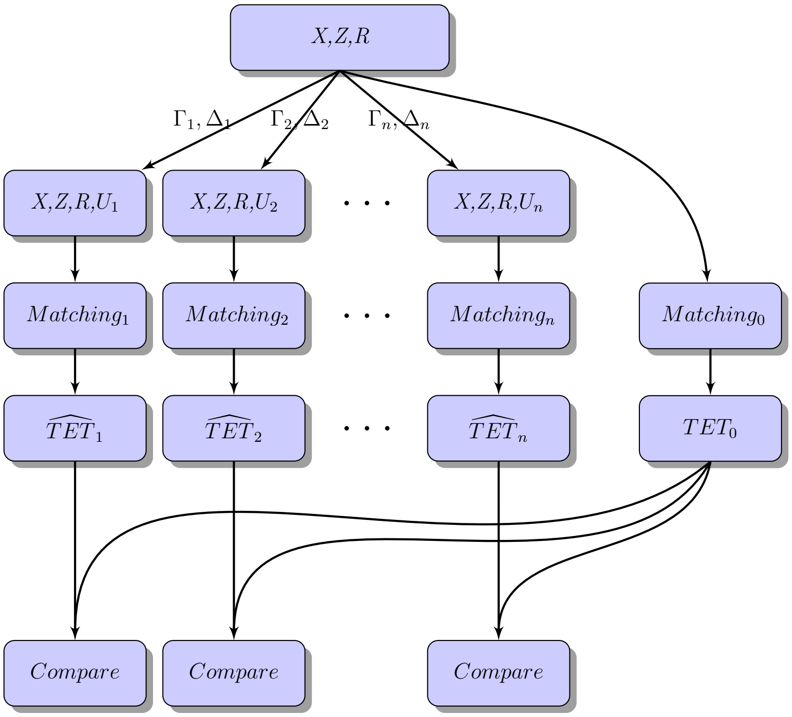


Figure 3.1: Diagram of estimating and comparing TET with different levels of Rosenbaum confounding for TET sensitivity analysis

3.3.1 Generating the Confounder U

In observational data, we are expected to have covariates X, treatment assignment Z, and outcome R recorded, and based on this information we are able to estimate the TET from a potential outcome matching. Now we discuss how a boolean confounder U can be generated for each individual for estimated TET sensitivity analysis.

We generate confounder U such that the distribution of the generated U reflects the relationship between U, Z, and R. This means, for an existing population with covariates X, treatment Z, and outcome R fixed, we need to control the distribution of U in this population such that this confounder U's effect on treatment assignment Z and outcome R can be controlled separately. Recalling Dual Sensitivity Analysis we reviewed in Chapter 2, Γ is used to describe confounder U's effect on treatment Z, and Δ on outcome R; therefore, we generate U based on different values of Γ and Δ. Here we use the term *Rosenbaum confounding* referring to the confounder setting with the same Γ and Δ.

We now show how a boolean confounder U can be generated for each individual in our existing population (i.e. dataset) with covariates X, treatment Z, and binary outcome R by controlling its effects using two parameters Γ and Δ.

This generating process requires the conditional probability,

$$Pr(U = u|X = x, Z = z, R = r). \tag{3.4}$$

For simplicity, in following probability notations we skip the item variables when variable instances are given, so the probability above is written as,

$$Pr(u|x, z, r). \tag{3.5}$$

According to the definition of conditional probability, this probability is equivalent to,

$$\frac{Pr(x, r, z, u)}{Pr(x, r, z)}. \tag{3.6}$$

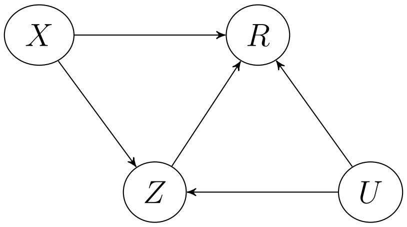
The denominator part in the above fraction is the same as the following marginal probability, where U is marginalized out,

$$Pr(x, r, z) = \sum_{\forall u'} Pr(x, z, r, u'). \tag{3.7}$$

Combining Equation 3.6 and Equation 3.7, we can rewrite Equation 3.5 as

$$Pr(u|x, z, r) = \frac{Pr(x, r, z, u)}{\sum_{\forall u'} Pr(x, z, r, u')}. \tag{3.8}$$

In Equation 3.8, both the numerator and denominator are in the format of a joint probability. Recall that the dependencies among X, Z, R, and U are given by the Bayes network shown in Figure 2.3:



Therefore, the joint probability can be expanded as

$$Pr(x, z, r, u) = Pr(x)Pr(u)Pr(z|x, u)Pr(r|x, z, u). \tag{3.9}$$

Then we expand both the nominator and denominator part of Equation 3.8 based on Equation 3.9 and cancel the common factor $Pr(x)$

$$\begin{aligned} Pr(u|x, z, r) &= \frac{Pr(x, r, z, u)}{\sum_{\forall u'} Pr(x, z, r, u')} \\ &= \frac{Pr(x)Pr(u)Pr(z|x, u)Pr(r|x, z, u)}{\sum_{\forall u'} Pr(x)Pr(u')Pr(z|x, u')Pr(r|x, z, u')} \\ &= \frac{Pr(u)Pr(z|x, u)Pr(r|x, z, u)}{\sum_{\forall u'} Pr(u')Pr(z|x, u')Pr(r|x, z, u')} \end{aligned} \tag{3.10}$$

which gives the conditional probability that $U = 1$ for any individual with given x, z, and r.

Now we separately prove that each probability we need in Equation 3.10 exist and can be fully determined given any choice of Γ and Δ:

$Pr(u)$

These values do not interact with Γ and Δ, so we can set them up with any reasonable values, as long as they sum up to 1.0. For the binary confounder setting, we set $Pr(U = 1) = Pr(U = 0) = 0.5$.

$Pr(z|x, u)$

For binary confounder U and binary treatment Z, this expression refers to four probabilities given a fixed $X = x$; here we denote these as $P_{z,u}$:

$$P_{1,1} = Pr(Z = 1|X = x, U = 1)$$

$$P_{0,1} = Pr(Z = 0|X = x, U = 1) = 1 - P_{1,1}$$

$$P_{1,0} = Pr(Z = 1|X = x, U = 0)$$

$$P_{0,0} = Pr(Z = 0|X = x, U = 0) = 1 - P_{1,0}$$

Recall the definition of Γ in Equation 2.21, if we push the ratio ρ to its limit, which is either Γ or $\frac{1}{\Gamma}$, the confounder U's effect will also be pushed to its limit, while still bounded by Γ. In our case, we set the effect of confounder U on treatment Z to be consistent; that is, with the same covariates X, an individual having $U = 1$ is always more likely to be treated. Therefore we have the following equation relating the above quantities:

$$\frac{P_{1,1}}{1 - P_{1,1}} = \Gamma \times \frac{P_{1,0}}{1 - P_{1,0}}. \tag{3.11}$$

We can also relate the quantities to each other through the propensity score

$$\begin{aligned} Pr(Z = 1|X = x) &= \sum_{\forall u'} Pr(Z = 1|X = x, U = u')Pr(U = u'|X = x) \\ &= P_{1,0}Pr(U = 0|X = x) + P_{1,1}Pr(U = 1|X = x) \\ &= P_{1,0}Pr(U = 0) + P_{1,1}Pr(U = 1). \end{aligned} \tag{3.12}$$

Note that in Equation 3.12, we need the probability $Pr(U = u'|X = x)$ to marginalize out confounder U; recall from Bayes network shown in Figure 2.3 that X and U are independent as long as R and Z are not observed, that is $Pr(U = u'|X = x) = Pr(U = u')$, so we safely replace $Pr(U = u'|X = x)$ as $Pr(U = u')$. Also note in Equation 3.12 $Pr(Z = 1|X = x)$ is essentially the propensity score given $X = x$ used in matching, which we can easily calculate by learning a propensity score model.

Combining Equation 3.11 and Equation 3.12 together, we get the following quadratic equation:

$$(\Gamma - 1)\, \tfrac{Pr(U=1)}{Pr(U=0)} P_{1,1}^2 + \left(\tfrac{Pr(Z=1|X=x)}{Pr(U=0)} - 1 - \left(\tfrac{Pr(Z=1|X=x)}{Pr(U=0)} + \tfrac{Pr(U=1)}{Pr(U=0)} \right) \Gamma \right) P_{1,1} + \tfrac{Pr(Z=1|X=x)}{Pr(U=0)} \Gamma = 0. \tag{3.13}$$

By solving Equation 3.13 we can get all four instances values of $P_{z,u}$ given X for each Γ.

***Pr*(*r*|*x*, *z*, *u*)**

Similarly, for binary outcome R and binary confounder U, we have eight probabilities denoted as $P_{r,z,u}$:

$$\begin{aligned}
P_{1,1,1} &= Pr(R=1|X=x, Z=1, U=1)\\
P_{1,0,1} &= Pr(R=1|X=x, Z=0, U=1)\\
P_{1,1,0} &= Pr(R=1|X=x, Z=1, U=0)\\
P_{1,0,0} &= Pr(R=1|X=x, Z=0, U=0)
\end{aligned}$$

plus their four counterparts, denoted as $P_{0,z,u} = 1 - P_{1,z,u}$. In Dual Sensitivity Analysis we have discussed in Section 2.26, Δ is used to define confounder U's effect on outcome R given covariates X and treatment Z, more specifictly, Δ represents how much the odds of having outcome R as 1 can vary with different confounder U. In our case, similar to the case for Γ, we also set confounder U's effect on outcome R as consistent; that is, given the same covariates $X = x$ and treatment $Z = z$, individuals with $U = 1$ will have a higher odds to get outcome $R = 1$, compared with individuals having $U = 0$. Maximizing this effect gives

$$\frac{P_{1,1,1}}{1-P_{1,1,1}} = \Delta \times \frac{P_{1,1,0}}{1-P_{1,1,0}} \tag{3.14}$$

and

$$\frac{P_{1,0,1}}{1-P_{1,0,1}} = \Delta \times \frac{P_{1,0,0}}{1-P_{1,0,0}}. \tag{3.15}$$

And again, we have additional information, this time from the outcome model,

$$\begin{aligned}
Pr(R=1|X=x, Z=1) &= \sum_{\forall u'} Pr(R=1|X=x, Z=1, U=u')Pr(U=u'|X=x, Z=1)\\
&= P_{1,1,0}Pr(U=0|X=x, Z=1) + P_{1,1,1}Pr(U=1|X=x, Z=1)
\end{aligned} \tag{3.16}$$

and

$$\begin{aligned}
Pr(R=1|X=x, Z=0) &= \sum_{\forall u'} Pr(R=1|X=x, Z=0, U=u')Pr(U=u'|X=x, Z=0)\\
&= P_{1,0,0}Pr(U=0|X=x, Z=0) + P_{1,0,1}Pr(U=1|X=x, Z=0).
\end{aligned} \tag{3.17}$$

The conditional probability $Pr(u|x,z)$ required in Equation 3.16 and Equation 3.17 can also be further expanded

$$\begin{aligned} Pr(u|x,z) &= \frac{Pr(u,x,z)}{Pr(x,z)} \\ &= \frac{Pr(u,x,z)}{Pr(z|x)Pr(x)} \\ &= \frac{Pr(z|x,u)Pr(u|x)Pr(x)}{Pr(z|x)Pr(x)} \\ &= \frac{Pr(z|x,u)Pr(u)}{Pr(z|x)} \end{aligned} \tag{3.18}$$

In Equation 3.18, $Pr(z|x)$ can be calculated from the propensity score given covariates $X = x$, and $Pr(z|x,u)$ is exactly the probability computed by solving 3.11 and 3.12 together as we showed above. Having substituted in their values, we can form two more quadratic equations based on 3.14, 3.15, 3.16 and 3.17 to compute the eight probabilities needed. Solving 3.14, 3.16, 3.17, and 3.18 together, we will have a set of $P_{r,z,u}$ given any value of Δ.

Note that we can estimate the probability $Pr(R = r|X = x, Z = z)$ in Equation 3.16 and Equation 3.17 using any reasonable model taking X and Z as inputs and predicting output $Pr(R = 1|X, Z)$. For example, we could use two decision tree models summarizing outcome R given X for each treatment group. (I.e., we do not need counterfactuals to estimate this.)

Now we have proved each item in 3.10 is accessible given Γ and Δ and estimatable quantities, which means, $Pr(u|x,z,r)$ is also accessible given Γ and Δ, so we can generate U given X, Z, R, Γ, and Δ according to the probability calculated in 3.10.

Given that we are able to generate a hypothesized hidden confounder U based on our data and based on Γ and Δ, now we can perform sensitivity analysis with respect to this confounder. Below, we provide the high-level algorithm as a guideline for TET estimation and sensitivity analysis.

```
Data: $X$, $Z$, and $R$ from observational data
Estimate $R|X, Z = 0$ as potential outcome $\widehat{R^0}$;
Match on $\widehat{R^0}$;
Estimate TET based on matching as $\widehat{\text{TET}}$;
for each level of confounding $\Gamma$, $\Delta$ do
    for each individual in the dataset do
        Compute probability $U_{\Gamma,\Delta} = 1$ given $X$, $Z$, $R$ for that individual;
        Sample $u$ for that individual accordingly
    end
    Estimate $R|X, U, Z = 0$ (using generated $u$);
    Match on $R^0$ using both $X$ and generated $U_{\Gamma,\Delta}$;
    Estimate TET based on matching as $\widehat{\text{TET}}_{\Gamma,\Delta}$;
    Compare $\widehat{\text{TET}}_{\Gamma,\Delta}$ with $\widehat{\text{TET}}$;
end
```

Algorithm 2: TET Sensitivity Analysis Algorithm

Chapter 4

Experimental Results

In this chapter, we show the results from our experiments regarding different topics we discussed in Chapter 3: First of all, we describe in detail the procedure for generating synthetic data under two different covariate settings, and with the generated synthetic data we compare the performance of propensity score matching and potential outcome matching in TET estimation. After that, we walk through the sensitivity analysis on estimated TET by introducing different levels of confounding into the synthetic data and examine the structural differences between the estimated TETs. Finally, we show a preliminary TET estimation based on VPS data in Section 4.6.

With our experimental results, we propose the following four conjectures:

1. Potential outcome matching performs much better than propensity score matching in TET estimation from observational data.
2. With no unmeasured confounding, it is possible to estimate the TET well using potential outcome matching, no matter whether the observed covariates are strongly or weakly predictive of treatment assignment and outcome.
3. With strongly predictive covariates, the estimated TET structure can be stable against strong Rosenbaum confounding encoded as a boolean.
4. With weakly predictive covariates, TET estimation can still provide some reasonable predictions of ITE against strong Rosenbaum confounding encoded as a boolean, but the structure of the tree will change.

4.1 Synthetic Data Setting

In this section, we describe in details how synthetic data was generated for our experiments.

Similar to the VPS data set, our synthetic data is also represented in a matrix format, where each row refers to one patient. With 50,000 individuals generated, each row has a series of covariates X to represent the patient's health metrics, with a boolean indicator of treatment Z assigned according to a treatment assignment model based on the patient's covariates. There are also two potential outcomes (R^0,R^1) for each patient generated from the outcome models, and each patient's outcome R^z is determined by the patient's covariates X and treatment assignment Z.

4.1.1 Covariates X

To have covariates X that mimic those of a patient's health metrics similar to the real world clinical data, we generate synthetic patient covariates following these principles:

- The distribution of synthetic covariates should be representative enough of a relatively large patient population, which means we need to generate covariates with a great diversity covering as many cases as possible.
- The synthetic covariates should include different types of covariates, specifically, we want to include both boolean covariates and real-valued covariates.
- To reflect the correlations between different covariates, which can be frequently seen in real world clinical data, we also want to generate X such that it involves interactions between covariates. That is, for a patient's covariates in X not all of them should be independent of each other. There should be dependencies between at least two of them.

In general, any data set satisfying these three principles could be used for our synthetic sensitivity analysis. For experimental purposes, we generate a relatively simple synthetic dataset with 8 covariates from X_0 to X_7, which we now describe.

$\boldsymbol{X_0, X_1}$
Covariates X_0 and X_1 are generated to serve as noise in our experiment. They have no causal relationship to other covariates, treatment, or outcome. X_0 is generated according to a Bernoulli distribution with probability of 0.5, while X_1 is generate from a standard normal distribution $N(0, 1)$.

$\boldsymbol{X_2, X_3}$

Covariates X_2 and X_3 are also randomly generated; however, they have a very strong causal relationship with some other covariates, the value of X_2 and X_3 can influence or even determine the value of some other covariates. X_2 is generated through the same Bernoulli distribution as the X_0 generator, and X_3 is generated from the same standard normal distribution of $N(0, 1)$ as in the X_1 generator.

$\boldsymbol{X_4, X_5, X_6, X_7}$

Values of these four covariates are strongly influenced by covariates X_2 and X_3. With X_4, X_5, X_6, X_7, we intend to introduce certain levels of interactions between different covariates, which increases the difficulties of both estimating treatment models that will be used in propensity score matching, as well as estimating outcome models used in estimated outcome matching. Among them, covariates X_4 and X_5 are in the format of boolean variables, while X_6 and X_7 are continuous valued. Pseudocodes for their generation can be found in Appendix B.1.

4.1.2 Outcome R

The outcome R should be generated individually for each individual based on their covariates X as well as their treatment assignment Z. For the convenience of estimated TET evaluation, we choose to generate both of the boolean potential outcomes without treatment r_i^0 and with treatment r_i^1 for each entry i based on individual i's covariates.

The reason to generate both of the potential outcomes for each individual is that with this information, specifically $r_i^1 - r_i^0$ the true ITE of individual i, we are able to directly learn the real TET as our benchmark for estimated TET evaluation. Note that these two potential outcomes are only used together when building the real TET, that is, when trying to estimate the TET we only use one of these two potential outcomes for each individual, with respect to the properties of observational study that these two potential outcomes can never be observed at the same time.

The potential outcome can be generated from any model, but to mimic the real world clinical setting, we require the model to follow these principles:

- The outcome model should not be a deterministic model, instead, it provides a boolean outcome based on a unique Bernoulli distribution calculated from patient covariates and treatment assignment for each individual patient, such that two patients with identical covariates and treatment assignment only share the same mean outcome, but not exactly the same outcome values.

- ITE of individual i, defined as $r_i^1 - r_i^0$, should depend on individual i's covariates alone.

- Treatment in general should benefit patients, that is, for the majority of patients $r_i^1 > r_i^0$ holds true, if a larger outcome value refers to a better health outcome.

- However, patients with extreme health status may not benefit from treatment, or treatment may tend to make the outcome worse. This reflects the fact that treatment might harm healthy patients with its side-effects, and for patients with extremely bad health status, treatment may not succeed at improving the outcome.

In general, any model satisfying these principles can be used as outcome models. For our experiment, for each individual we generate two pairs of potential outcomes with two different settings based on how influential the covariates X is to potential outcomes (R^0, R^1), here we describe them one by one.

"Strong" Covariate Setting

In this setting, covariates X strongly influence the outcomes, such that X can be used to predict these two potential outcomes with a very low error-rate. To generate outcomes with this setting, we have the two potential outcome models that are nearly deterministic.

We choose to build two separate models to generate R^1 and R^0, and these two models are in the simple decision tree format. Their structures can be found in Appendix B.2.

"Weak" Covariate Setting

In this setting, covariates X still influence the outcomes, but the influcence is weak compared with the "strong" setting. X is still very helpful in predicting these two potential outcomes; However, X does not contain all the information to predict outcomes perfectly, that is, using X to predict these two potential outcomes will have an obvious error-rate that cannot be easily ignored. To generate outcomes with this setting, we use the same structure as in the strong setting, but we introduce more randomness into the two potential outcome modes. Note that estimating ITE with this setting is essentially more difficult than the previous strong covariate setting, because the covariates contain less information helping us to distinguish individuals with different outcomes, which is very likely to have negative influences on our estimation.

We choose to build two separate models to generate R^1 and R^0 with the same decision tree structures as in the previous “Strong” setting, however, we change the probability inside each leaf node, such that the new models now generate outcomes with more randomness compared with the previous “Strong” setting. These two models can be found in Appendix B.3.

4.1.3 Treatment Z

Similar to the potential outcomes, treatment assignment Z should also be individually generated for each entry based on an individual's covariates X. In general, any treatment assignment model can be used in our experiment, as long as it does not violate the Strongly Ignorable Treatment Assumption for propensity score matching. So we require the treatment assignment model to follow these principles:

- The treatment assignment model is not deterministic, which means for each patient the models returns a non-zero probability of being treated.
- This non-zero probability is calculated for each patient solely based on the patient's covariates.
- In order to generate synthetic data for matching, the probability of treatment assignment should not be equal across all individuals, which means the probability of being treated for each individual should cover a wide range of $(0.0, 1.0)$.
- To mimic the real-world treatment assignment strategy, tree models taking individual covariates as inputs can be designed for treatment generation, that may reflect the implicit decision-making approach of clinicians.

For our experiments, we design the treatment assignment model as a decision tree taking individual's covariates X and providing a probability of assigning this individual treatment. The detailed model structure can be found in Appendix B.4.

4.1.4 Confounder U

We generate confounders with different levels of confounding according to the approach we described in Section 3.3.1. As we have two different outcome settings, namely, “Strong” and “Weak”, we generate confounders with the same level of confounding for each of them separately.

To examine the robustness of estimated TET structures with different types of confounding *besides* Rosenbaum confounding, we also include the conditional probabilities of boolean confounder U being 1 as the continuous value variable PU into the matching process. This probability can be regarded as an extreme type of confounding, which we will discuss later in our experiment results.

4.2 Matching Settings

In order to show that outcome potential matching performs better than propensity score matching when estimating the TET, we apply both of these two matching methods with the same dataset of covariates X, treatment assignment Z, outcome R, as well as the boolean confounder U when confounding is introduced.

For propensity score models, we choose both the classic logistic regression model as well as the decision tree model. For potential outcome model, we choose the decision tree model alone, for we believe a decision tree is a more flexible model for grouping individual with the same expected outcome.

As we focus on the ITE of treated individuals, we match each treated individual with control individuals. To further reduce bias in the final matching results, we choose to conduct matching with replacement, that is, each treated individual can be matched to multiple control individuals, and each control individual can be used for matching multiple times. We merge all the matched pairs with the same treated individual as a single match pair by weighting each match pair with respect to the number of times the control individual gets matched, this merging process allows us to benefit from matching with replacement without introducing more bias into the covariates of the treated group. We use the Matching package [24] for R [26] to conduct matching in all of our following experiments.

4.3 TET Settings

To evaluate the performance of estimated TET, we need to know the structure of the *real* TET and compare all estimated TETs with this real TET.

A real TET can be learned from the real ITE. As we already have two potential outcomes generated for each individual in our synthetic data, we can access the real TET easily. The real TET can be learned by firstly labeling each individual i with $r_i^1 - r_i^0$, which is the real ITE of i, then use a decision tree learning algorithm to predict the real ITE based on individual i's covariates. Note that when learning the real TET, all the information we need are covariates X, and the two potential outcomes R^0, R^1. We do not need any information of treatment assignment, as the true ITE should not be influenced by treatment Z according to the Strongly Ignorable Assumption.

This learned real TET will be used as the benchmark to evaluate the estimated TET from either propensity score matching or estimated outcome matching. After matching, for each matched pair (i,j), assuming individual i is given the treatment, we use $r_i - r_j$ to label the covariates of individual i, then learn another decision tree to predict $r_i - r_j$ based on individual i's covariates. The learned tree will be our estimated TET.

In our experiment, we use the NLS-DT algorithm to learn our decision tree. In each covariate selection process we set the bootstrap number to 50, and we choose the most voted covariate that has been voted at least 25 times as our splitting covariate. The learned NLS-DT is then pruned using error-reduce pruning.

4.4 Estimating TET

The first experiment we conduct with our synthetic data shows that we can estimate the TET from observational data with the help of estimated outcome matching, while we cannot estimate the TET well with propensity score matching.

As we have mentioned before, for each individual i in the synthetic data, we generate both of the potential outcomes r_i^1 and r_i^0 with and without treatment. With this information we can access the true ITE of each individual and represent the real TET by decision tree learning. Note that this is an experiment for estimating TET without any confounding, so we do not include the generated confounder U when conducting this experiment.

The procedure of representing the real TET is straightforward: for each individual i we take all its covariates x_k as splitting candidates, we label the covariates of each individual i with the true ITE, which is defined as $r_i^1 - r_i^0$ and can be easily computed from these two potential outcomes we generated. We use these labeled individuals as our training data to learn a decision tree via the NLS-DT algorithm. The final real TET learned from the true ITE in "Strong" setting data is shown in Figure 4.1, and the real TET in "Weak" setting data is shown in Figure 4.2.

Comparing the real "Strong" TET in Figure 4.1 learned from the true ITE of each individual in the dataset, we see that the real TET shares a strong similarity with each

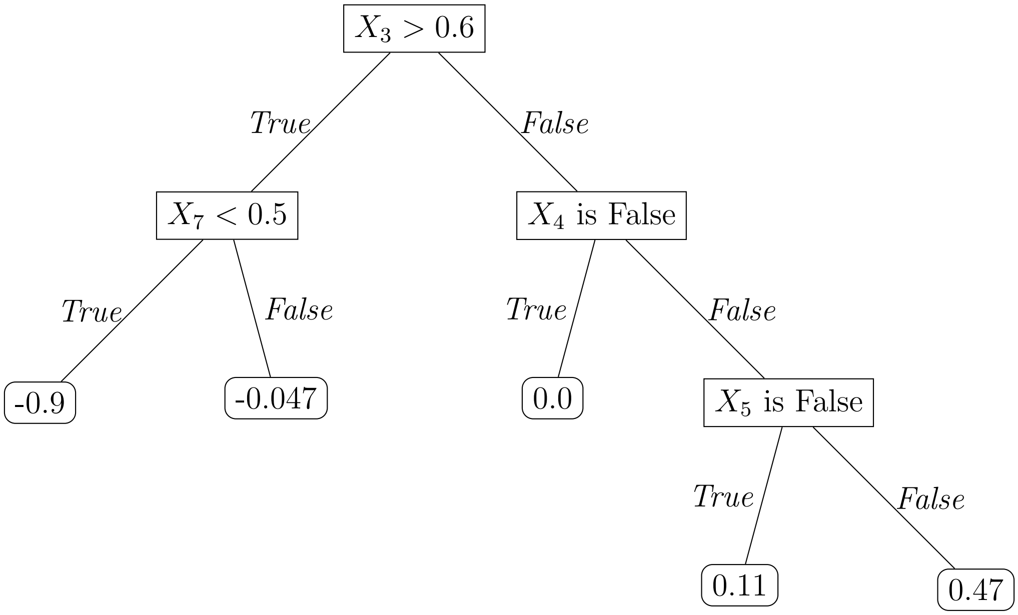


Figure 4.1: Real TET learned from “Strong” covariate setting data

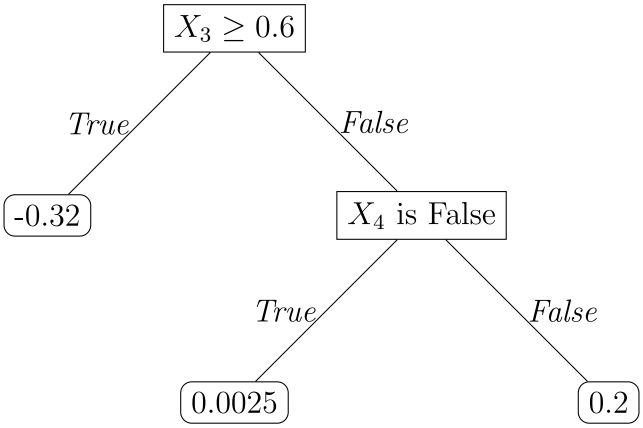


Figure 4.2: Real TET learned from “Weak” covariate setting data

of the outcome models shown in Appendix B.2. In fact, it can be regarded as these two outcome models “merged” together. The cross-validated error-rate of real “Strong” TET without confounding learned using true ITE is 4.958%; it shows that this real TET performs very well in estimating ITE.

The real “Weak” TET shown in Figure 4.2 learned from the true ITE of each individual can be regarded as a simplified version of the real "Strong" TET. This can be explained as we have more randomness in the "Weak" data setting, such that covariates X contain less information helping us to estimate the ITE. This missing piece of information helps the real "Strong" TET to fully develop its structure, while it stops the real "Weak" TET from further growing. The cross-validated error-rate of real "Weak" TET without confounding learned using true ITE is around 24.8%, which means this real TET does not perform so well in estimating the true ITE.

Both of these real TETs are the benchmarks to evaluate all estimated TET, for with observational data we would never have access to the true ITE of each individual, thus we can only estimate the structure of real TET. In another words, these two real TETs *are* the models assigning different levels of treatment effect given individual's covariates. And we should not expect the estimate TETs perform even better than these two real TETs.

4.4.1 Estimate TET with propensity score

Now we show that TET estimated from propensity score matching does not perform well, that is, the estimated TET learned from propensity score matching results does not represent the structure of the real TET as shown in Figure 4.1 and Figure 4.2. We show the following TETs estimated from propensity score matching with two different propensity score models.

Logistic Regression Model

Logistic regression is the most common propensity score model used to estimating ATT and ATE. The following estimated TET is learned from all treated individual's covariates, with each entry of covariates labeled as $r_i - r_j$, assuming in matched pair (i, j) individual j is from the control group and is matched to i based on propensity scores provided by a logistic regression model:

Comparing the estimated "Strong" TET shown in Figure 4.3 with the real "Strong" TET shown in Figure 4.1, we see that even though the right branch of the estimate TET does share a certain amount of similarity with the real TET, the general structure of this estimated TET does not represent the real TET at all: it even picks a different covariate as the root node.

Considering the fact that neither our treatment model or outcome models are complex or difficult to predict in the "Strong" setting, the estimated TET should be close to the

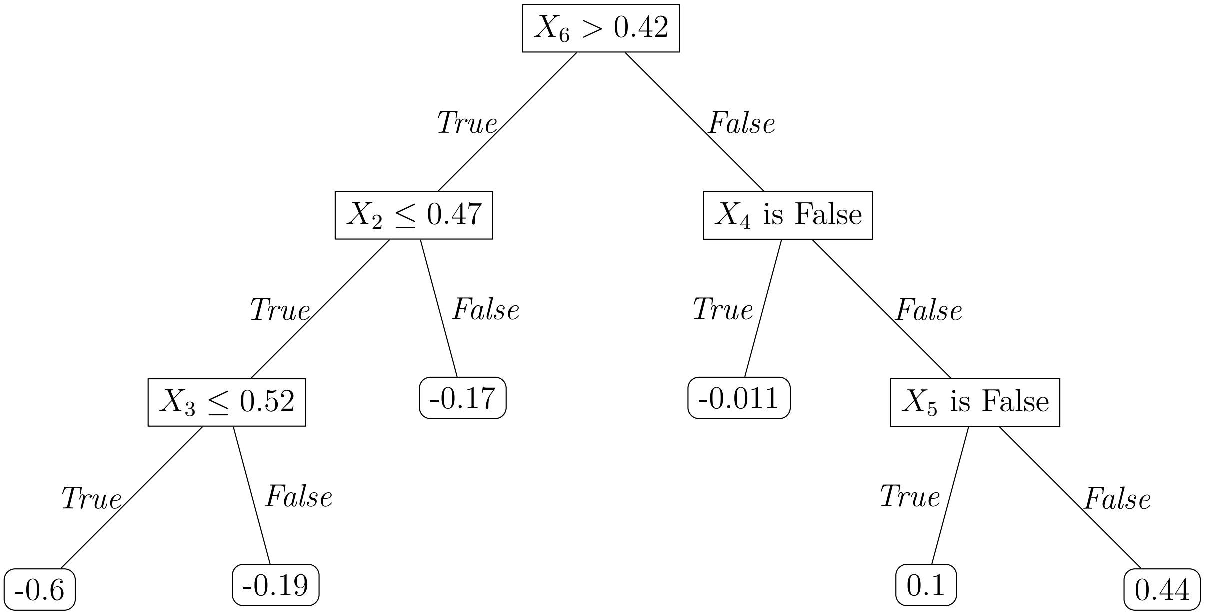


Figure 4.3: Estimated TET from "Strong" covariate setting data using logistic regression propensity score matching

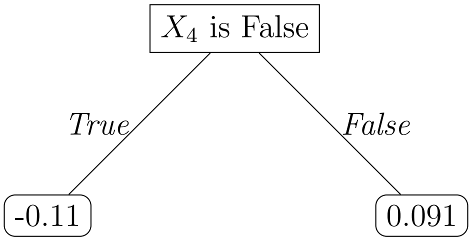


Figure 4.4: Estimated TET from "Weak" covariate setting data using logistic regression propensity score matching

real TET if the matching is producing match pairs that are meaningful for ITE estimation. Apparently, propensity score matching with logistic regression as the propensity score model cannot represent the real TET, and should not be considered for TET estimation.

On the other hand, the estimated "Weak" TET comparing with the real "Weak" TET does not helping us at all to estimate the ITE: they are totally different in structure, the estimated TET even picks the only node incorrectly.

Decision Tree Model

One might argue that the estimated TETs in Figure 4.3 failed to represent the real TET structure because logistic regression is not the correct propensity score model. To further show that propensity score matching does not help when estimating TET, we use a different propensity score model: the decision tree model. This is the "correct" propensity score model as our synthetic data uses a decision tree to select treatment. The TETs estimated using propensity score matching with decision tree as the propensity score model are shown in Figure 4.5 and Figure 4.6.

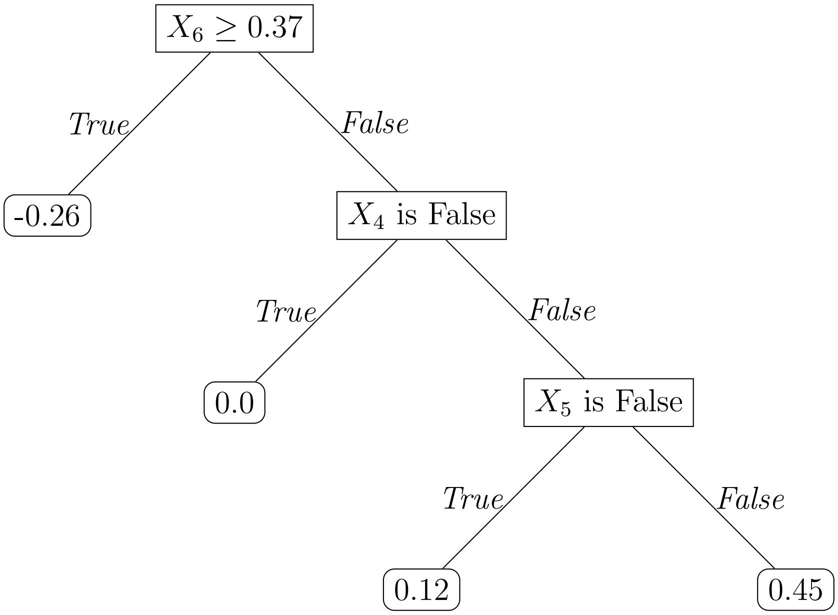


Figure 4.5: Estimated TET from "Strong" covariate setting of data using decision tree propensity score matching

If we compare this estimated "Strong" TET with the real "Strong" TET in Figure 4.1, similar to the previous estimated "Strong" TET in Figure 4.3 with logistic regression model, it still fails to pick the correct covariate for the root node, and it loses more information to further classify its left branch. The general structure is different from the real TET, and most importantly, it fails to distinguish the potential ITE differences hidden in its left branch by simply predicting all individuals with $X_6 \geq 0.37$ as benefiting from the treatment, which is not only incorrect but also very dangerous: for as we know in the real TET shown in Figure 4.1 there is a subgroup of individuals in the left branch that treatment tends to hurt instead of improving their outcomes.

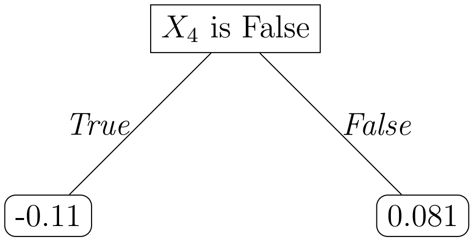


Figure 4.6: Estimated TET from "Weak" covariate setting of data using decision tree propensity score matching

As before, the structure of the estimated "Weak" TET compared with the real "Weak" TET does not make any sense.

By comparing estimated TET shown in Figure 4.3 and estimated TET shown in Figure 4.5 based on two different propensity score matching approaches, we show that propensity score matching does not perform well in estimating TET in both "Strong" and "Weak" settings, more specifically, using each matched pair based on propensity score matching as training data to learn a decision tree predicting ITE given covariates only results in a decision tree with structure far from the real TET.

4.4.2 Estimate TET with potential outcome

Now, we show that estimated outcome matching with a decision tree to estimate the potential outcome can provide reliable matched pairs such that they can be used to train a decision tree as the estimated TET.

As we focus on the ITE on the treated individuals, we use the decision tree model to predict each individual's potential outcome without treatment, this decision tree, namely the potential outcome tree, is learned from summarizing all control individual's outcome given their covariates. After that, we feed this potential outcome tree with all individuals covariates from our data, such that we have the estimated outcome without treatment for all treated and control individuals. We then use this estimated potential outcome as the criteria to conduct our estimated outcome matching with replacement between the treated and control groups.

Finally, for each matched pair (i, j), we take the covariates of the treated individual i, label the covariates with the actual outcome differences $r_i - r_j$. We use these labeled covariates as training data to learn a decision tree using the NLS-DT algorithm. The learned decision tree is regarded as our estimation of TET (See Figures 4.7 and 4.8):

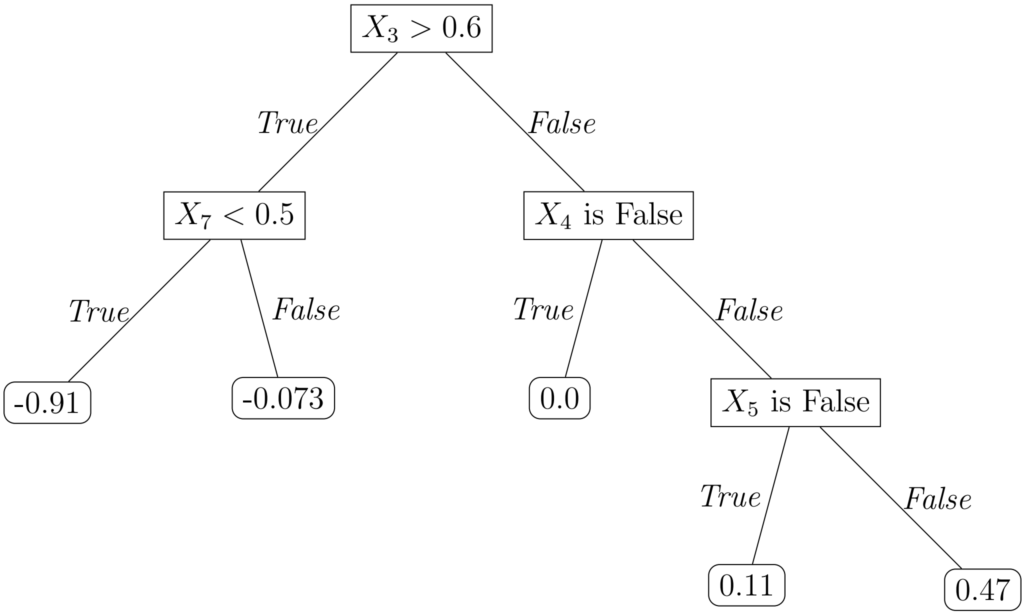


Figure 4.7: Estimated TET from "Strong" covariate setting of data with estimated outcome matching with decision tree outcome estimator.

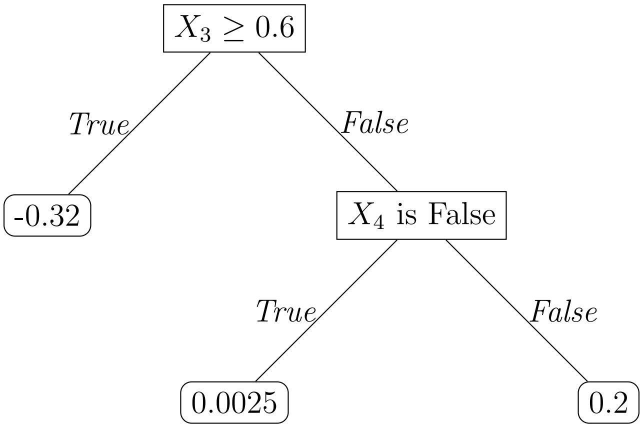


Figure 4.8: Estimated TET from "Weak" covariate setting of data with estimated outcome matching with decision tree outcome estimator.

We immediately observe the estimated "Weak" TET is exactly the same as the real "Weak" TET, we also observe the estimated "Strong" TET is almost exactly the same as the real "Strong" TET in Figure 4.1, in fact, they share exactly the same structure, only with minor differences between some of the leaf node values, which can be explained as random noise.

The cross-validated error-rate of the estimated "Weak" TET is exactly the same as the real "Weak" TET, and the error-rate of the estimated "Strong" TET without confounding learned through outcome matching is 4.991%, which is very close to the error-rate of the real "Strong" TET learned on true ITE. This shows that this estimated TET via outcome matching performs well in estimating ITE from observational data.

Considering the fact the real 'Strong" TET shown in Figure 4.1 is trained with all individual's true ITE as the label, while the estimated "Strong" TET in Figure 4.7 is only trained with the matched outcome differences as the label, and we only use the covariates of all treated individuals. It is obvious that the treated individuals are only a sub-population of the data, and this treated group is expected to have a strong bias over all covariates, they can all be regarded as increasing the difficulty of estimating the real TET.

However, as we have shown, the TET estimated from estimated outcome matching represents the real TET perfectly. In fact, in our experiment they can be regarded as the same decision tree. This shows that matching with potential outcome predicted by a decision tree model can provide us matched pairs such that we label them with the actual outcome differences between individuals matched together, and use the matched results to estimate the real TET.

4.4.3 Conjectures

So far, with our experimental results, we have attempted to prove the evidence for following two conjectures:

- With observational data, matched pairs from propensity score performs badly in estimating the real TET.
- With observational data, matched pairs from potential outcome with decision tree estimating the outcomes performs very well in estimating the real TET.

4.5 Sensitivity Analysis

As we have shown that TET can be estimated successfully based on potential outcome matching, in this experiment, we now focus on evaluating the robustness of the estimated

TET to unmeasured confounding. We show how to conduct sensitivity analysis on the estimates TET; for this, we include the confounder U in our synthetic data.

We have set the confounder as a boolean variable. To generate them we firstly compute the conditional probability of confounder U being 1 for each individual given the covariates, treatment, as well as outcome, then we convert the probability into a boolean confounder for each individual through a randomization process. A diagram showing the concepts of sensitivity analysis in our experiment can be found in Appendix A.

4.5.1 Confounding Settings

We introduce our synthetic data with different levels of confounding defined in Dual Sensitivity Analysis that can be described with parameter Γ and Δ. According to the method we developed in Section 3.3.1, for the simplicity of our experiments, we generate four different sets of confounder, denoted as $U_{\Gamma,\Delta}$, with different confounding controlled by parameters Γ and Δ used in Section 2.4.3:

- $U_{4,4}$, refers to the situation where with the same covariates, the confounder being true will strongly *increase* the probability of being treated and the probability of corresponding outcome being 1.

- $U_{4,0.25}$, refers to the situation where with the same covariates, the confounder being true will strongly *increase* the probability of being treated, while strongly *decrease* the probability of corresponding outcome being 1.

- $U_{0.25,4}$, refers to the situation where with the same covariates, the confounder being true will strongly *decrease* the probability of being treated, while strongly *increase* the probability of corresponding outcome being 1.

- $U_{0.25,0.25}$, refers to the situation where with the same covariates, the confounder being true will strongly *decrease* the probability of being treated and the probability of corresponding outcome being 1.

Note that each of these eight confounder sets will have a very strong level of confounding, as most of the Rosenbaum and Dual Sensitivity Analysis only focus on Γ and Δ within a much smaller range. The reason we introduce very strong confounding into the estimated outcome matching is that, though confounding is expected to change the matching results, we have no idea how it will eventually influence the estimated TET structure based on estimated outcome matching. In fact, with such a strong confounding, we expect to see the real TET without confounding being totally different from the real TET with confounding, and the estimated TET without confounding being totally different from the estimated TET with confounding.

For the real TET without confounding and estimated TET without confounding, both of them will always maintain exactly the same structure as in Figure 4.1 and Figure 4.7, since we are only generating confounder $U_{\Gamma,\Delta}$ with different levels of confounding for the same set of data as we discussed in Section 4.1, datasets $X, Z, R, U_{\Gamma,\Delta}$ will always maintain the same covariates X, treatments Z, as well as outcomes R. Considering the fact that we only need X, Z, R to learn the real TET or estimate the TET without confounding, their structures will not change.

In the following sections, we use the notation of $\mathrm{TET}_{U,\Gamma,\Delta}$ to denote the real TET learned with confounding dataset $U_{\Gamma,\Delta}$.

4.5.2 TET with confounding

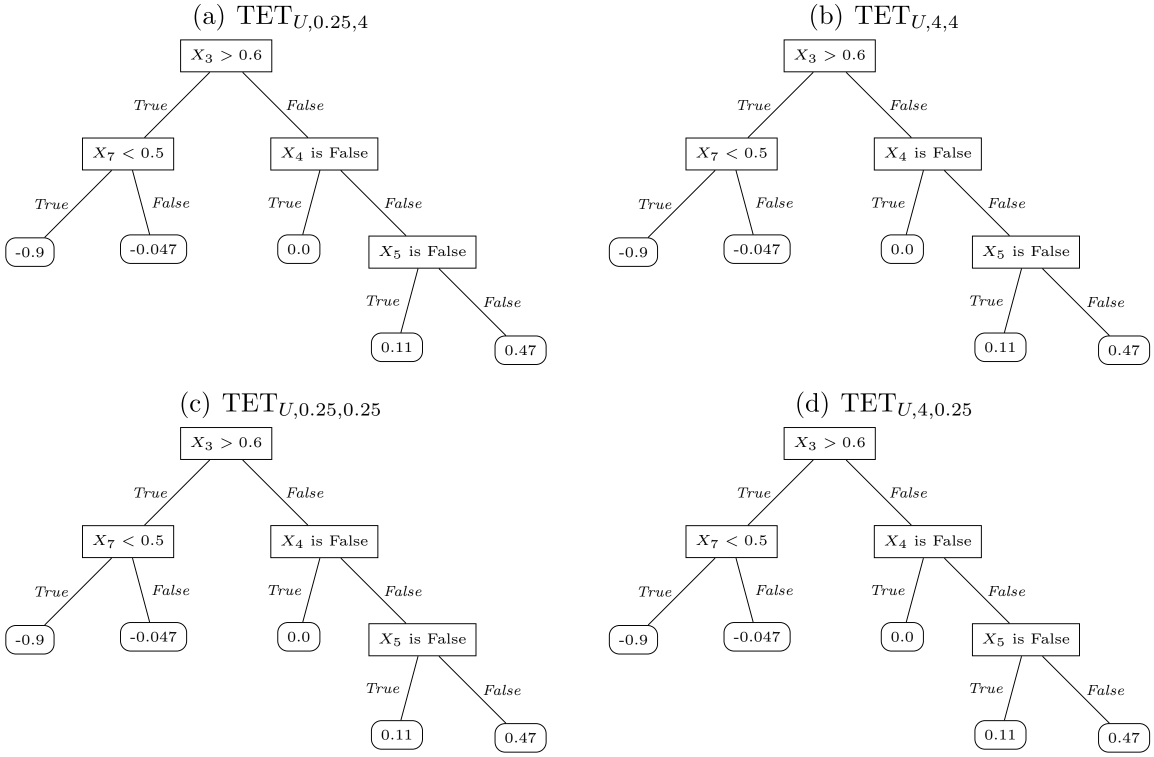
In this section, we separately learn the real TETs with boolean confounder $\mathrm{TET}_{U,\Gamma,\Delta}$, as well as their estimations with estimated outcome matching. For all TET estimations, we use a decision tree predicting potential outcomes without treatment learned from all control individuals, and for the outcome matching we choose to match with replacement. And we show the experimental results with "Strong" and "Weak" covariate settings separately.

With "Strong" Covariate Setting

To our surprise, with the "Strong" covariate setting data, the experiment result shows all four $\mathrm{TET}_{U,\Gamma,\Delta}$ shown in Figure 4.9 remain the same structure as the real TET without confounding shown in Figure 4.1, at the same time, all their four estimations $\widehat{\mathrm{TET}}_{U,\Gamma,\Delta}$ as shown in Figure 4.10 also remain the same structure as the estimated TET without confounding shown in Figure 4.7. The error-rate of these decision trees are also shown in Table 4.1 and Table C.3.

The real "Strong" TET and estimated "Strong" TET keeping their original structure even with such a strong confounding encoded as boolean confounders means a lot to our ITE estimation mechanism: it means that even if there is a boolean confounder with a very strong Rosenbaum confounding [1] in the observational data, we may still be able to estimate the ITE by estimating the TET through estimated outcome matching, and the estimated

[1] Means the confounding can be controlled as described by Γ and Δ together.



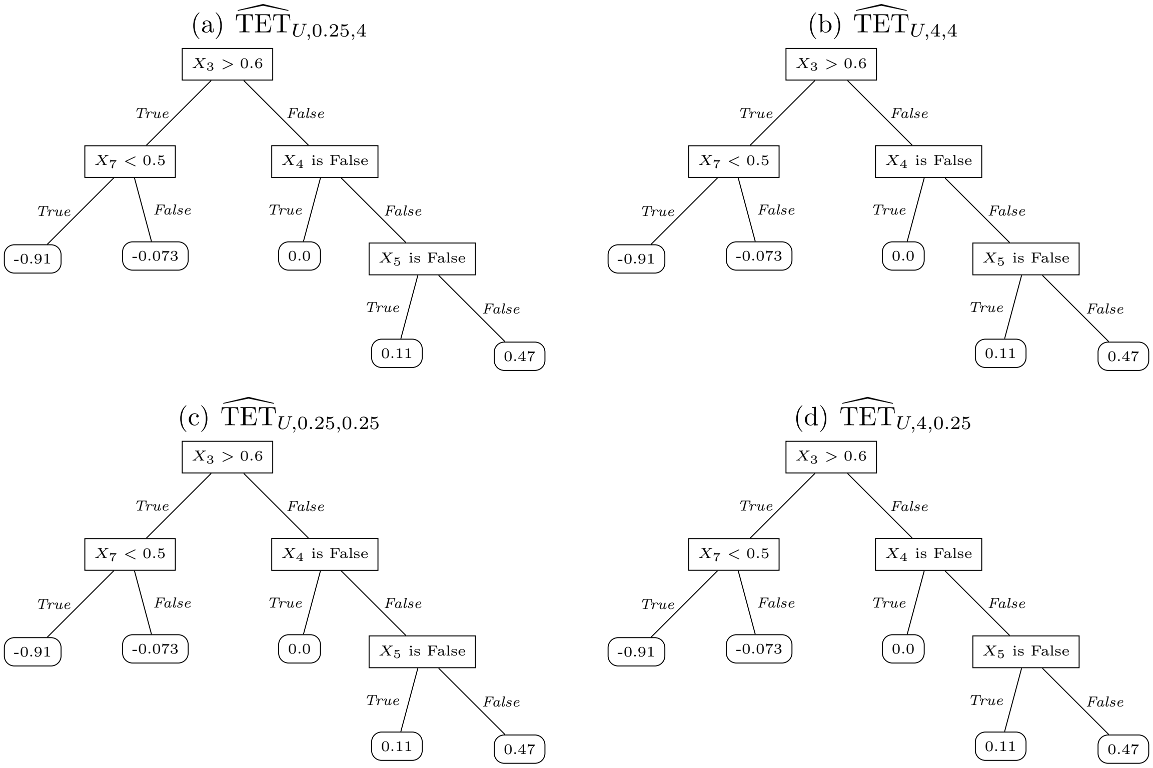
(a) $\text{TET}_{U,0.25,4}$

(b) $\text{TET}_{U,4,4}$

(c) $\text{TET}_{U,0.25,0.25}$

(d) $\text{TET}_{U,4,0.25}$

Figure 4.9: Comparison between $\text{TET}_{U,\Gamma,\Delta}$ models from "Strong" covariate setting of data



(a) $\widehat{\text{TET}}_{U,0.25,4}$

(b) $\widehat{\text{TET}}_{U,4,4}$

(c) $\widehat{\text{TET}}_{U,0.25,0.25}$

(d) $\widehat{\text{TET}}_{U,4,0.25}$

Figure 4.10: Comparison between $\widehat{\text{TET}}_{U,\Gamma,\Delta}$ models with “Strong” covariate setting of data

Table 4.1: Comparison between error-rates of $\text{TET}_{U,\Gamma,\Delta}$ with “Strong” setting data

Γ Δ	0.25	4
0.25	4.957%	4.958%
4	4.958%	4.957%

Table 4.2: Comparison between error-rates $\widehat{\text{TET}}_{U,\Gamma,\Delta}$ with “Strong” setting data

Γ Δ	0.25	4
0.25	4.991%	4.991%
4	4.991%	4.991%

TET can be stable against a boolean confounder with a strong confounding defined by Γ and Δ.

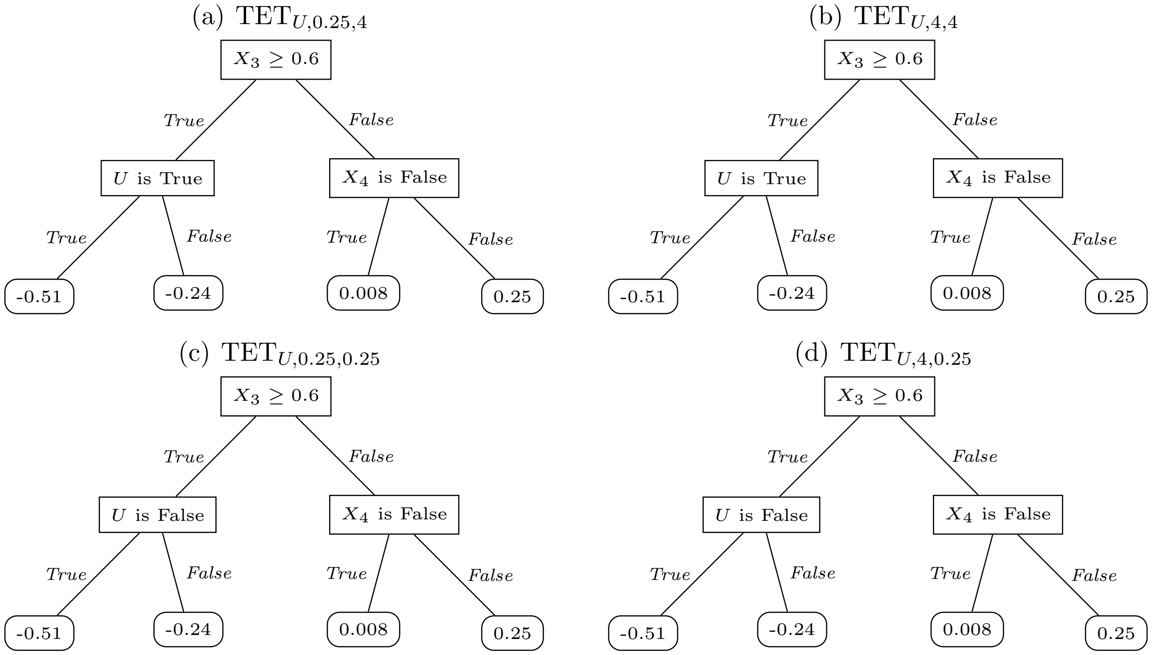
With “Weak” Covariate Setting

Different from the previous results, with the “Weak” covariate setting data, the experiment result shows all four $\text{TET}_{U,\Gamma,\Delta}$ shown in Figure 4.11 change their structures compared with the real “Weak” TET without confounding shown in Figure 4.2. However, this structural change can be regarded as an improvement over the original real “Weak” TET, since the estimated TET generally develops leaf nodes with the information provided by U based on the original real “Weak” TET. We show their improved error-rates in Table 4.3.

At the same time, some of the four estimated TETs in the “Weak” $\widehat{\text{TET}}_{U,\Gamma,\Delta}$ setting as shown in Figure 4.10 also change their structures. The error-rate of these estimated “Weak” TETs are also shown in Table 4.3 and Table 4.4.

We observe that the real “Weak” TETs in Figure 4.12 change their structures by adding the confounder U as splitting nodes while maintaining the original tree structures, the error-rate comparison in Table 4.3 shows that this structural change slightly improves the predictive power of the original real “Weak” TET from 24.8%.

While comparing the estimated “Weak” TETs in Figure 4.12 with the real “Weak” TETs they attempt to estimate in Figure 4.11, we observe that the estimated “Weak” TETs also change their structures to mimic the real “Weak” TETs learned from the real ITE. Some of them do not share exactly the same structures with each other, also, these



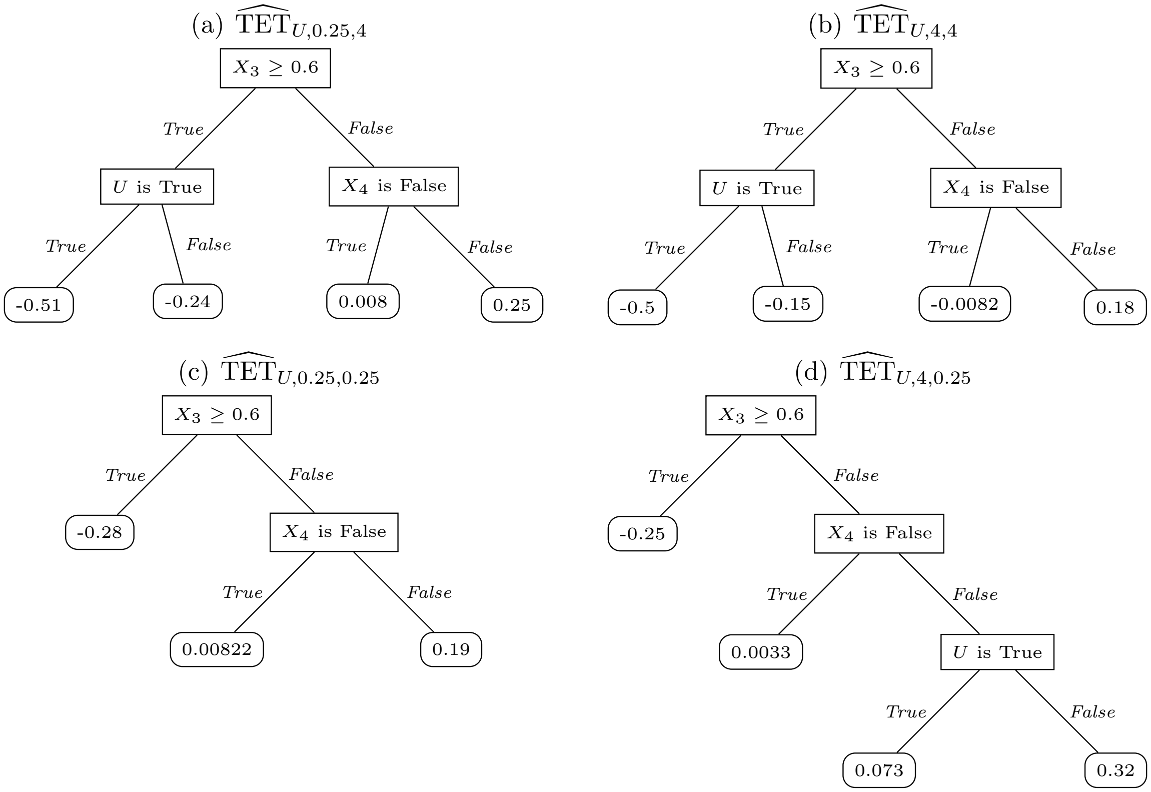
(a) $\mathrm{TET}_{U,0.25,4}$

(b) $\mathrm{TET}_{U,4,4}$

(c) $\mathrm{TET}_{U,0.25,0.25}$

(d) $\mathrm{TET}_{U,4,0.25}$

Figure 4.11: Comparison of $\mathrm{TET}_{U,\Gamma,\Delta}$ models from “Weak” covariate setting of data



(a) $\widehat{\mathrm{TET}}_{U,0.25,4}$

(b) $\widehat{\mathrm{TET}}_{U,4,4}$

(c) $\widehat{\mathrm{TET}}_{U,0.25,0.25}$

(d) $\widehat{\mathrm{TET}}_{U,4,0.25}$

Figure 4.12: Comparison of $\widehat{\mathrm{TET}}_{U,\Gamma,\Delta}$ models from "Weak" covariate setting of data

Table 4.3: Comparison between error-rates of $\text{TET}_{U,\Gamma,\Delta}$ with “Weak” setting data

Γ Δ	0.25	4
0.25	24.2%	24.3%
4	24.3%	24.3%

Table 4.4: Comparison between error-rates $\widehat{\text{TET}}_{U,\Gamma,\Delta}$ with “Weak” setting data

Γ Δ	0.25	4
0.25	24.5%	24.9%
4	25.1%	24.5%

estimated TETs do perform reasonably well in estimating the real TETs. This is be shown by their error-rates in Table 4.4.

Considering the fact that covariates in the “Weak” setting are not very helpful for ITE estimation, and we are comparing the real TETs learned from all population with the estimated TETs learned from treated population. With such a strong confounding encoded as boolean confounders, the experiment results reveals a lot of TET estimation mechanism: we can infer that even if there is a boolean confounder with a very strong Rosenbaum confounding in the observational data, we may still be able to estimate the ITE by estimating the TET through estimated outcome matching. Even with covariates very weak for ITE estimation, the estimated TET shows up to be stable against a boolean confounder with a strong confounding defined by Γ and Δ.

With Extreme Confounding

To further experiment with the estimated TET’s structure stability, we also estimated the TET using datasets containing the *conditional probability* of U being 1 rather than containing U itself. This probability can be regarded as an extreme confounding that does not directly correspond to Rosenbaum confounding and so is not a main part of this thesis, but we include the detailed experimental results in Appendix C for the interested reader.

4.5.3 Conjectures

From the experimental results of sensitivity analysis on estimated TET, we propose the following conjectures:

- With observational data of strong covariates, TET estimated based on outcome matching shows to be stable against strong Rosenbaum confounding encoded as boolean confounders.
- With observational data of weak covariates, TET estimated based on outcome matching can still provide reasonable ITE estimations in situations of strong Rosenbaum confounding encoded as boolean confounders.

4.6 Preliminary Application on VPS data

In this section, we show the preliminary result of TET estimation based on potential outcome matching from the real-world VPS data, we also conduct a preliminary sensitivity analysis on the estimated TET by introducing Rosenbaum confounding into the VPS data.

4.6.1 VPS Data

Collected by the Virtual PICU System (VPS) and Children's Hospital Los Angeles (CHLA), the VPS data contains information of 58,772 individuals, with each ICU patient recorded as an entry of 138 covariates, 1 treatment assignment, and 3 outcomes. The covariates are recorded in different types including boolean, real-valued, and categorical variables. The treatment assignment is recorded as a boolean with True indicating patient is ventilated. The outcomes are observed and recorded after the treatment assignment. In the VPS data, 18,610 patients are treated with ventilation, and 40,162 patients are not given ventilation as treatment.

Due to the high missing-rates of some of the covariates and outcomes, we use mortality status as the outcome for our analysis. Among all 58,772 patients, 1,521 of them do not survive during their stay in ICU. In our experiments, we encode the outcome as 1 (True) if a patient did **not** survived in the ICU, and as 0 (False) for patient survived in ICU as shown in Table 4.5. A VPS data description with more details can be found in Appendix D.

Table 4.5: Outcomes of treated and control groups in VPS data

	$Mortality = 1$	$Mortality = 0$
Treated	1360	17250
Control	161	40001

4.6.2 VPS TET Estimation

For the TET estimation, we use the same matching setting as in our experiments with synthetic data, that is, we conduct matching with replacement and merge all matched pairs with the same treated individual as one. We label them with the actual outcome differences, and use these labeled covariates to learn a decision tree through the NLS-DT algorithm. We show the learned decision tree as our estimated TET in Figure 4.13.

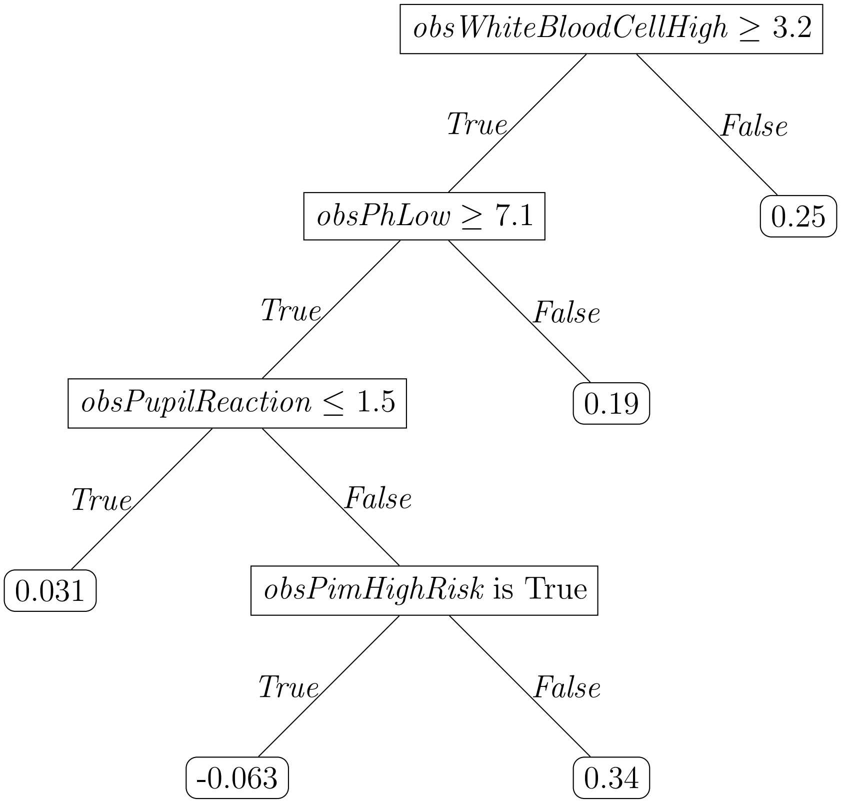


Figure 4.13: Estimated TET from VPS

After observing the estimated TET, we notice that the branches indicating the patients with worse health conditions (e.g. pupil reaction being low is bad, and white blood cell count bing high is also bad) tend to have better[2] treatment effect from ventilation, while patients with normal health conditions tends to have worse treatment effect from ventilation. This in general makes sense, since ventilators are frequently used to try to save very sick patients, while for more healthy patients ventilation may be less necessary. One strange feature of the tree is the question about *obsPhLow*; normal blood pH is approximately 7.4, so the "True" branch of this question actually indicates healthier patients, at least according to blood pH.

We also notice this estimated TET attempts to identify the subgroup of patients who benefit from ventilation shown as the leaf node with ITE of -0.063. (Recall that in this case, a negative treatment effect reflects a decrease in mortality.) However, the estimated TET shows that only a small group of patients would benefit from ventilation. Part of the reason may be the VPS data has a very high missing rate for a great number of covariates. For this preliminary estimation of TET we only select those patients with complete covariates recorded in the data. Given the fact that patient's covariates are sometimes missing for a reason, such as clinicians skip the recording because the covariates are normal[3], this selecting process essentially introduces even more covariate bias into our estimation, that is, we are very likely selecting a subgroup of extremely sick patients for our estimation, which partially explains why the estimated ITE does not show up as significantly helping patients. Another possible problem is that, unlike in our synthetic data settings, the outcomes in the VPS data are extremely unbalanced–more than 97% of all patients survived in the ICU; therefore we have very limited data to distinguish patients who died from patients who survived.

4.6.3 VPS TET Sensitivity Analysis

To evaluate the structural quality of the estimated TET in Figure 4.13, we introduce a strong level of Rosenbaum confounding as the boolean confounder U using the same approach described in Section 3.3.1. We set both Γ and Δ as 4.0, since this is the confounder we worry about the most for the ventilator triage problem: individuals having this confounder U as True are very likely to have bad outcomes and being treated at the same time, which tends to make the estimated treatment effect worse than the real treatment effect. We show the estimated TET with this strong Rosenbaum confounding in Figure 4.14.

[2] In this case, less positive.

[3] As confirmed by our co-workers from VPS.

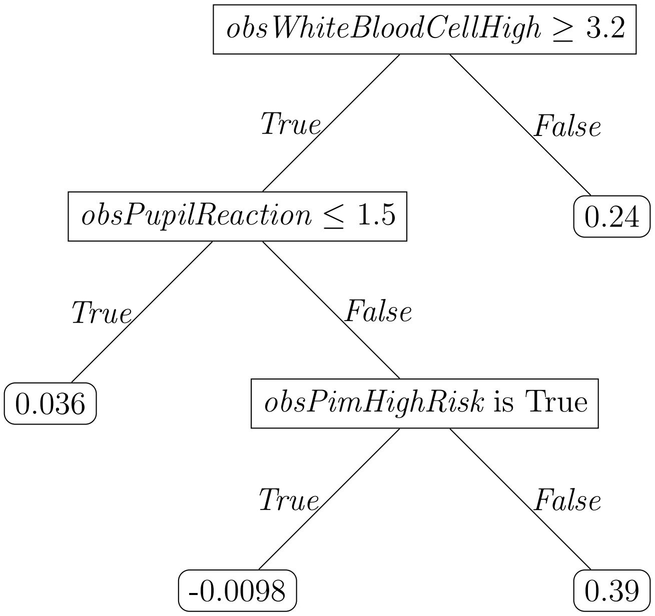


Figure 4.14: Estimated TET from VPS

As we can observe, even though the original estimated TET only distinguishes a small subgroup of patients benefiting from the ventilation, it still shows up to be somehow "stable" against a very strong level of Rosenbaum confounding: with confounding the estimated TET slightly changes its structure by removing a splitting node, and changes some of its leaf node values. However, we do observe that this confounder influences the TET estimation such that treatment effect tends to get a bit worse[4], but in general the structure of estimate TET remains to be stable, except that the node asking about blood pH is no longer present.

One potential explanation is that the covariates in the VPS data predict treatment and outcome well, simply because mortality is so rare. So we may be in a situation similar to the "Strong" covariate setting we proposed in Section 4.5.2, such that even with strong Rosenbaum confounding its structure remains relatively stable.

Generally, a meaningful TET estimation of this VPS data requires dealing well with the data missing value problem, such that it is possible to estimate the TET for the whole treated individuals with less bias. In the future we plan to either impute the missing values or encoding missingness as another covariate to further study the TET estimation on VPS data.

[4] In this case, leaf node values slightly increase.

Chapter 5

Conclusions

Estimating the Average Treatment Effect (ATE) from observational data has been well studied by statisticians and can be achieved by models such as propensity score matching, where its estimated treatment effect is averaged across either the treated population or the whole population from data. In this work, we attempt to solve a related problem known as the Individual Treatment Effect (ITE) estimation problem, which is motivated by the real-world ventilator triage problem. We search for a model that can be learned from observational data, and we expect this model can help to estimate and predict different levels of treatment effect on each individual by examining an individual's covariates.

We proposed the Treatment Effect Tree (TET) model to solve the ITE estimation problems, inspired by the classic propensity score matching that has been widely used for ATE estimation. Instead of learning the propensity score, this method learns the potential outcome without treatment using each control individual's covariates labeled with the actual outcome as training sample for decision tree algorithm. After that we use this learned potential outcome tree model to estimate each individual's potential outcome in the data, such that each individual corresponds to a potential outcome estimation. We then match each treated individual with one or several control individuals based on their potential outcome similarity. Finally, for each matched pair, we label the treated individual's covariates with the difference between treated individual's actual outcome and control individual's outcome. These labeled covariates are used to train a Node-Level-Stabilized (NLS) decision tree, and the learned NLS tree can be used to estimate ITE given an individual's covariates.

Considering the fact that matching can be vulnerable to unmeasured confounding, which is the hidden covariate independent of other covariates while influencing the treatment assignment and outcome at the same time, inspired by the Rosenbaum and Dual Sensitivity Analysis, we proposed an empirical sensitivity analysis as an extension of Dual Sensitivity Analysis to evaluate the robustness of estimated TET in situations of different levels of confouding.

To evaluate our proposed methods, we generated synthetic data including covariates, treatment assignments, as well as two potential outcomes (with and without treatment) for an individual. With these two potential outcomes we access the real TET using the true ITE of each individual in our data. We then modify the synthetic data such that only one potential outcome corresponding the individual's treatment assignment can be included in the data. We estimate the TET based on estimated outcome matching, and from the experimental results we find the estimated TET based on estimated outcome matching shares the same tree structure as the real TET learned from real ITE. We show that TET is the model to estimate for ITE estimation problems, and estimated outcome matching can provide matched pairs helpful for estimating TET. We also notice that even though propensity score matching performs very well in estimating ATE, the TET estimated based on propensity score matching results does not represent the real TET structure at all.

During the experiments we find that though with a very strong level of Rosenbaum confouding, both the real TET and the estimated TET learned from data with boolean confounders managed to maintain their original structures as long as covariates are strong in predicting outcomes, this can be explained as the boolean confounder does not influence the TET estimation or the ITE very much compared with covariates. We argue that given strong covariates, the structure of estimated TET based on estimated outcome matching can be very stable against strong levels of Rosenbaum confounding encoded as boolean confounders. We also notice that even with weak covariates and strong Rosenbaum confounding, the estimated TET can still provide reasonably good ITE estimations.

Finally, we presented a preliminary analysis of the VPS data to attempt to solve the ventilator triage problem.

In conclusion, we propose our model for solving the ITE estimation problems by learning the TET based on estimated outcome matching, given strong covariates. The estimated TET shows to be stable against strong Rosenbaum confounding encoded as boolean confounders.

Chapter 6

Discussions and Future Plans

In this chapter, we discuss some of the topics that we think are interesting to both the TET estimation and its sensitivity analysis, we also propose our future plans based on these topics.

6.1 Estimated Outcome Matching

As shown in the experiment, estimating TET based on estimated outcome matching turned out to perform quite well, and we have been thinking about why potential outcome — more specifically, the predictions from potential outcome without treatment decision tree — is the right metric to match on.

One explanation can be: ITE is defined as the differences between two potential outcomes. Ideally, the TET should group individuals sharing the similar ITE together as a leaf node. However, this means the ideal TET should ignore these two potential outcomes, but only focus on their differences. To achieve this, one approach can be finding reliable method to estimate their two potential outcomes, which is the idea behind matching.

Estimated outcome matching, on the other hand, is based on the assumption that two individuals sharing a similar outcome without treatment should be matched with each other. This can be interpreted as assuming individuals sharing the similar outcome without treatment are expected to have similar ITE, while at first glance it does not sound quite promising. However, if we reconsider what estimated outcome matching is trying to do, it is actually trying to estimate the averaged treatment effect on the treated within each subgroup divided by potential outcomes, that is, estimated outcome matching's matched pairs can only be regarded as meaningful for subgroup ATT estimation, and this is why we choose to use decision tree as our potential outcome estimator, for it performs well in grouping individuals.

So, estimated outcome matching with replacement is not exactly predicting the ITE, instead, it tries to predict one ATT for each subgroup. Then why this can be used to represent the real TET learned with the true ITE of each individual?

Considering the fact, all matched pairs' treated individual covariates are labeled and used as training data for the decision tree, one can expected that the true ITE is different from the label we attach to each treated individual covariates, such that grouping them together as a subgroup based on outcome without treatment is essentially including several individuals that does belong to this ATE subgroup, but not sharing the same ITE at all. One natural way to look at this problem is that these "noisy" individuals can be categorized into even smaller subgroups inside the subgroup defined by similar potential outcome without treatment, which is the leaf node of potential outcome tree. It is also a reasonable assumption that these even smaller subgroups can be distinguished by individual's covariates, or there must be confounding involved in the data.

Estimated outcome matching assumes further that these smaller subgroups can be distinguished by the treated individual's covariates, such that the TET learning algorithm can categorize these "noisy" individuals out of the rest of individuals in the potential outcome tree leaf node by splitting on covariates that distinguish these "noisy" individuals from the rest of individuals. From this point of view, estimated outcome matching using a decision tree can be regarded as conducting exact matching on those covariates influential to outcomes.

Instead of matching on just one potential outcomes, we plan to experiment on matching on both of two potential outcomes, we would like to see the estimated TETs using these two different matching metrics based on more complex synthetic data and models, to see if matching on two potential outcomes performs better than just matching on potential outcome without treatment.

6.2 U with Rosenbaum Confounding

Another interesting observation from our experiment is the boolean confounder sometimes does not change the structure of estimated TET, while its *probability* used as a confounder changes the structure significantly as shown in Appendix C.

As we explained, U and its conditional probability PU have totally different effects on ITE estimation: PU can be regarded as an extreme confounding while U is not. This makes us think these two confoundings fall into different categories of confounding, and we question if Rosenbaum confounding is precise enough to describe and control confounding in this scenario, or whether we should consider other mechanisms.

For this problem, we plan to experiment further with the boolean confounder generation. We hope to find a method of generation such that we are able to control the generated boolean confounder to have exactly the same level of confounding to TET estimation as if we used PU as a variable.

APPENDICES

Appendix A

Sensitivity Analysis Experiment Diagrams

Here we show a diagram of the sensitivity analysis on both real and estimated TET with and without confounding A.1.

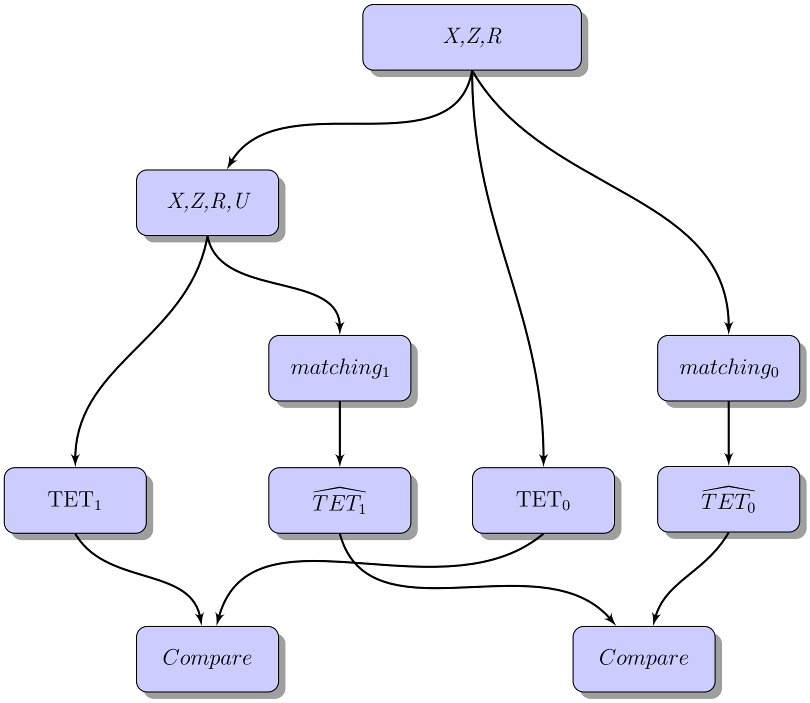


Figure A.1: Comparing real TET and estimate TET in sensitivity analysis.

Appendix B

Synthetic Data Generation Pseudocode

In this appendix, we show the pseudocode for generating synthetic data that has been used in our experiment.

B.1 Covariates Generation

For covariates X_0, X_1, X_2, and X_3 are simply drawn from either a Bernoulli distribution or the standard normal distribution $N(0, 1)$, here we focus on the generation of X_4, X_5, X_6, and X_7.

B.1.1 X_4, X_5

With covariates from X_0 to X_3 generated, X_4 and X_5 can be generated according to following procedures:

```
for each individual i do
    Get $x_2$, $x_3$ from covariates of $i$;
    if $x_3$ >1/3 then
        $x_4$ is drawn from Bernoulli distribution with probability 0.1;
        $x_5$ is drawn from Bernoulli distribution with probability 0.1;
    else
        if $x_2$ <0.7 then
            $x_4$ is drawn from Bernoulli distribution with probability 0.25;
            $x_5$ is drawn from Bernoulli distribution with probability 0.25;
        else
            $x_4$ is drawn from Bernoulli distribution with probability 0.95;
            $x_5$ is drawn from Bernoulli distribution with probability 0.95;
        end
    end
end
```

Algorithm 3: Generating X_4 and X_5

B.1.2 X_6, X_7

With covariates from X_0 to X_5 generated, X_6 and X_7 can be generated according to following procedures:

```
for each individual i do
    Get $x_2$, $x_3$, $x_4$, $x_5$ from covariates of $i$;
    $x_6 \leftarrow (4^{x_3} + x_2 \times random(0,1))/5$;
    $x_7 \leftarrow (x_3 \times 7^{x_3} + (1 - x_2) \times random(0,1))/8$;
end
```

Algorithm 4: Generating X_6 and X_7

B.2 "Strong" Setting Outcomes

With all covariates from X_0 to X_7 generated, we use two simple decision tree models to generate the two potential outcomes with treatment R^1 and without treatment R^0 for each individual.

We also design these two separate tree models with different covariates and cutpoint values as splitting nodes, such that the R^0 tree and R^1 tree have different structures, which guarantees that the real TET's structure is not exactly the same as the R^0 or R^1 tree. In fact, the real TET's structure can only be learned from training another new decision tree taking each individual's covariates labeled with ITE. This essentially increases the difficulty of TET estimation.

The models in this setting are more deterministic compared with models in the "Weak" setting.

B.2.1 R^0

Given covariates from X_0 to X_7, the potential outcome without treatment can be generated through the decision tree model shown in Figure B.3:

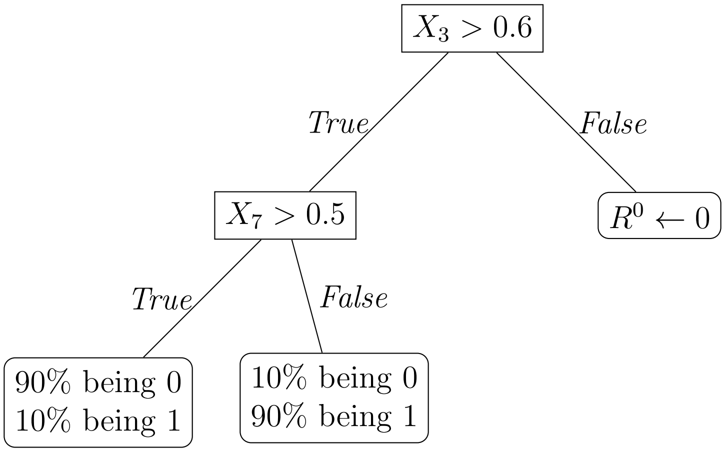


Figure B.1: R^0 Generator

B.2.2 R^1

Given covariates from X_0 to X_7, the potential outcome with treatment can be generated through the decision tree model shown in Figure B.4:

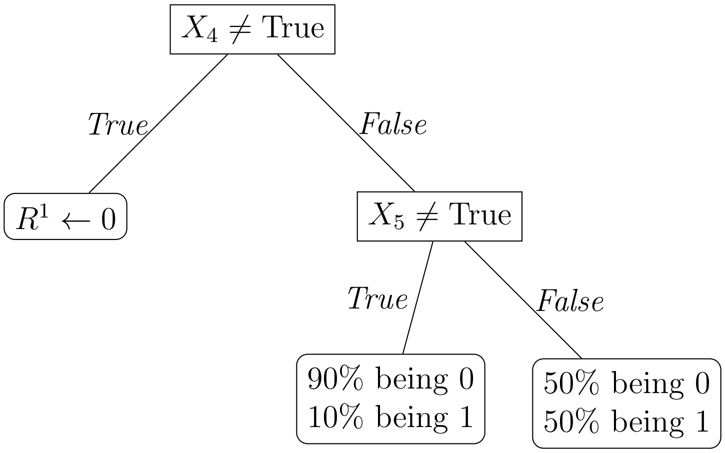


Figure B.2: R^1 Generator

B.3 "Weak" Setting Outcomes

We take the same decision tree structures from the "Strong" setting to design models generating outcomes with this "Weak" setting. We change the probability inside each leaf node, such that these two models now are less deterministic compared with models in the "Strong" setting.

B.3.1 R^0

Given covariates from X_0 to X_7, the potential outcome without treatment can be generated through the decision tree model shown in Figure B.3:

B.3.2 R^1

Given covariates from X_0 to X_7, the potential outcome with treatment can be generated through the decision tree model shown in Figure B.4:

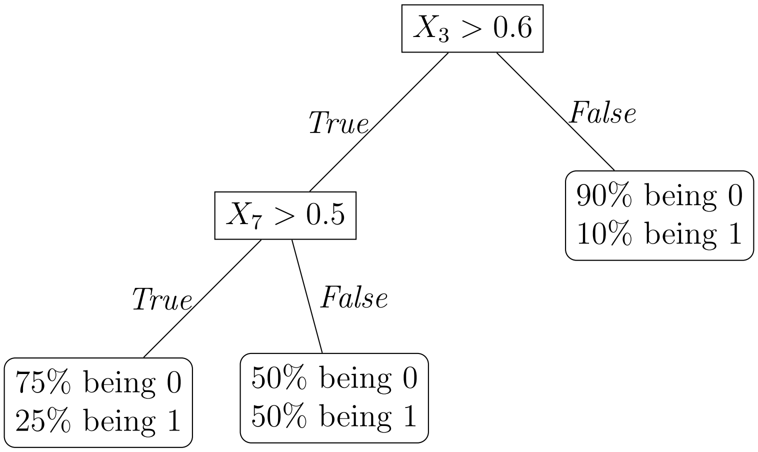


Figure B.3: R^0 Generator

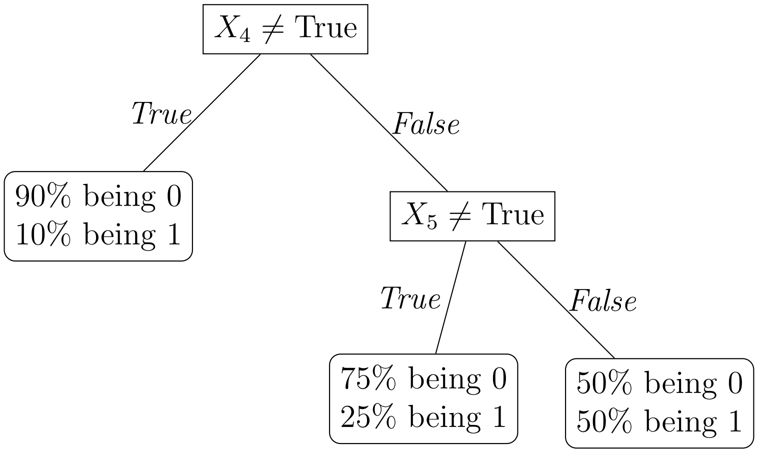


Figure B.4: R^1 Generator

B.4 Treatment Generations

With all covariates from X_0 to X_7 generated, we design the treatment assignment model as a simple decision tree model. We also design this treatment assignment tree model with covariates that do not show up in any of these two outcome trees. This is to make sure that we are not able to predict outcome simply given treatment, or vice versa.

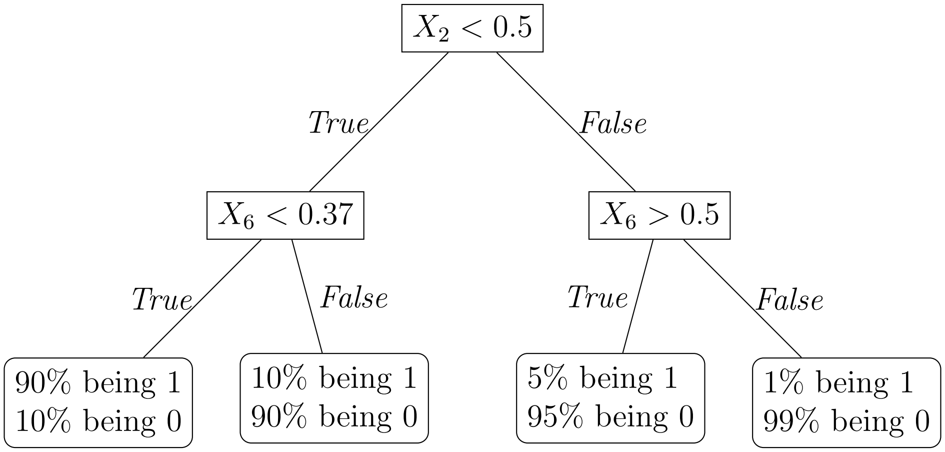


Figure B.5: Z Generator

Appendix C

Extreme Confounding with PU

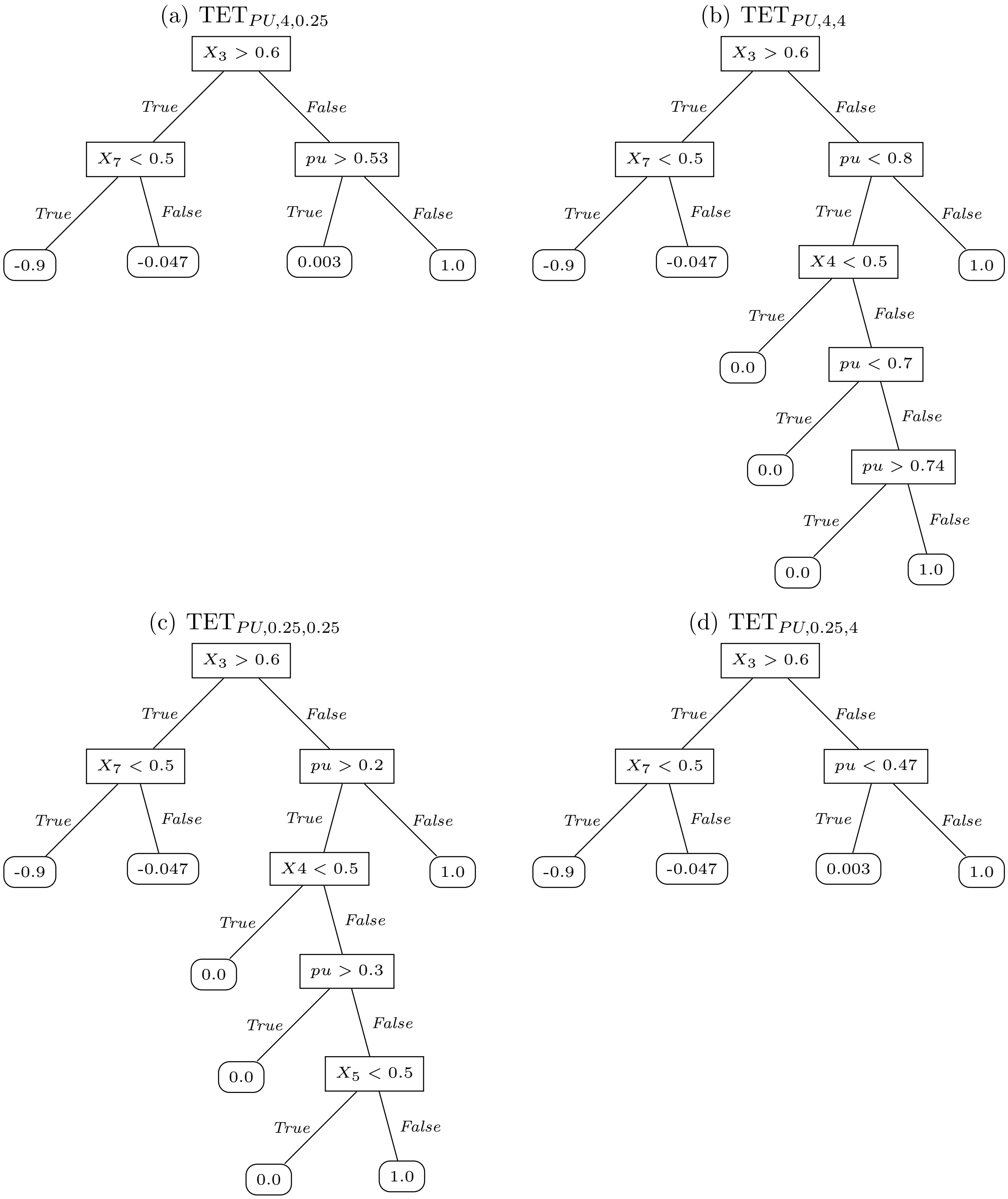
In this Chapter, we introduce an extreme confounding into our synthetic data: the conditional probability of the boolean confounder U being 1 during its generation process. We also generated four other datasets that include the conditional probability of confounder U being 1 encoded as a continuous confounder. We name these four datasets as $PU_{\Gamma,\Delta}$ to distinguish from datasets $U_{\Gamma,\Delta}$ with boolean confounders included.

From the experimental results, we find the estimated TETs structural change influenced by PU is very different from the influence of the boolean confounder U: all four $\mathrm{TET}_{PU,\Gamma,\Delta}$ change their structures significantly compared with the real TET without confounding, and all their four estimations $\widehat{\mathrm{TET}}_{PU,\Gamma,\Delta}$ also change significantly compared with the goals they attempt to estimate. Here we show their structures side-by-side in Figure C.1 and Figure C.2.

We can easily observe that all these real TETs with confounder probabilities change their structures from the real TET without confounding shown in Figure 4.1, more specifically, the confounder probability changes the real TET structure by showing up as a splitting node in the real TET, and with such a strong level of confounding, the new confounder probability node shows up very close to the root node of each real TET.

This kind of changing the real TET can be interpreted as the confounder probability being very influential to the ITE and this influence can be captured by the real TET model, that is, confounding defined by Rosenbaum and Dual Sensitivity Analysis results in changing the structure of real TET without confounding.

Besides, we also observe that a TET tends to "mirror" the structure of another corresponding TET with the same level but different direction of confounding, more specifically,



(a) $\mathrm{TET}_{PU,4,0.25}$

(b) $\mathrm{TET}_{PU,4,4}$

(c) $\mathrm{TET}_{PU,0.25,0.25}$

(d) $\mathrm{TET}_{PU,0.25,4}$

Figure C.1: Comparison of real $\mathrm{TET}_{PU,\Gamma,\Delta}$

$\text{TET}_{PU,\Gamma,\Delta}$ tends the have a very similar structure with $\text{TET}_{PU,1/\Gamma,1/\Delta}$, but with the PU conditions reversed. This can be explained as the confounder having exactly the opposite effect to treatment and outcome, and it turns out to have the exact opposite effect to ITE as well.

And we find that though structures of these real TETs change from the real TET without confounding, their error-rates are almost the same as the error-rate of real TET without confounding, here we show a table of error-rate comparison in Table C.1.

Given the fact that, as a variable very influential to treatment Z and outcome R and included in our data for matching, this PU is equivalent as a covariate containing more information that can help us to estimate ITE better. This error-rate comparison between the real TET with and without confounding further also confirms our previous statement: TET is the correct model to describe the ITE.

Table C.1: Comparison between $\text{TET}_{PU,\Gamma,\Delta}$ error-rates

Γ Δ	0.25	4
0.25	4.959%	4.956%
4	4.956%	4.959%

When looking at the four estimated TETs with confounding encoded as PU, all four $\widehat{\text{TET}}_{PU,\Gamma,\Delta}$ dramatically change their structures compared with the corresponding real TET with confounding that they attempt to estimate through estimated outcome matching, here we show these four estimated TET with confounding side-by-side in Figure C.2.

As we can easily observe, the confounder probability PU has a really strong influence to all the TETs estimated through estimated outcome matching, in each of these trees, PU shows up as a splitting node, and significantly changes the estimated TET structure far from the structure of real TET with confounding as shown in Figure C.1. Each of the estimated TET can be regarded as totally different from the real TET it attempts to estimate.

Given the fact that the boolean confounder U and this conditional probability PU are both derived from the same level of Rosenbaum confounding, since they are all generated according to the same Γ and Δ, then why do these estimated TETs based on U and PU end up in totally different quality?

We propose the following explanation: though U and PU are both derived from the same level of Rosenbam confounding, they actually have different kinds of effect on estimate outcome matching, or any statistical treatment effect estimation based on comparing metrics between treated and control. The probability PU is computed such that the boolean confounder U distributed according to PU is expected to have certain Rosenbaum confounding, PU itself can be a perfect predictor of treatment and outcomes at the same time. If we included PU as a covariate, we can perfectly predict treatment and outcomes of each individual by just looking at PU alone. This means covariate PU itself is easily splitting the whole population into two groups regarding treatment, such that inside each group the distribution of PU will be totally different, in fact, we can say that there will not be any "overlapping" part between treated and control groups in the covariate space, and this indicates that any comparison between these two groups will hardly provide meaningful information regarding treatment effect. This will eventually make any current matching strategy useless for treatment effect estimation.

On the contrary, boolean confounder U is converted from the probability PU through randomization, this means even though the distribution of U is strongly influential to treatment and outcome, we may have reasonable estimation of treatment of outcome based on U, but it would be almost impossible for us to perfectly predict treatment and outcome perfectly from the boolean covariate U. That means, if we include the boolean confounder U as a covariate, the treated and control group still share a overlapping part between each other, with this overlapping part we are able to estimate treatment effect by cleverly comparing the population or sub-populations from these two groups, such as matching.

To confirm this idea, here we show the comparison between the original estimated outcome decision tree that was learned from data set without confounding or with boolean confounder, as well as the real potential outcome model that has been used in R^0 generation in Figure C.3. By comparing these two decision tree models, we observe that the $\widehat{R^0}$ did a very good job of estimating the potential outcome without treatment even with the boolean confounder U included in the data, this good estimation promises a good matching scalar for TET estimation. And this can also explain why estimated TET with estimated outcome matching performs well in situations of boolean confounding.

We also show the comparison among the estimate potential outcome trees with different levels of confounding probabilities included in the data in Figure C.4. As we can observe the confounder probability PU perfectly predicts the outcome without treatment in all of these four models, leaving no space for matching or any other treatment effect estimation at all. This property of PU eventually made matching on estimated outcome not providing any meaningful information for treatment estimation, which made these estimated TET in Figure C.2 not only very different from the real TET in structure, but also extremely poor in predictive power as shown in Table C.2.

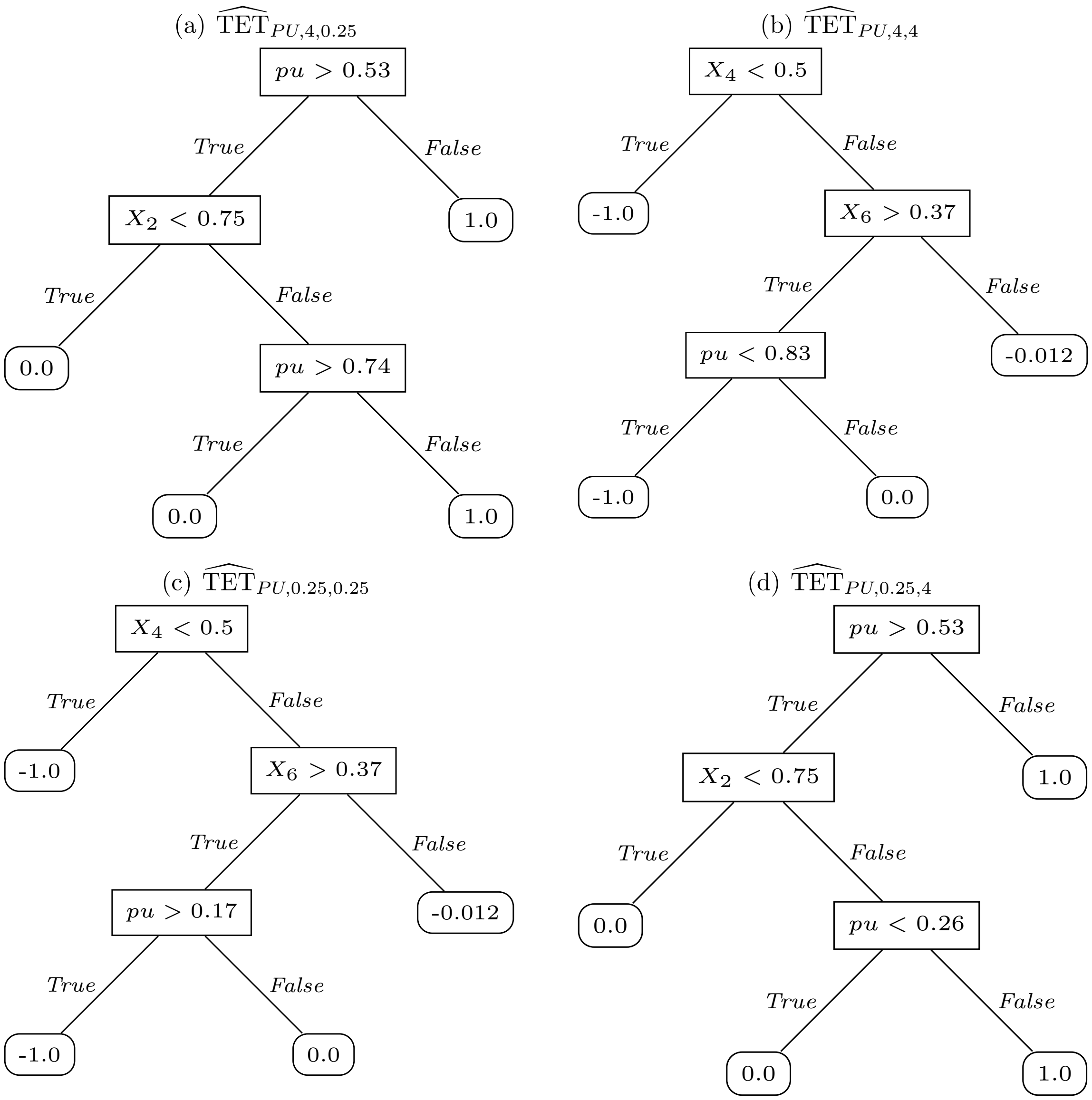
Table C.2: Comparison between $\widehat{\mathrm{TET}}_{PU,\Gamma,\Delta}$ error-rates

Δ \ Γ	0.25	4
0.25	38.175%	49.745%
4	49.745%	38.175%

Table C.3: Comparison between error-rates $\widehat{\mathrm{TET}}_{U,\Gamma,\Delta}$ with "Strong" setting data

Δ \ Γ	0.25	4
0.25	4.991%	4.991%
4	4.991%	4.991%

Note that this does not indicate that estimating TET based on estimated outcome matching is bad with continuous value confounder, by including PU we actually introduced the "worst" confounding that could ever possibly happen to observational study, we argue that with this kind of confounding, no treatment effect estimation strategy would ever provide any reasonable estimation.



(a) $\widehat{\text{TET}}_{PU,4,0.25}$

(b) $\widehat{\text{TET}}_{PU,4,4}$

(c) $\widehat{\text{TET}}_{PU,0.25,0.25}$

(d) $\widehat{\text{TET}}_{PU,0.25,4}$

Figure C.2: Comparison of $\widehat{\text{TET}}_{PU,\Gamma,\Delta}$

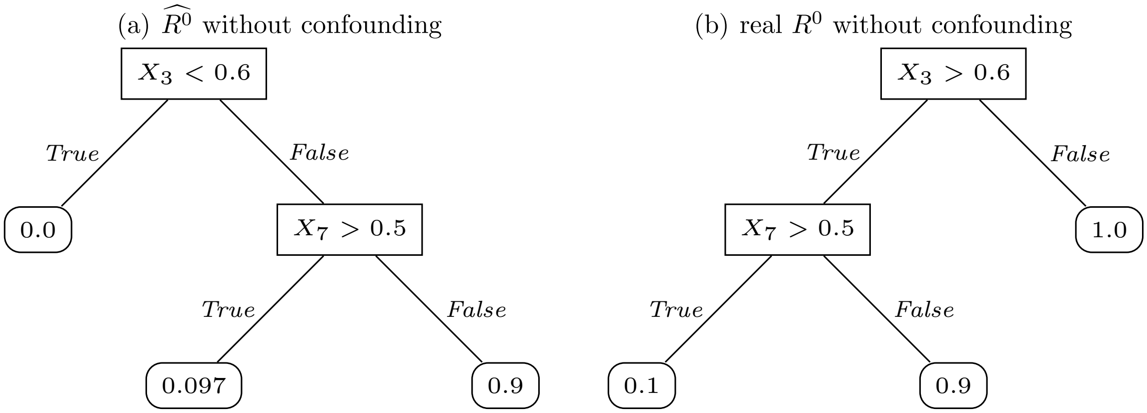
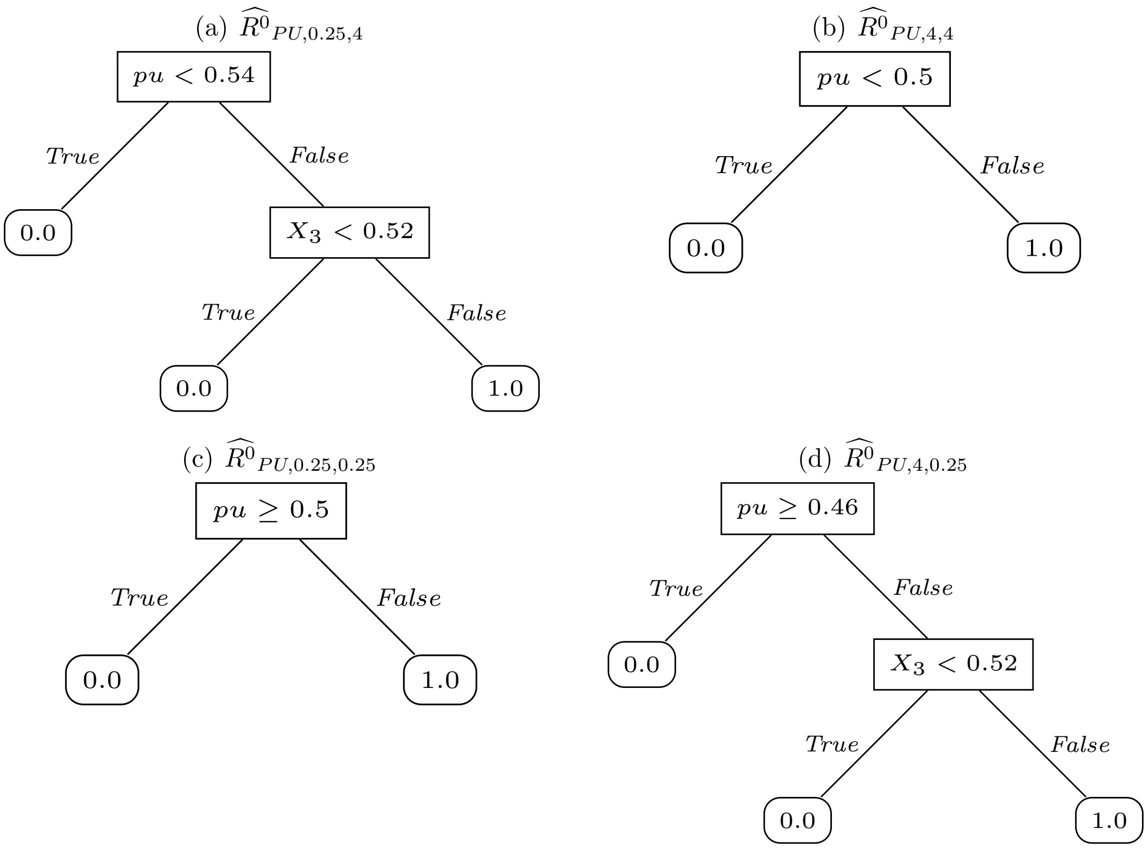


Figure C.3: Comparison of $\widehat{R^0}$ model with real R^0 model without confounding



(a) $\widehat{R^0}_{PU,0.25,4}$

(b) $\widehat{R^0}_{PU,4,4}$

(c) $\widehat{R^0}_{PU,0.25,0.25}$

(d) $\widehat{R^0}_{PU,4,0.25}$

Figure C.4: Comparison between $\widehat{R^0}_{PU,\Gamma,\Delta}$ models

Appendix D

VPS Data Description

In this Chapter, we provide a description of VPS data, including what covariates VPS data contains, and their missing rates.

Collected by VPS and CHLA together, the VPS data is a 58,772 × 142 matrix with following types of variables:

- Unique numeric episode identifier episode ID in range $1 \cdots 58772$.
- 3 patient characteristics (*age*, *weight*, *gender*).
- 9 covariates related to patient origin or reason for coming.
- 15 binary covariates related to diagnoses collected for PIM [25] score.
- 7 clinical observation covariates collected at the time of admission.
- 9 binary covariates related to diagnoses collected for PRISM [13] score.
- 46 clinical observation covariates collected after 12 hours for PRISM score.
- 44 medical procedures the patient has undergone (e.g. *ventilation*).
- 6 covariates related to resources utilization.
- 3 values related to outcome (POPC[1], PCPC[2], and *mortality*).

[1] Pediatric Overall Performance Category.

[2] Pediatric Cerebral Performance Category.

Table D.1: Covariates missing rates in VPS data

Covariate	Treated NA rates	Control NA rate
maxPh	24%	77%
minPh	34%	83%
maxPCO$_2$	24%	77%
minPCO$_2$	34%	83%
WorstComaStatus	85%	76%
...	...	...

Table D.2: Outcomes missing rates in VPS data

Outcome	Treated NA rates	Control NA rate
POPC	77%	90%
PCPC	85%	97%

A great number of those covariates and outcomes have a high missing rate, here we show some of them in Table D and Table D. Especially the high missing rates of outcome *POPC* and *PCPC*, which leaves only the *mortality* data for us to use as outcome in TET estimation.

References

[1] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer, 1998.

[2] Peter C Austin. A critical appraisal of propensity-score matching in the medical literature between 1996 and 2003. *Statistics in medicine*, 27(12):2037–2049, 2008.

[3] Bénédicte Briand, Gilles R Ducharme, Vanessa Parache, and Catherine Mercat-Rommens. A similarity measure to assess the stability of classification trees. *Computational Statistics & Data Analysis*, 53(4):1208–1217, 2009.

[4] Felix Dannegger. Tree stability diagnostics and some remedies for instability. *Statistics in medicine*, 19(4):475–491, 2000.

[5] Joseph L Gastwirth, Abba M Krieger, and Paul R Rosenbaum. Dual and simultaneous sensitivity analysis for matched pairs. *Biometrika*, 85(4):907–920, 1998.

[6] Xing Sam Gu and Paul R Rosenbaum. Comparison of multivariate matching methods: Structures, distances, and algorithms. *Journal of Computational and Graphical Statistics*, 2(4):405–420, 1993.

[7] David W Hosmer Jr and Stanley Lemeshow. *Applied logistic regression*. John Wiley & Sons, 2004.

[8] Jyotsna Jalan and Martin Ravallion. Estimating the benefit incidence of an antipoverty program by propensity-score matching. *Journal of Business & Economic Statistics*, 21(1):19–30, 2003.

[9] Michael Lechner. Program heterogeneity and propensity score matching: An application to the evaluation of active labor market policies. *Review of Economics and Statistics*, 84(2):205–220, 2002.

[10] Peter McCullagh and John A Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.

[11] Tom M Mitchell. *Machine learning*. McGraw-Hill, Boston, MA, 1997.

[12] John Neter, Michael H Kutner, Christopher J Nachtsheim, and William Wasserman. *Applied linear statistical models*, volume 4. Irwin Chicago, 1996.

[13] Murray M Pollack, Kantilal M Patel, and Urs E Ruttimann. Prism III: an updated pediatric risk of mortality score. *Critical care medicine*, 24(5):743–752, 1996.

[14] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[15] June Machover Reinisch, Stephanie A Sanders, Erik Lykke Mortensen, and Donald B Rubin. In utero exposure to phenobarbital and intelligence deficits in adult men. *JAMA*, 274(19):1518–1525, 1995.

[16] Paul R Rosenbaum. From association to causation in observational studies: The role of tests of strongly ignorable treatment assignment. *Journal of the American Statistical Association*, 79(385):41–48, 1984.

[17] Paul R Rosenbaum. Sensitivity analysis for certain permutation inferences in matched observational studies. *Biometrika*, 74(1):13–26, 1987.

[18] Paul R Rosenbaum. *Observational studies*. Springer, 2002.

[19] Paul R Rosenbaum and Donald B Rubin. Assessing sensitivity to an unobserved binary covariate in an observational study with binary outcome. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 212–218, 1983.

[20] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[21] Paul R Rosenbaum and Donald B Rubin. Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American Statistical Association*, 79(387):516–524, 1984.

[22] Donald B Rubin. Matching to remove bias in observational studies. *Biometrics*, pages 159–183, 1973.

[23] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688, 1974.

[24] Jasjeet S Sekhon. Multivariate and propensity score matching software with automated balance optimization: the matching package for R. *Journal of Statistical Software*, 42:1–52, 2011.

[25] Anthony Slater, Frank Shann, and Gale Pearson. Pim2: a revised version of the paediatric index of mortality. *Intensive Care Medicine*, 29(2):278–285, 2003.

[26] R-Core Team et al. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2012.