Ludwig-Maximilians-Universität München
Institut für Statistik

Comparing Methods to Estimate Personalised Treatment Effects from Observational Data

Master's Thesis

Author: Daniela Buchwald

Supervision: Dr. Heidi Seibold

Date: April 17, 2019

Abstract

The main objective of a clinical trial is to estimate the effect of a new treatment compared to standard therapy or placebo. In personalised medicine, the differences of treatment effects in distinct population subgroups or even individual patients are quantified. However, in some cases only observational data is available. Thus, confounders can lead to biased estimates. One way to consider confounding is adjusting standard methods by the propensity score, i.e. the probability of receiving a treatment given the covariates. In order to estimate personalised treatment effects, multiple tree-based and regression spline-based methods can be applied.

The aim of this thesis is to assess and compare the performance of such methods. For this, a simulation study is conducted. Eight different datasets are generated with three different numbers of observations, respectively. The results are evaluated by considering bias and root mean squared error (RMSE).

According to the simulation study, an adjustment for confounding reduces bias in almost all methods if strong confounding is present. The bayesian additive regression trees (BART) method shows good performance even without adjustment. Especially for stepwise treatment effect functions, it is superior to other methods. PTO forest, causal forest and GLM trees with inverse probability of treatment weighting (IPTW) are well performing and computationally less demanding alternatives. Causal MARS performs well for linear treatment effect functions, even for a small number of observations. However, it has a long running time. For a large sample size, BART is a competitive method.

Contents

List of Figures

List of Tables

List of Algorithms

Notation

A_m	Associated set of leaf node parameters of the mth binary tree $\{\mathcal{B}_b^{BA}\}$
$b_d(x)$	dth basis function in MARS model
$\{\mathcal{B}_b^{BA}\}_{b=1,...,B}$	Binary tree for sum-of-trees in BART model with B leaf nodes
$\{\mathcal{B}_b^{CT}\}_{b=1,...,B}$	Causal tree with B leaf nodes
$\{\mathcal{B}_b^{MOB}\}_{b=1,...,B}$	Model-based tree with B leaf nodes
C	Confounder
C_1/C_2	Child nodes in causal forest with n_{C_1}/n_{C_2} number of observations
c	Element of covariate matrix ($c \in X_{ij}$)
$e(x)$	Propensity score (conditional treatment probability)
G	Random forest model
$g()/h()$	Link and response function
$\ell(x, \{\mathcal{B}_b\})$	A leaf $\ell \in \{\mathcal{B}_b\}$ such that $x \in \ell$
$\mathcal{M}()$	Parametric model in a model-based tree
N^{te}	Number of observations in a test sample
N^{tr}	Number of observations in a training sample
O_i	Quantity that contains outcome and treatment assignment ($\{Y_i, T_i\}$) for observation i
P	Parent node in causal forest with n_P number of observations
p	Marginal treatment probability
$\mathcal{S}$	Data sample
$\mathcal{S}^{est}$ / $\mathcal{S}^{te}$ / $\mathcal{S}^{tr}$	Estimation sample / Test sample / Training sample
$\mathcal{S}_{control}$ / $\mathcal{S}_{treat}$	Subsample for control units / treated units
T_i	Binary indicator for the treatment for observation i
$\tilde{T}_i$	Centered treatment of observation i
w	Weights achieved by inverse probability of treatment weighting
X_{ij}	jth observed covariate / partitioning variable for observation i
Y_i	Observed binary outcome for observation i
Y_i^0	Potential outcome without treatment for observation i
Y_i^1	Potential outcome with treatment for observation i
Y_i^*	Transformed outcome of observation i
$\tilde{Y}_i$	Centered outcome of observation i

α	Parameter that controls depth of causal trees
$\alpha_i(x)$	Weight in causal forest that measures relevance of observation i for a specific x
β_d	dth coefficient in MARS model
ϵ	Residual
η_i	Linear predictor of observation i
θ	Vector of parameters fitted in a model-based tree (μ and τ^*)
$\theta(x)$	Quantity of interest in generalised random forests, here: treatment effect
$\mu(x)$	Mean effect
$\mu(t, x)$	Conditional mean function for treatment t
π_i	(Conditional) probability of observing $y_i = 1$ for observation i
ρ	Pseudo-outcome in causal forest
σ^2	Variance of residuals
τ	Average treatment effect
$\tau(x)$	Conditional average treatment effect
$\tau^*(x)$	Treatment effect expressed as log odds ratio
Φ	Cumulative distribution function of a standard normal distribution
Ψ	Objective function in model-based trees
ψ	Score function

Running Indices

$b = 1, \ldots, B$	Number of leaf nodes in a partition $\mathcal{B}$
$d = 1, \ldots, D$	Number of basis functions in MARS
$i = 1, \ldots, n$	Number of observations
$j = 1, \ldots, p$	Number of covariates / partitioning variables
$m = 1, \ldots, M$	Number of trees in a forest
$s = 1, \ldots, S$	Number of propensity strata

Abbreviations

AIPTW	**A**ugmented **I**nverse **P**robability of **T**reatment **W**eighting
ATE	**A**verage **T**reatment **E**ffect
BART	**B**ayesian **A**dditive **R**egression **T**rees
CART	**C**lassification **A**nd **R**egression **T**ree
CATE	**C**onditional **A**verage **T**reatment **E**ffect
(C)TMLE	(**C**ollaborative) **T**argeted **M**aximum **L**ikelihood **E**stimation
DAG	**D**irected **A**cyclic **G**raph
GLM	**G**eneralised **L**inear **M**odel
GRF	**G**eneralised **R**andom **F**orest
IPTW	**I**nverse **P**robability of **T**reatment **W**eighting
LASSO	**L**east **A**bsolute **S**hrinkage and **S**election **O**perator
MARS	**M**ultivariate **A**daptive **R**egression **S**plines
ML	**M**aximum **L**ikelihood
MOB Tree	**Mo**del-**B**ased Tree
OLS	**O**rdinary **L**east **S**quares
OOB	**O**ut-**O**f-**B**ag
PTO Forest	**P**ollinated **T**ransformed **O**utcome Forest
RCT	**R**andomised **C**ontrolled **T**rial
(R)MSE	(**R**oot) **M**ean **S**quared **E**rror
SUTVA	**S**table **U**nit **T**reatment **V**alue **A**ssumption
SVM	**S**upport **V**ector **M**achine

1 Introduction

Demonstrating superiority of a new treatment over placebo or standard of care is the aim of several clinical studies. For this, the effect of a treatment on the outcome has to be estimated.

Many statistical procedures assume a constant treatment effect. This corresponds to identical effects for all patients. In heterogeneous populations, this assumption might be incorrect. Thus, patient's characteristics influence the efficacy of treatments. In this case, estimating an average treatment effect for the whole population without distinguishing between patient subgroups is inappropriate. To circumvent this problem, methods estimating personalised treatment effects and taking heterogeneity into account are required. An old-fashioned way is to specify the subgroups in advance. Nevertheless, this makes the detection of unexpected treatment heterogeneity infeasible.

Furthermore, in most cases the treatment is considered to be randomly assigned. That means patients in the population are randomly allocated to the treatment or control group. However, in some situations this might not be realistic or ethical and the treatment assignment cannot be randomised.

In the easiest case, random allocation to treatment groups and an equal treatment effect can be assumed for all patients. For this, estimation of an average treatment effect without considering confounders is sufficient. But if these two conditions are not fulfilled, this can lead to biased results. Hence, there is a growing interest in using methods that can handle both situations, i.e. heterogeneous treatment effects as well as non-randomised datasets. In this thesis, several of these methods are presented and evaluated.

This thesis is organised as follows: In Chapter 3, the theory and challenges of estimating causal effects are explained. Afterwards, the role of the propensity score in observational data is presented in Chapter 4. Then, the methods which are going to be compared are described in Chapter 5. In Chapter 6, a simulation study to evaluate these methods is presented. Additionally, this chapter provides some further analyses on one specific method, the IPTW weighted GLM tree. Finally, the results are discussed and suggestions for further research are given in Chapter 7. As an introduction, related work on estimating personalised treatment effects from observational data is summarised in the following chapter.

2 Related Work

Early work of heterogeneous treatment effect estimation simply compared predefined subpopulations. In previous years, many methods have been developed to estimate heterogeneous treatment effects without defining subgroups in advance.

New techniques for heterogeneous treatment effect estimation adapt standard machine learning methods. They can flexibly estimate and handle a potentially large number of covariates. One of those methods are classification and regression trees (CART), e.g. interaction trees by Su *et al.* (2009). Furthermore, causal trees by Athey & Imbens (2015) and GLM trees by Zeileis *et al.* (2008) are based on the idea of CART and used in the present thesis. Based on regression trees also random forest methods can be applied. In this work, the causal forests by Wager & Athey (2018) and the generalised random forests by Athey *et al.* (2018) are discussed. Foster *et al.* (2011) use regression forests to estimate the effect on the outcome in treatment and control group separately. In order to receive the treatment effect, they consider the corresponding difference. "Virtual twins" and "counterfactual random forests" base on the random forest algorithm as well. Virtual twins are based on the idea of estimating counterfactual outcomes. For this, a virtual twin for each observation i is created. A virtual twin complies with a datapoint which is similar to the original datapoint with respect to all covariates. However, the treatment T_i is replaced with the counterfactual outcome $1 - T_i$. The counterfactual outcome is obtained by running the datapoint down a forest, which was created based on the whole dataset. Additionally, it is possible to improve the virtual twin approach by manually including treatment interactions in the design matrix. This method is called virtual twins interaction. Counterfactual random forests are an extension of virtual twins interaction. In this method, separate forests are fitted for each treatment group rather than one single forest. In the next step each observation is run down its natural forest as well as its counterfactual forest. This leads to the counterfactual treatment effect estimate. The counterfactual forests can be extended by replacing the Breiman forests by synthetic forests, developed by Ishwaran & Malley (2014). In this method, forests which use the original features and synthetic features are combined. Multiple Breiman forests ("baselearners") (Breiman 2001) are grown with different values of the tuning parameters *mtry* and *nodesize*. The parameter *mtry* indicates the number of variables randomly sampled as candidates at each split, *nodesize* specifies the minimum size of terminal nodes. Each forest generates a predicted value which

complies with the synthetic feature. The synthetic forest results by including the new input synthetic features as well as all original features. Xie *et al.* (2012) model the treatment effect as a function of the propensity score, using one parametric and two non-parametric methods.

Other machine learning methods to estimate heterogeneous treatment effects are the least absolute shrinkage and selection operator (LASSO) (Qian & Murphy 2011, Tian *et al.* (2014)), support vector machines (SVM) (Imai & Ratkovic 2013, Zhao *et al.* (2012)), boosting (Powers *et al.* 2018) and neural nets (Schwab *et al.* 2018). Knaus *et al.* (2017) combine non-experimental causal empirical models with lasso-type estimators. There are also bayesian machine learning methods. One of them is called "bayesian additive regression trees" (BART) (Hill 2011) and is further applied in this thesis.

Zhang *et al.* (2012a) and Zhang *et al.* (2012b) provide algorithms that can handle randomised as well as observational datasets. Haoda Fu (2016) develops a method which can deal with even more than two treatment groups for randomised and for observational studies. Nevertheless, this algorithm is limited to a small number of covariates.

Most methods are based on randomised controlled trials (RCTs). Nowadays, there is a growing need for personalised medicine solutions that handle non-randomised datasets. Working with non-randomised datasets is complicated due to confounding. Methods that are robust to confounding incorporate for example propensity scores, G-formula or targeted-maximum-likelihood estimation (TMLE). The TLME was developed by Luque-Fernandez *et al.* (2018) and has the advantage that it is a double robust method. That means either the outcome or the exposure model has to be correctly specified. Other double robust methods are the augmented inverse probability of treatment weighting (AIPTW) and the collaborative TMLE (CTMLE) (Lendle *et al.* 2013). Some methods can be either used as a G-computation approach, such as BART (Hill 2011), or they can be adjusted by the propensity score. The propensity score adjustment is further analysed in this thesis for a selection of methods.

3 Causal Effects

3.1 Potential Outcomes Framework and Average Treatment Effects

Imagine a study where the researcher is interested in estimating the effect of a treatment T on an outcome Y. In this study the treatment is assumed to be binary:

$$T = \begin{cases} 1 & \text{if a treatment was given (treatment)} \\ 0 & \text{if a placebo was given (control)} \end{cases}$$

In general, there might be more than two treatment groups.

The outcome could be either categorical or numeric (e.g. the time of survival). On this occasion the outcome is considered to be binary. A positive outcome (e.g. patient survived) corresponds to $Y = 1$ and a negative outcome (e.g. patient died) to $Y = 0$.

The treatment effect can be illustrated by a directed acyclic graph (DAG) where the arrow symbols the direction of the causal effect:

$$\text{Treatment T} \longrightarrow \text{Outcome Y}.$$

Potential or counterfactual outcomes of a person i are the outcomes we would see under each possible treatment:

$Y_i^{t=1}$: Outcome that would have been observed under treatment value $t = 1$

$Y_i^{t=0}$: Outcome that would have been observed under treatment value $t = 0$.

This means, each person has two potential outcomes. However, only one outcome is observed, the other is counterfactual. The observed outcome of individual i is defined by

$$Y_i = T_i \cdot Y_i^1 + (1 - T_i) \cdot Y_i^0.$$

This leads to

$$Y_i = \begin{cases} Y_i^1 & \text{for } T_i = 1 \\ Y_i^0 & \text{for } T_i = 0. \end{cases}$$

Besides the treatment assignment T_i and the observed outcome Y_i, a vector X_i is observed for each person. It contains the baseline covariates.

The effect of the treatment is generally defined as the difference between the two

potential outcomes $Y^1 - Y^0$. In nonheterogeneous treatment settings a common causal estimand of interest is the population average treatment effect (ATE). It is defined by

$$\tau = \mathbb{E}[Y_i^1 - Y_i^0] = \mathbb{E}[\mathbb{E}[Y_i^1|X_i] - \mathbb{E}[Y_i^0|X_i]] = \mathbb{E}(\tau(X_i)). \tag{1}$$

Consequently, it compares the outcome if people were treated with $T = 1$ to the outcome if the same people were treated with $T = 0$ (Hernan & Robins 2018).

However, it is not always correct to assume that all individuals in a population have the same treatment effect. The treatment effect could be heterogeneous and might differ between subpopulations. Thus, some medications might only be effective for a specific group of patients. Hence, it is important to figure out the relevant covariates. According to Abrahams (2008) (p. 11): "The right drug for the right patient at the right time is the mantra of personalized medicine". It is a new step to bring health care to a higher level of effectiveness and safety. Consequently, the main interest is not in estimating the average treatment effect of the whole population, but rather in estimating the individual (or heterogeneous) treatment effects for all values of x. For this, the Conditional Average Treatment Effect (CATE)

$$\tau(x) = \mathbb{E}[Y_i^1 - Y_i^0|X_i = x]$$

is considered.

Wendling *et al.* (2018) (p. 3) describe the CATE as a "useful estimand to assess the heterogeneity in treatment effect and personalise causal inference". The marginal average treatment effect ATE is simply the expectation over CATE (see Equation 1).

In reality either Y_i^1 or Y_i^0 but not both can be observed. Consequently, the causal or treatment effect is not observed for any individual. This is called the "fundamental problem of causal inference" (Holland 1986). It is not possible to train machine learning methods on this difference and $\tau(x)$ cannot be directly estimated without further restrictions. The conditions for estimating treatment effects are not the same in randomised controlled trials (RCTs) and observational datasets. This difference is further explained in the next sections.

3.2 Randomised Controlled Trials (RCT)

In a randomised controlled trial (RCT), the respective treatment T is randomly assigned. This means, allocation is not influenced by any covariate and the distribution of X is assumend to be the same in both treatment groups. Thus, the treated subjects will not differ systematically from the untreated subjects in measured and unmeasured baseline characteristics. This leads to an unbiased estimate of the treatment effect

$$\begin{aligned}
\tau(x) &= \mathbb{E}[Y_i|T_i = 1, X_i] \\
&= \mathbb{E}[T_i \cdot Y_i^1 + (1 - T_i) \cdot Y_i^0|T_i = 1, X_i] \\
&= \mathbb{E}[T \cdot Y_i^1|T_i = 1, X_i] + \mathbb{E}[(1 - T_i) \cdot Y_i^0|T_i = 1, X_i] \\
&= \mathbb{E}[Y_i^1|T_i = 1, X_i] \\
&= \mathbb{E}[Y_i^1|X_i].
\end{aligned} \tag{2}$$

Correspondingly, the expectation of Y among people with $T = 1$ ($\mathbb{E}(Y|T = 1)$) is similar to the mean of Y if the whole population was treated with $T = 1$ ($\mathbb{E}(Y^1)$).

The last equation ($\mathbb{E}[Y_i^1|T_i = 1, X_i] = \mathbb{E}[Y_i^1|X_i]$) applies because the treatment assignment is independent of potential outcomes. Hence, the assumption of *unconfoundedness* (sometimes also called *exchangeability* or *ignorability*)

$$\{Y_i^0, Y_i^1\} \perp\!\!\!\perp T_i|X_i \tag{3}$$

holds (Austin & Stuart 2015).

In the randomised case, it is not necessary to condition on X. Due to its random allocation, the assigned treatment group is independent of any covariates. Thus, in Equation 3 X can be omitted.

With the assumption of unconfoundedness, the treatment effect can be directly estimated by comparing the outcomes of both treatment groups:

$$\mathbb{E}[Y_i^1 - Y_i^0|X_i] = \mathbb{E}[Y_i|T_i = 1, X_i] - \mathbb{E}[Y_i|T_i = 0, X_i] \tag{4}$$

(Imbens & Wooldridge 2009).

To identify causal effects, additional assumptions are required.

a) SUTVA

The first assumption is the "Stable Unit Treatment Value Assumption" (SUTVA). It ensures that there is no interaction between units (no interference). That means that the treatment applied to one unit does not effect the outcome for another unit. Furthermore, there is only one version of each treatment level. This implies that potential outcomes must be well-defined.

Sometimes this assumption is also called "no-multiple-versions-of-treatment assumption". It guarantees that the potential outcomes for each individual under each possible treatment are well-defined and take on a single value (Rubin 1980).

b) Consistency

The consistency assumption

$$Y = Y^t \text{ if } T = t \qquad \forall t$$

indicates that the potential outcome under treatment $T = t$ is equal to the observed outcome if the treatment actually received is $T = t$.

c) Positivity

Pursuant to this assumption, the treatment assignment is not deterministic for each set of values for X. This means everybody in the population should have the probability

$$\mathbb{P}(T = t|X = x) > 0 \qquad \forall t, x$$

to get treated (Hernan & Robins 2018).

3.3 Observational Studies

In some situations randomisation can be unethical or not realistic. For instance, it might be unethical to prevent potential students from going to college in order to study the effect of college attendence on future career success (Athey & Imbens 2017). In such a situation some covariates C, called confounder, influence the treatment as well as the outcome. This is illustrated in Figure 3.1. Therefore, the treatment assignment is no longer randomised.

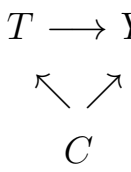

Figure 3.1: Confounding

With the presence of confounding, the arrow between T and Y is not the only connection between treatment and outcome. In addition, another path exists ($T \leftarrow C \rightarrow Y$). Thus, the confounder opens a "backdoor path" from the treatment T to the outcome Y.

In the randomised case, without the presence of confounding, the connection between T and Y is called "causation" as well as "association". The entire association of T and Y is due to the causal effect of T on Y. In the presence of confounders, there is more than one source of association. Thus, the general rule "association isn't causation" holds (Hernan & Robins 2018).

In observational datasets, Equation 2 does not apply any longer because

$$\mathbb{E}[Y_i^1|T_i = 1] \neq \mathbb{E}[Y_i^1].$$

Consequently, the treatment effect cannot be estimated like in a RCT anymore. Since the potential outcomes are typically not independent of treatment assignment, simply comparing outcomes between treatment groups as in Equation 4 is no longer feasible, i.e.

$$\mathbb{E}[Y_i^1 - Y_i^0|X_i] \neq \mathbb{E}[Y_i|T_i = 1, X_i] - \mathbb{E}[Y_i|T_i = 0, X_i].$$

This could lead to a biased estimate of the treatment effect ("selection bias"). There exists a subject to treatment selection bias where treated subjects differ systematically from control subjects. For example older people could be more likely to get $T = 1$.

Thus, confounding variables can result in biased estimates and have to be taken into account in statistical modelling.

To still be able to estimate treatment effects in observational data, the assumption of *strong ignorability* is needed. It consists of the assumption that $0 < \mathbb{P}[T = 1|X = x] < 1$ and the unconfoundedness assumption.

In accordance with the unconfoundedness assumption, no unmeasured confounders exist.

This means treatment T can be thought as being randomly assigned among people with the same values of X. Furthermore, the people in the treatment and control group are expected to be exchangeable. As defined in Equation 3, the unconfoundedness assumption can be written as

$$\{Y_i^0, Y_i^1\} \perp\!\!\!\perp T_i | X_i$$

and is already fulfilled in a RCT (Austin & Stuart 2015).

Under strong ignorability, the following equations hold:

$$\mu(1, x) = \mathbb{E}[Y_i^1 | X_i = x] = \mathbb{E}[Y_i | T_i = 1, X_i = x] \tag{5}$$

$$\mu(0, x) = \mathbb{E}[Y_i^0 | X_i = x] = \mathbb{E}[Y_i | T_i = 0, X_i = x]. \tag{6}$$

According to Hahn *et al.* (2017) the treatment effect can be estimated like in a randomised controlled trial by

$$\tau(x) = \mu(1, x) - \mu(0, x). \tag{7}$$

In the binary case, the expectation of Y is simply the probability of observing $Y = 1$. Since

$$\mu(1, x) = \mathbb{E}[Y_i | T = 1, X_i = x] = \mathbb{P}[Y_i = 1 | T_i = 1, X_i = x] \in [0, 1]$$

and

$$\mu(0, x) = \mathbb{E}[Y_i | T = 0, X_i = x] = \mathbb{P}[Y_i = 1 | T_i = 0, X_i = x] \in [0, 1]$$

the treatment effect only takes on values between -1 and 1.

4 The Propensity Score

To reduce the bias in the treatment effect estimation from observational data, the backdoor path from the treatment to the outcome needs to be eliminated. This corresponds to erasing the arrow between the confounder C and the treatment T in Figure 3.1. For this purpose, the propensity score is an appropriate method. This score is defined as the conditional probability of receiving the treatment given the observed baseline covariates or confounders.

$$e(x) = \mathbb{P}(T = 1|X).$$

Rosenbaum & Rubin (1983) demonstrate that conditional on the propensity score, treatment status is independent of measured baseline covariates. Thus, subjects in the treatment and control group with the same propensity score will have similar distributions of observed baseline covariates. This means the variables X are balanced between the two treatment groups. Therefore, the propensity score is called "balancing score". Hence, with the propensity score it is possible to create a "pseudo-randomised" sample.

Alternatively, all confounders could be accounted for by including them as covariates. Nevertheless, this might lead to the problem of over-parameterising.

Rosenbaum & Rubin (1983) present distinct methods on how to use the propensity score for observational data:

- *Covariate Adjustment*: Include the propensity score as a covariate in the model.
- *Stratification or Subclassification*: Group patients by similar propensity scores and compute the treatment effect for each group. The overall ATE is the average of these treatment effects, weighted by the overall frequency of each group.
- *Matching*: Choose pairs of patients with similar propensity scores. Discard the unmatched patients. This method is discussed in more detail in Section 4.2.
- *Inverse Probability of Treatment Weighting (IPTW)*: Assign a weight to each patient. If the patient is treated, the weight is equal to the inverse of the propensity score. If the patient is not treated, it is equal to the inverse of one minus the propensity score. Thus, underrepresented patients receive higher weights and vice versa. A propensity score close to 0 or 1 leads to large variances in the results.

This method is further explained in Section 4.1.

If the assumption of strong ignorability (see Section 3.3) holds, all these propensity score methods can produce unbiased estimates (Wendling *et al.* 2018).

Under the assumption of strong ignorability, Rosenbaum & Rubin (1983) show that given the propensity score, the treatment assignment is independent of the potential outcomes:

$$\{Y^0, Y^1\} \perp\!\!\!\perp T|X \quad \Rightarrow T \perp\!\!\!\perp \{Y^0, Y^1\}|e(x).$$

In the following, the inverse probability of treatment weighting as well as matching are further explained.

4.1 Inverse Probability of Treatment Weighting (IPTW)

Weighting people by the inverse probability of receiving treatment creates a synthetic sample. In this sample, the treatment assignment is independent of measured baseline covariates. Thus, weighting is like creating a pseudo-randomised sample.

The weights in IPTW are achieved by

$$w = \frac{T}{e(X)} + \frac{1-T}{1-e(X)}.$$

According to this, each patient's weight is equal to the inverse probability of receiving the respective treatment.

Based on these weights, the weighted estimators for the conditional mean functions are derived by

$$\mathbb{E}\left[\frac{TY}{e(X)}\middle|X = x\right] = \mathbb{E}[Y|T = 1, X = x] \text{ and}$$
$$\mathbb{E}\left[\frac{(1-T)Y}{1-e(X)}\middle|X = x\right] = \mathbb{E}[Y|T = 0, X = x].$$

These equations are the basis for the weighted propensity score estimators. This leads to the treatment effect

$$\tau(x) = \mathbb{E}\Big[\frac{TY}{e(X)} - \frac{(1-T)Y}{(1-e(X))}\Big|X = x\Big].$$

Moreover, this equation is the immediate consequence of unconfoundedness.

Due to the weighting approach, observations can be treated as in a randomised experiment.

In many applications, machine learning algorithms like boosting, neural networks or random forests are used to estimate the propensity score, which is then applied to transform the result to $\tau(x)$ (Wager & Athey 2018).

A problem of the propensity score weighting occurs if the propensity score is close to 0 or 1. This causes large weights and increases the variability of the estimated effects (Austin & Stuart 2015).

4.2 Matching

Another way of balancing data by the propensity score is matching. Here, data is preprocessed to create a pseudo-randomised sample by selectively omitting observations from the data. Afterwards, personalised treatment effects can be estimated using the preprocessed data. As before, the idea is to erase the relationship between the treatment T and the confounder covariates X. This is achieved by producing a dataset with the same distribution of the propensity score in the treatment and control group.

Ho *et al.* (2007) suggest different types of matching. These matching procedures are implemented in the `R` package `MatchIt`, which is further described in Section 6.4.

The *exact matching* technique matches each treated unit with all possible control units that have the exact same covariate values. Subclasses are formed with all units in the same subclass having the same covariate values. In a dataset with many covariates or covariates with a large number of values, an exact matching is difficult or even impossible. Then, the *subclassification* approach can be used instead. Here, a predefined number of subgroups is created. In each subgroup, the distribution of covariates in treated and control groups should be as similar as possible. In the `R` package, the default number of subclasses is 6. The propensity score is automatically estimated via logistic regression but can also be changed. In the *optimal matching* method, the respective

matched sample with the smallest average absolute distance across all the matched pairs is chosen. The *full matching* method is a type of subclassification matching which creates the subclasses in an optimal way. "Optimal" means that a weighted average of the estimated distance between each treated subject and each control subject within each subclass is minimised. The result consists of multiple matched sets, where each matched set contains one treated unit and one or more control units or vice versa.

The matching method used in this thesis is *nearest neighbour* matching. This technique selects the best control subjects matched for each individual in the treatment group. For this purpose, a distance option has to be chosen. For propensity score matching in `R`, that means that the propensity score is the distance measure, this option has to be `logit`. This means that the propensity score is estimated from a logistic regression. The order of the treated subjects can be defined by the user: from the largest to the smallest value, vice versa or randomly, as it is done in this thesis. This results in new treatment and control groups with greater overlap in their propensity scores. However, it is possible that treated and control units are matched with propensity scores relatively far apart. This is the case if, at that stage of the matching, a distant control has the shortest distance to the respective treated observation. As a result, there is no guarantee that only similar treated and control subjects are matched. To prevent this, the `caliper` feature can be used. The matched treated and control units will always be within the caliper's distance of each other. The caliper is defined as the number of standard deviations of the distance measure within which to draw control units. For this thesis, it is set to 0.2. It is also possible to combine this method with a 'mahalanobis'- metric matching with respect to a specific variable within each caliper (Ho *et al.* 2011).

4.3 Advantages and Disadvantages

Elze *et al.* (2017) compare different propensity score methods with a traditional covariate adjustment. The traditional covariate adjustment can give also good performance, but it is not suitable for small datasets with many covariates. Their results are summarised in Table 4.1.

Method	Advantages	Disadvantages
Weighting	- Retains whole dataset - Easy to implement - Can create a pseudo-randomised population with a perfect covariate balance	- Propensity score values near 0 or 1 and thus extreme weights can lead to unstable results
Matching	- Reliable - Provides good covariate balance in most circumstances - Simple to analyse and interpret - Possibility to simply present preprocessed data	- Some unmatched patients are excluded from the analyses - Less precise
Stratification	- Retains whole dataset - Possibility to detect interactions between treatment and outcome risk - Provides effect estimates for every stratum	- Performs less well in datasets with few outcomes and a large number of strata - Bad performance for strong confounding
Propensity Score as Covariate	- Good performance	- Similar to traditional covariate adjustment

Table 4.1: Advantages and disadvantages of different propensity score methods

5 Methods

Throughout the last chapters, the importance of personalised medicine, the theory and challenges of estimating causal effects, and the role of the propensity score were introduced. In this chapter, methods for estimating personalised treatment effects are described. All methods require the assumption of unconfoundedness. Details about the implementation in `R` and the corresponding packages are depicted in Section 6.4.

Most methods estimate treatment effects by taking the difference between the treatment groups, like defined in Equation 7. One way is to estimate the expected outcomes Y given the covariates X for the treatment groups separately. Afterwards the difference of the resulting values is evaluated. Another possibility is to estimate the treatment effect directly. To detect heterogeneity in the data, the algorithm searches for the biggest differences in the effects. In the present thesis, different methods for both approaches are applied.

Another way of calculating treatment effects is considering the odds ratio instead of the difference, such as in GLM trees.

To account for potential confounders, the methods are adjusted, most of them by using the propensity score. Nevertheless, the causal forest takes another approach to consider observational data.

5.1 Model-Based Recursive Partitioning (GLM Trees)

One method to estimate personalised treatment effects is model-based (MOB) recursive partitioning. In this procedure patient subgroups are identified automatically.

The approach builds on the idea of incorporating parametric models into trees: Rather than fitting one global model, local models on subsets of the data are fitted by recursive partitioning. This results in a tree in which every leaf is associated with a fitted model, e.g. a model based on maximum likelihood estimation or a linear regression model. In the case of a generalised linear model, the model is called *GLM tree*.

5.1.1 Tree Building

As a first step, the tree building phase is described. Consider a parametric model $\mathcal{M}((Y,T),\theta)$ with

$$\theta = \begin{pmatrix} \mu \\ \tau^* \end{pmatrix} \begin{array}{l} \text{intercept(s)} \\ \text{treatment effect} \end{array}$$

which is fitted to data denoted by (Y,T). Therefore, Y is the dependent variable and T the regressor.

As mentioned above, in GLM trees the treatment effect is not defined as the difference between the expected outcome in the two treatment groups, but as the log odds ratio. Therefore, the treatment effect is denoted by $\tau^*(x)$ instead of $\tau(x)$. To compare the estimation and performance with other methods, the log odds ratio is in this thesis finally converted to the difference. More details are given in Section 5.1.3.

Given n observations, the parametric model $\mathcal{M}((Y,T),\theta)$ is fitted by minimising a selected objective function

$$\hat{\theta} = \text{argmin}_{\theta\in\Theta} \sum_{i=1}^{n} \Psi((Y,T)_i,\theta). \tag{8}$$

The objective function is used to estimate the parameters and for partitioning. This includes testing and split point estimation. In the case of ordinary least squares (OLS), the objective function Ψ is the error sum of squares. In the case of maximum likelihood (ML), it is the negative log-likelihood. Equation 8 is equivalent to solving the score function

$$\sum_{i=1}^{n} \psi((Y,T)_i,\hat{\theta}) = 0,$$

where $\psi((Y,T)_i,\hat{\theta}) = \frac{\partial\Psi((Y,T)_i,\theta)}{\partial\theta}$.

In the binary case, minimising the objective function results in a maximum likelihood estimator. Thus, the objective function is the negative log-likelihood.

In many situations, the global model $\mathcal{M}((Y,T),\theta)$ does not result in a good fit for all observations in the dataset. Nevertheless, the fit can be improved by partitioning the observations in subgroups. For this purpose, recursive partitioning based on p

partitioning variables $X_1, \ldots, X_p$ can be applied.

Consider a partition $\{\mathcal{B}_b^{MOB}\}_{b=1,\ldots,B}$ with B leaf nodes. In each leaf a model $\mathcal{M}((Y,T),\theta_b)$ with a node specific parameter θ_b holds. The respective partition $\{\mathcal{B}_b^{MOB}\}$ is defined by the p partitioning variables X. These variables contain characteristics of each person that potentially influence the intercept as well as the treatment effect. If the correct partition $\{\mathcal{B}_b^{MOB}\}$ is given, the parameters θ_b can be easily computed by minimising the segmented objective function. This leads to subgroup specific intercepts and treatment effects. For a binary response, generalised linear models with a binomial family are fitted. The resulting trees are called *GLM trees* (Zeileis *et al.* 2008).

Intercepts and treatment effects of the subgroups are estimated via model-based recursive partitioning. For this, parameter instabilities are detected by testing for non-constant parameters. Since the interest lies in detecting non-constant intercepts and treatment effects, the following partial score functions are applied:

$$\psi_\mu((Y,T),\hat{\theta}) = \frac{\partial\Psi((Y,T),\theta)}{\partial\mu}$$

$$\psi_{\tau^*}((Y,T),\hat{\theta}) = \frac{\partial\Psi((Y,T),\theta)}{\partial\tau^*}.$$

In the case of constant intercepts and treatment effects between the subgroups, the partial score functions are independent of the partitioning variables. A correlation between a partial score function with some patient characteristics indicates that certain information was not taken into account. This means that the intercepts and treatment effects differ between the subgroups and parameter instability exists. Hence, the corresponding test is an independence test between the partial score functions and the partitioning variables (Seibold *et al.* 2016).

For this purpose, Zeileis & Hornik (2007) introduce so-called "M-fluctuation tests". These tests are based on the idea to check whether the scores fluctuate randomly around their zero mean or if there are some systematic deviations from zero over a partitioning variable. Using these tests requires a distinction between numerical and categorical partitioning variables. For numerical partitioning variables the supLM (Lagrange Multiplier) test can be applied. The supLM statistic is the supremum of all single-split LM statistics. For categorical partitioning variables, a χ^2 test is proposed. The χ^2 statistic captures the fluctuation within each of the categories of the partitioning

variable. The χ^2 test is also an LM-type test and asymptotically equivalent to the corresponding likelihood ratio test.

If there is a significant instability considering any of the partition variables, the node is split into locally optimal segments. Afterwards the procedure is repeated. For this, the variable with the lowest p-value is chosen, respectively (Zeileis *et al.* 2008).

The split point is obtained by using the objective function. For each possible split, the model is estimated in the two resulting subgroups. Their objective functions are summed up afterwards. The split optimising the segmented objective function is chosen. It corresponds to the minimum of the summed objective functions across all splits. (Zeileis & Hothorn).

The steps of the algorithm, described by Zeileis *et al.* (2008), are summarised in Algorithm 1.

Algorithm 1: Model-based recursive partitioning

1. Fit parametric model to a dataset with all observations.
2. Test for parameter instability over a set of partitioning variables $X_1, \ldots, X_p$.
3. If there is some overall parameter instability, select the variable X_j associated with the highest parameter instability, otherwise stop.
4. Compute the split point(s) that locally optimise the objective function.
5. Split the node with respect to the variable associated with the highest instability X_j into child nodes and repeat the procedure.

5.1.2 Tree Pruning

In order to reduce the size and complexity of the tree, some of its sections are removed. For this purpose, either a pre-pruning or post-pruning strategy can be applied. Pre-pruning the tree is included in the tree-building phase: the algorithm stops when no significant parameter instabilities are detected anymore. In the present thesis, post-pruning is additionally applied manually. The cross table of the treatment and outcome in each leaf is inspected. There should be at least four observations in each cell. However, in some leaves this does not hold. In these leaves, the estimated coefficients are not meaningful anymore. To avoid this problem, the cross table in each leaf of a tree is checked. If there are less than four observations in one cell, the leaf is pruned.

5.1.3 Conversion of the Treatment Effect

As already described in the previous chapter, the treatment effect for a binary response is generally defined by

$$\tau(x) = \mathbb{P}(Y = 1|T = 1, X) - \mathbb{P}(Y = 1|T = 0, X). \tag{9}$$

In the GLM tree, the treatment effect is directly estimated by a generalised linear model. Thus, the treatment effect is expressed by a log odds ratio, denoted by τ^*. It is defined by

$$\begin{aligned}
\tau^*(x) &= \log\Big(\frac{\mathbb{P}(Y = 1|T = 1, X)/\mathbb{P}(Y = 0|T = 1, X)}{\mathbb{P}(Y = 1|T = 0, X)/\mathbb{P}(Y = 0|T = 0, X)}\Big) \\
&= \log\Big(\frac{\mathbb{P}(Y = 1|T = 1, X)}{\mathbb{P}(Y = 0|T = 1, X)}\Big) - \log\Big(\frac{\mathbb{P}(Y = 1|T = 0, X)}{\mathbb{P}(Y = 0|T = 0, X)}\Big) \\
&= \text{logit}(\mathbb{P}(Y = 1|T = 1, X)) - \text{logit}(\mathbb{P}(Y = 1|T = 0, X)).
\end{aligned}$$

To obtain the treatment effect expressed as difference (see Equation 9), the log odds ratio is transformed by the response function (see Section 6.3 for details):

$$\text{logit}(\mathbb{P}(Y = 1|T, X)) = \mu(x) + \tau^*(x) \cdot I_{t=1}.$$

This results in the equations

$$\begin{aligned}
\mathbb{P}(Y = 1|T = 0, X) &= \frac{\exp(\mu(x))}{1 + \exp(\mu(x))}, \\
\mathbb{P}(Y = 1|T = 1, X) &= \frac{\exp(\mu(x) + \tau^*(x))}{1 + \exp(\mu(x) + \tau^*(x))},
\end{aligned}$$

whereas $\mu(x)$ = mean effect function (intercept).

All other methods that are outlined in the following, either estimate $\mathbb{P}(Y = 1|T = 1, X)$ and $\mathbb{P}(Y = 1|T = 0, X)$ separately or they directly estimate the difference.

5.2 Causal Trees

Another possibility of estimating heterogeneous treatment effects are *causal trees*. This method directly computes treatment effects with adapted regression trees. The

treatment effects are expressed as the difference between the expected outcomes of the two treatment groups. Regression trees partition observations into subgroups of similar outcomes. It is a well-suited method to identify important predictors of outcomes and to partition observations into groups with similar characteristics.

5.2.1 The Classical CART Algorithm

The classification and regression tree (CART) algorithm, introduced by Breiman *et al.* (1984), can be either used for a continuous or categorical response. Continuous response variables result in so-called regression trees and categorical response variables in classification trees. They consist of two parts: the tree building phase and cross-validation to select the complexity parameter for subsequent pruning. In the tree building phase, the observations of the training sample are recursively partitioned. In each leaf, all possible splits are evaluated with the help of a "splitting" (in-sample goodness of fit) criterion. In regression trees this is the mean squared error (MSE).

To prevent overfitting, cross-validation is used. A penalty term for the tree depth is specified which is added to the criterion. The penalty parameter represents the costs of a leaf. Smaller leaf penalties lead to deeper trees and smaller leaves. This results in higher variance estimates of leaf means and therefore to a larger average MSE across the cross-validation samples. Applying the penalty term, only splits leading to an improvement of the goodness of fit criterion larger than some threshold are considered. The penalty term is chosen, such that the goodness of fit criterion in cross-validation samples is maximised. For cross-validation, the training sample is repeatedly split into two subsamples, one to build the tree and estimate the conditional means, the other to evaluate the estimates.

Athey & Imbens (2015) use the idea of regression trees to detect heterogeneous treatment effects in a population. They propose four different tree types, which differ with respect to the splitting criterion: Transformed Outcomes Trees (TOT), Fit-Based Trees (F), Squared T-Statistic Trees (TS) and Causal Trees (CT). In the following, causal trees are described and applied. According to Athey & Imbens (2015), this is the preferred method.

5.2.2 Honest Causal Trees

There are two main differences between classical regression trees and causal trees. At first, the focus of causal trees is on estimating conditional average treatment effects rather than predicting outcomes. Thus, causal trees are based on a splitting criterion that maximises treatment effect heterogeneity. It focuses on mean squared error of the estimation of the treatment effects instead of mean squared error of predictions of outcomes. The treatment effect in each leaf is estimated by taking the difference between the sample average of the treated group and the sample average of the control group, respectively.

Furthermore, the method relies on sample splitting. For this purpose, the dataset is divided in two parts. In the first part the optimal partition (training sample) is constructed and in the second part the effects within the leaves (estimation sample) are estimated. After building/splitting the tree with the training sample, the estimation sample is sent down the tree to a leaf node. The treatment effect is subsequently estimated within each leaf by taking the difference between means of the treated and control group. In the binary case the "mean" corresponds to the probability of observing 1. This procedure is called *honest* estimation. According to Athey & Imbens (2015), standard machine learning methods are biased because they use the same training data for model selection and estimation. Spurious correlations between covariates and outcome affect the selected model. Honest methods avoid this problem by using different (and independent) information for selecting the model and for estimation. Systematic bias in the estimation is ignored by adjusting the splitting and cross-validation criteria. Instead, they focus on the trade-off between leaf size and variance. Small leaf size leads to more precise estimations, but variance increases at the same time. Honesty eliminates the bias but at the same time there is also a potential loss of precision resulting from a smaller sample size.

Furthermore, Athey & Imbens (2015) propose an adaptive, non-honest version. The trees are estimated without sample splitting. Nevertheless, the honest version should be preferred because of the reduction of the bias. All other methods mentioned above (Transformed Outcomes Trees (TOT), Fit-Based Trees (F), Squared T-Statistic Trees (TS)) provide adaptive as well as honest versions.

5.2.2.1 Honest Splitting

One problem of estimating the conditional average treatment effect is that the true value of the treatment effect is not observed. However, it can be estimated. These estimates are used for splitting as well as for cross-validation. Correspondingly, a causal tree is a data-driven approach to partition the data into subpopulations.

For each observation a triple (Y_i^{obs}, X_i, T_i) is observed. Given a sample $\mathcal{S}$, let $\mathcal{S}_{treat}$ be the subsample for the treated and $\mathcal{S}_{control}$ for the control units.

Let $\{\mathcal{B}_b^{CT}\}_{b=1,\dots,B}$ be a causal tree with B leaf nodes ℓ and $\ell(x, \{\mathcal{B}_b^{CT}\})$ a leaf node $\ell \in \{\mathcal{B}_b^{CT}\}$ such that $x \in \ell$.

The population average outcome in both treatment groups of the partition $\{\mathcal{B}_b^{CT}\}$ is estimated by

$$\hat{\mu}(t, x; \mathcal{S}, \{\mathcal{B}_b^{CT}\}) = \frac{1}{\#(\{i \in \mathcal{S}_t : X_i \in \ell(x, \{\mathcal{B}_b^{CT}\})\})} \sum_{i \in \mathcal{S}_t : X_i \in \ell(x, \{\mathcal{B}_b^{CT}\})} Y_i^{obs}.$$

Thus, the estimation of the causal effect is

$$\hat{\tau}(x; \mathcal{S}, \{\mathcal{B}_b^{CT}\}) = \hat{\mu}(1, x; \mathcal{S}, \{\mathcal{B}_b^{CT}\}) - \hat{\mu}(0, x; \mathcal{S}, \{\mathcal{B}_b^{CT}\}).$$

The mean squared error for the treatment effect is defined by

$$\text{MSE}_\tau(\mathcal{S}^{te}, \mathcal{S}^{est}, \{\mathcal{B}_b^{CT}\}) = \frac{1}{N^{te}} \sum_{i \in \mathcal{S}^{te}} \{(\tau_i - \hat{\tau}(X_i; \mathcal{S}^{est}, \{\mathcal{B}_b^{CT}\}))^2 - \tau_i^2\},$$

whereas $\mathcal{S}^{te}$ is the test sample, N^{te} is the number of observations in the test sample and $\mathcal{S}^{est}$ corresponds to the estimation sample. As mentioned above, $\mathcal{S}^{est}$ is an independent sample in the honest estimation algorithm for estimating the leaf means and generating unbiased estimates.

Taking the expectation of $\text{MSE}_\tau(\mathcal{S}^{te}, \mathcal{S}^{est}, \{\mathcal{B}_b^{CT}\})$ over the estimation and test sample leads to the adjusted expected MSE:

$$\text{EMSE}_\tau(\{\mathcal{B}_b^{CT}\}) = \mathbb{E}_{\mathcal{S}^{te}, \mathcal{S}^{est}}[\text{MSE}_\tau(\mathcal{S}^{te}, \mathcal{S}^{est}, \{\mathcal{B}_b^{CT}\})].$$

This criterion cannot be evaluated, because the true treatment effect τ_i is unobserved. It is estimated as follows.

For the adaptive version, an estimator of the infeasible in-sample goodness of fit criterion can be constructed by

$$-\widehat{\text{MSE}}_\tau(\mathcal{S}^{tr}, \mathcal{S}^{tr}, \{\mathcal{B}_b^{CT}\}) = \frac{1}{N^{tr}} \sum_{i \in \mathcal{S}^{tr}} \hat{\tau}^2(X_i; \mathcal{S}^{tr}, \{\mathcal{B}_b^{CT}\}), \tag{10}$$

whereas $\mathcal{S}^{tr}$ is the training sample and N^{tr} the number of observations in the training sample.

This estimator can be modified to an unbiased *honest* tree estimator by decoupling the model selection from the model estimation. The sample is split in two subsamples, one to build the tree and one to estimate the effects. The splitting objective function is defined by

$$-\text{EMSE}_\tau(\{\mathcal{B}_b^{CT}\}) = \mathbb{E}_{X_i}[\tau^2(X_i; \{\mathcal{B}_b^{CT}\})] - \mathbb{E}_{\mathcal{S}^{est}, X_i}[\mathbb{V}(\hat{\tau}^2(X_i; \mathcal{S}^{est}, \{\mathcal{B}_b^{CT}\}))].$$

Using only the training sample $\mathcal{S}^{tr}$,

$$\begin{aligned} -\widehat{\text{EMSE}}_\tau(\mathcal{S}^{tr}, \{\mathcal{B}_b^{CT}\}) = &\frac{1}{N^{tr}} \sum_{i \in \mathcal{S}^{tr}} \hat{\tau}^2(X_i; \mathcal{S}^{tr}, \{\mathcal{B}_b^{CT}\}) \\ &- \frac{2}{N^{tr}} \sum_{\ell \in \{\mathcal{B}_b^{CT}\}} \left(\frac{S^2_{\mathcal{S}^{tr}_{treat}}(\ell)}{p} + \frac{S^2_{\mathcal{S}^{tr}_{control}}(\ell)}{1-p} \right) \end{aligned} \tag{11}$$

is the corresponding estimator for the infeasible criterion, with $p = N^{tr}/n$, i.e. the marginal treatment probability (the probability of allocation to the treatment group) (Athey & Imbens 2015). As described by Athey *et al.* (2016b), a parameter α can be added to Equation 11 to control the depth of the tree. That leads to the estimator:

$$\begin{aligned} -\widehat{\text{EMSE}}_\tau(\mathcal{S}^{tr}, \{\mathcal{B}_b^{CT}\}) = &\alpha \cdot \frac{1}{N^{tr}} \sum_{i \in \mathcal{S}^{tr}} \hat{\tau}^2(X_i; \mathcal{S}^{tr}, \{\mathcal{B}_b^{CT}\}) \\ &- (1-\alpha) \cdot \frac{2}{N^{tr}} \sum_{\ell \in \{\mathcal{B}_b^{CT}\}} \left(\frac{S^2_{\mathcal{S}^{tr}_{treat}}(\ell)}{p} + \frac{S^2_{\mathcal{S}^{tr}_{control}}(\ell)}{1-p} \right). \end{aligned}$$

α-values close to 1 indicate that the tree prefers leaves with heterogeneous effects (more weight on the first part of the equation), which results in deeper trees. α-values near 0 means that the tree prefers leaves with a good fit (more weight on the second part of

the equation). Thus, the trees are getting smaller.

5.2.2.2 Honest Cross-Validation

For cross-validation in the adaptive version, the objective function defined in Equation 10 can be applied. It is evaluated for the samples $\mathcal{S}^{tr,cv}$ and $\mathcal{S}^{tr,tr}$. This leads to the cross-validation criterion $-\widehat{\text{MSE}}_\tau(\mathcal{S}^{tr,cv}, \mathcal{S}^{tr,tr}, \{\mathcal{B}_b^{CT}\})$. $\mathcal{S}^{tr,tr}$ is the part of the training sample for building a new tree and estimating the conditional means and $\mathcal{S}^{tr,cv}$ for evaluating the estimates.

For cross-validation in the honest version, the objective function from Equation 11 is used, but evaluated for the cross-validation sample: $-\widehat{\text{EMSE}}_\tau(\mathcal{S}^{tr,cv}, \{\mathcal{B}_b^{CT}\})$. Nevertheless, it might have a higher variance than the adaptive criterion. This is due to the smaller sample-size in the cross-validation sample.

The algorithm of causal trees is summarised in Algorithm 2.

Algorithm 2: Honest causal tree

1. Divide data into tree-building $\mathcal{S}^{tr}$ and estimation samples $\mathcal{S}^{est}$.
2. Recursively partition covariates into a deep partition $\{\mathcal{B}_b^{CT}\}$:
- Select split that minimises $\widehat{\text{EMSE}}$ over all possible binary splits.
- Preserve minimum number of treated and control subjects in each child leaf.

3. Use cross-validation to select the depth of the partition.
4. Select partition $\{\mathcal{B}_b^{CT}\}^*$ by pruning $\{\mathcal{B}_b^{CT}\}$, i.e. pruning leaves that provide the smallest improvement in goodness of fit.
5. Estimate the treatment effects in each leaf of $\{\mathcal{B}_b^{CT}\}^*$ using $\mathcal{S}^{est}$.

Causal trees assume unconfoundedness. To adjust for confounding and to remove bias in the estimates, IPTW can be applied (Athey & Imbens 2015).

A disadvantage of causal trees is that the treatment effects are not personalised. Instead, treatment effects are estimated per subgroup. All individuals in one subgroup are assumed to have the same treatment effect. For this problem Wager & Athey (2018) propose causal forests.

5.3 Causal Forests

Causal forests extend the popular random forest algorithm (Breiman 2001). Building forests reduces the variance of single trees. Distinct trees are created by applying bootstrap sampling.

A causal forest can be implemented in different ways. The algorithm to grow a forest as well as the applied splitting rule to maximise heterogeneity in the treatment effect differ.

Athey *et al.* (2018) propose *generalised random forests* (GRF), which express heterogeneity in a key parameter of interest. The main idea is based on random forest. Thus, recursive partitioning, subsampling and random split selection are kept. Nevertheless, the prediction of a test point x is not obtained by averaging over the trees, but by using an adaptive nearest neighbour weighting. For this purpose, the forests are treated as a type of adaptive nearest neighbour estimator. Each observation gets weighted according to the frequency it falls into the same leaf as the target observation, i.e. the target value of the covariate vector. Random forests can also be thought of being an adaptive kernel method. The classical kernel weighting is replaced by forest-based weights, which are derived from a forest designed to express heterogeneity. Thus, the algorithm is a computationally efficient way to grow forest-based weighting functions.

The algorithm begins by computing a linear, gradient-based approximation to the nonlinear estimating equation to be solved. The reason for using a gradient-based approximation is, that a direct maximisation of a criterion would be computationally costly. Hence, the algorithm is closely related to gradient boosting. Afterwards, the approximation is applied to specify the tree-split point as in a standard regression tree. As in causal trees (see Section 5.2), splitting is performed by simply maximising the variance of the treatment effect, instead of the variance of the outcome. An honest splitting procedure is also possible.

5.3.1 Forest-Based Local Estimation

Imagine for each observation $i = 1, \ldots, n$ a quantity O_i is observed which contains an outcome Y_i and the treatment assignment T_i ($O_i = \{Y_i, T_i\}$). That quantity encodes information about $\theta(\cdot)$, which shall be estimated. $\theta(\cdot)$ can be any quantity. In the case

of causal forests, it is defined as the treatment effect. To estimate $\theta(\cdot)$, equations of the form

$$\mathbb{E}[\psi_{\theta(x)}(O_i)|X_i = x] = 0 \quad \forall x \tag{12}$$

have to be solved, whereas $\psi(\cdot)$ is some scoring function. For this purpose, some kind of similarity weights $\alpha_i(x)$ are defined. The respective weight $\alpha_i(x)$ measures the relevance of observation i to fit $\theta(\cdot)$ for a specific x. Afterwards, $\theta(\cdot)$ is fitted with an empirical version of the estimating equation:

$$\hat{\theta}(x) \in \text{argmin}_\theta \Big\{ \Big|\Big| \sum_{i=1}^{n} \alpha_i(x)\psi_\theta(O_i) \Big|\Big|_2 \Big\},$$

with $||x||_2 = \sqrt{x_1^2 + \cdots + x_n^2}$ the euclidean norm.

In the case of a unique root, $\hat{\theta}(x)$ solves $\sum_{i=1}^{n} \alpha_i(x)\psi_{\hat{\theta}(x)}(O_i) = 0$.

The weights are traditionally obtained by a deterministic kernel function, which performs well in a low dimensional parameter-space. In GRF, forest-based weights are used. First, a set of M trees is grown. Let $\mathcal{S}_m^{tr}(x)$ be the set of training examples of the mth tree that fall into the same leaf as x. The weights correspond to the frequency of the training example i and x being in the same leaf:

$$\alpha_{mi}(x) = \frac{I(\{X_i \in \mathcal{S}_m^{tr}(x)\})}{|\mathcal{S}_m^{tr}(x)|}, \quad \alpha_i(x) = \frac{1}{M} \sum_{m=1}^{M} \alpha_{mi}(x).$$

The weights add up to 1. Furthermore, they define the forest-based adaptive neighbourhood of x. This is demonstrated in Figure 5.1. The rectangles in this illustration correspond to the leaf nodes. In each tree, the training examples located in the same leaf as the test point x receive the same positive weight. All other training examples get a weight of zero. The forest averages these tree-based weightings.

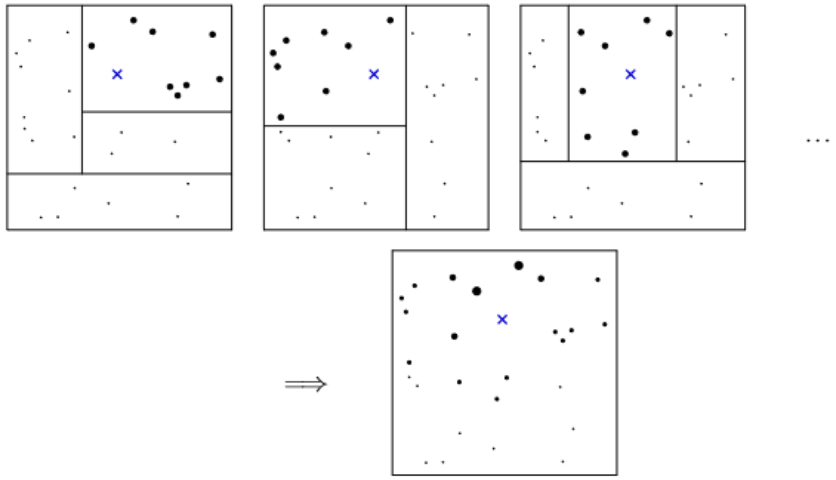

Figure 5.1: Random forest weighting function (Athey et al. 2016, p. 6)

5.3.2 Splitting to Maximise Heterogeneity

The quality of a tree shall be best possible improved by the chosen split. The splitting procedure focuses on heterogeneity in $\theta(x)$. The resulting trees combined into a forest should induce weights that lead to good estimates of $\theta(x)$.

Each split starts with a parent node P given a sample of data $\mathcal{S}$. Let $\hat{\theta}_P(\mathcal{S})$ be the solution to the estimating equation

$$\hat{\theta}_P(\mathcal{S}) \in \text{argmin}_\theta \left\{ \left\| \sum_{\{i \in \mathcal{S}: X_i \in P\}} \psi_\theta(O_i) \right\|_2 \right\}. \tag{13}$$

Then, P is divided into two child nodes C_1, C_2 to obtain optimal estimates of θ. The splits are chosen to maximise

$$\Delta(C_1, C_2) = n_{C_1} n_{C_2} / n_P^2 (\hat{\theta}_{C_1}(\mathcal{S}) - \hat{\theta}_{C_2}(\mathcal{S}))^2,$$

whereas n_P is the number of observations in the parent node and n_{C_j} the number of observations in each child node.

$\hat{\theta}_{C_1}$ and $\hat{\theta}_{C_2}$ are solutions to the estimating equation (see Equation 13) achieved in the child nodes. This is called the exact loss criterion. With this criterion the solution of $\hat{\theta}_{C_1}$ and $\hat{\theta}_{C_2}$ might be computationally expensive. Therefore, an alternative approximate

criterion $\tilde{\Delta}(C_1, C_2)$ is applied. That criterion creates gradient-based approximations and has to be maximised. It is defined by

$$\tilde{\Delta}(C_1, C_2) = \sum_{j=1}^{2} \frac{1}{|\{i : X_i \in C_j\}|} \Big(\sum_{\{i: X_i \in C_j\}} \rho_i \Big)^2, \tag{14}$$

with ρ_i corresponding to some pseudo-outcomes obtained in a first labelling step. The pseudo-outcomes are computed by using a derivative matrix. Maximising Equation 14 leads to child nodes C_1 and C_2. After splitting, the observations in each child node are relabelled by solving the estimating equation (see Equation 13). Overall, the forest consists of many gradient trees. According to Knaus *et al.* (2018), the treatment effect is estimated by

$$\hat{\tau}(x) = \sum_{i=1}^{n} T_i \alpha_i(x) Y_i - \sum_{i=1}^{n} (1 - T_i) \alpha_i(x) Y_i.$$

For observational data, the GRF can be implemented with a local centering approach. It reduces the bias in the case of confounders. For this, the outcome Y_i and the treatment T_i are locally centered before building the forest. Hence, the effect of the features X_i is regressed out on all outcomes separately. Define $y(x) = \mathbb{E}[Y_i | X = x]$ and $t(x) = \mathbb{E}[T_i | X = x]$ as the conditional marginal expectations of Y_i and T_i. The centered outcomes are defined by

$$\tilde{Y}_i = Y_i - \hat{y}^{(-i)}(X_i) \quad \text{and} \quad \tilde{T}_i = T_i - \hat{t}^{(-i)}(X_i),$$

whereas $\hat{y}^{(-i)}$ and $\hat{t}^{(-i)}$ are the leave-one-out estimates of the marginal expectations. The expected response and the treatment propensities are estimated by regression trees. They are the nuisance parameters $\nu(x)$ in the splitting and estimation procedure that can be added optionally. Afterwards, the causal forest is ran on the centered outcomes $\{\tilde{Y}_i, \tilde{T}_i\}_{i=1}^n$ instead of $\{Y_i, T_i\}_{i=1}^n$ (Athey *et al.* 2018).

Thus, the causal forest is the only method in this thesis, where confounding is not considered by a propensity score adjustment.

5.3.3 Alternative Causal Forest Algorithms

Wager & Athey (2018) introduce two more types of causal forests closely related to GRF.

- *Double Sample Trees*: They are similar to GRFs without centering. The main difference is that the exact loss criterion is used for splitting instead of a gradient-based loss criterion. Additionally, treatment effects are computed separately in each tree, rather than using a specific weighting scheme.
- *Propensity Forest*: This forest obtains its neighbourhood function via a classification forest on the treatment assignments. Thus, it only applies the treatment assignment to place splits. Heterogeneous effects are ignored (Athey *et al.* 2018).

Athey *et al.* (2018) compare the four different types of causal forests (Double Sample Tree, Propensity Forests and GRF with and without local centering) in three different simulation studies. The simulations include heterogeneous treatment effects with and without confounding and a non-heterogeneous treatment effect with confounding. They demonstrate that GRFs with centering perform well in all settings. In case of confounding, GRFs without centering lead to only slightly worse results. The double sample trees perform poorly with confounding. Propensity forests cannot handle strong treatment effect heterogeneity. Due to these results, GRFs with local centering are used in this thesis.

5.4 Bayesian Additive Regression Trees (BART)

A further method is the *Bayesian Additive Regression Tree* (BART), developed by Chipman *et al.* (2010). In the following, this model is used as a G-computation approach. "In G-computation any multivariable regression model can be used to regress the outcome on treatment status and baseline covariates. Using the fitted model, the two potential outcomes can be estimated for each patient and then the CATE is estimated by taking the difference between the two imputed potential outcomes" (Wendling *et al.* 2018 p. 3). Many off-the-shelf statistical learning algorithms can be used for G-computation. This is an advantage over causal learning algorithms.

In general, BART is a nonparametric method for fitting functions. It uses the sum of small regression trees. Hill (2011) describes how these trees are used to estimate personalised causal effects. For estimating the treatment effect as defined in Equation 7, $\mu(1, x)$ and $\mu(0, x)$ need to be estimated. The idea is to regress the outcome Y on the treatment status $T_i = t$ and the baseline covariates $X_i = x$.

Accordingly, the model

$$Y = \mu(t, x) + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

is specified. BART approximates $\mu(t, x)$ with a sum-of-trees model:

$$\mu(t, x) = \sum_{m=1}^{M} g(t, x; \{\mathcal{B}_b^{BA}\}_m, A_m).$$

Consequently, the model is expressed as:

$$Y = g(t, x; \{\mathcal{B}_b^{BA}\}_1, A_1) + \cdots + g(t, x; \{\mathcal{B}_b^{BA}\}_M, A_M) + \epsilon, \quad \epsilon \sim N(0, \sigma^2).$$

Let $\{\mathcal{B}_b^{BA}\}_m$ be the mth binary tree with $b = 1, \ldots, B$ leaf nodes. Decision rules send an observation with (t, x) left or right down to a leaf node of a tree $\{\mathcal{B}_b^{BA}\}$. Each of the B leaf nodes contains a parameter. This parameter corresponds to the mean response of the subgroup of observations that fall in that node. A_m is defined as the associated set of leaf node parameters of the mth tree. Hence, $g(t, x; \{\mathcal{B}_b^{BA}\}_m, A_m)$ is defined as the value obtained by sending (t, x) down the tree. BART can be regarded as a bayesian version of gradient boosting, because it is based on refitting residuals from former trees: The sum-of-tree models take the predictions from the preceding base learner, $g(t, x, \{\mathcal{B}_b^{BA}\}_1, A_m)$, and subtract it from the observed response y to form residuals. The next tree is then fitted to these residuals. This is performed M times.

Overfitting is limited by using a prior, which keeps the trees small. Each tree is only allowed to contribute a small part to the overall fit. For this, a prior over all parameters of the sum-of-tree model is specified. Those parameters are σ and $(\{\mathcal{B}_b^{BA}\}_m, A_m)$.

The algorithm works like a Gibbs-Sampler: The prior is random, the distribution of the posterior is unknown, and samples from the posterior are drawn to estimate desired values. For this purpose, the MCMC (Markov Chain Monte Carlo) method is applied. Thus, fitting and inference are accomplished with an iterative Bayesian backfitting MCMC algorithm that generates samples from a posterior. Hence, σ and $(\{\mathcal{B}_b^{BA}\}_m, A_m)$ are redrawn at each iteration, but only σ is identified (Hill 2011).

For binary outcomes BART can be extended by using a probit model:

$$\mathbb{P}[Y_i = 1 | T_i = 1, X_i = x] = \Phi\Big[\sum_{m=1}^{M} g(t, x; \{\mathcal{B}_b^{BA}\}_m, A_m)\Big]$$

with Φ = the cumulative distribution function of a standard normal distribution.

The prior is similar to that for continuous outcomes, but the model sets σ to 1 so that only priors on $(\{\mathcal{B}_b^{BA}\}_m, A_m)$ are needed (Green & Kern 2012).

In case of strong confounding in observational studies, the propensity score can be included as a baseline covariate in the outcome model (given that there are no unmeasured confounders) (Wendling *et al.* 2018). However, the BART model is in general robust to confounding, because it can model the response surface very flexibly. Thus, propensity score adjustment is usually not necessary (Hill 2011).

5.5 Pollinated Transformed Outcome (PTO) Forest

Another approach to estimate personalised treatment effects is the *Pollinated Transformed Outcome* (PTO) *Forest*. This algorithm was introduced by Powers *et al.* (2018). In a first step it transforms the outcome with the inverse of the propensity score. Hence, each patient's transformed outcome is equal to the original outcome divided by the probability of receiving the respective treatment. For $T_i = 0$, the new value is multiplied by minus one. Afterwards, a random forest model is fitted with the new transformed outcome as response based on the whole dataset. The forest is then "pollinated" with the treated and control population. "Pollinate" means that the control and treated observations are sent down the tree separately and new predictions are computed for each leaf node. The treatment effect is obtained by taking the difference of these predictions. Optionally, a random forest can be fitted to this difference. This optional step helps with the interpretability of the results, because the importance scores of the variables are obtained. The algorithm is defined in Algorithm 3. This method is similar to the virtual twins, described in Chapter 2.

Algorithm 3: Pollinated transformed outcome (PTO) forest

1. Transform outcome: $Y_i^* = T_i \cdot \frac{Y_i}{e(X_i)} - (1 - T_i) \cdot \frac{Y_i}{1-e(X_i)}$
2. Fit random forest G with transformed outcome $(G = Y^* \sim \dots)$
3. "Pollinate" G for populations with $t = 0$ and $t = 1$ separately: Send data down each tree and compute new predictions $\rightarrow G_1$, G_0
4. Compute treatment effect: $\delta_i = G_1(X_i) - G_0(X_i)$
5. Optionally: Fit a random forest G^* to δ_i. Then, use G^* for prediction of the treatment effect: $\hat{\tau}(x) = G^*(x)$.

5.6 Causal Multivariate Adaptive Regression Splines (MARS)

Besides the PTO forest, Powers *et al.* (2018) developed another method to estimate personalised treatment effects. This algorithm is called the *causal Multivariate Adaptive Regression Splines* (MARS) and is an alternative to tree-based methods. A disadvantage of trees is the potential high bias in the estimate as they use average treatment effects within each leaf as prediction. MARS is inspired by recursive partitioning but has the advantage that the bias of trees is weakened. Furthermore, continuous models with continuous derivatives are built. This leads to more power and flexibility (Powers *et al.* 2018).

5.6.1 The Original MARS Algorithm

The causal MARS algorithm is based on the idea of multivariate adaptive regression splines (MARS), developed by Friedman (1991). The method solves regression-type problems with the main purpose of predicting values. MARS is a nonparametric extension of linear models which identifies "non-linearities" and interactions automatically. Accordingly, no assumptions about the underlying functional relationship between dependent and independent variables are necessary. Hence, it is a flexible regression method where spline basis functions are added in each step and bias in predictions is prevented. The outcome is defined by

$$Y = f(x_1, \ldots, x_n) + \epsilon, \quad \text{where} \quad \hat{f}(x) = \sum_{d=1}^{D} \beta_d b_d(x),$$

with $b_d(x)$ = basis function and β_d = coefficient.

The relation between a dependent and independent variables is modelled by using a set of coefficients and basis functions. These are entirely derived from the data. The model selects a weighted sum of basis functions from the space of basis functions that span all values of each predictor, i.e. for each variable and for all possible nodes. In this step, interactions between variables are also considered. The basis functions are added to the model to maximise an overall least squares goodness of fit criterion (i.e. minimise the prediction error). It automatically determines the most important independent variables and the most significant interactions among them. The input space is divided

into regions, each with its own regression equation. Each breakpoint is estimated from the data and defines the region of a particular regression equation. These breakpoints have a similar effect as step functions. See Algorithm 4 for details.

Algorithm 4: Multivariate adaptive regression splines

1. Define $f(x) = \beta_0$ (which means: $b_1(x) = 1$).
2. Consider adding function pairs of the form: $\{(x_j - c)_+, (c - x_j)_+\}$ and the products of variables in the model with these pairs.
3. Choose pair with biggest reduction in the "training error" when adding it to the model.
4. Regression coefficients are estimated via OLS.

In CART, the function pairs have the form: $\{I_{\{x_j - c \geq 0\}}, I_{\{c - x_j > 0\}}\}$.

MARS can also handle classification problems. The model is fitted on the indicator variables of the categorical response variable and the predicted scores are computed. Each observation is assigned to the class with the highest predicted score.

5.6.2 Causal MARS

To estimate personalised treatment effects with MARS models, a MARS is fitted for each treatment group, respectively. In each step the same basis function is chosen and added to each model. In order to find the best basis function in terms of explaining treatment effect variance, compare:

- Decrease in training error by including the basis in both models with *different* coefficients (RSS_τ).
- Decrease in training error by including the basis in both models with *same* coefficients (RSS_μ).

The basis function which maximises $dRSS = RSS_\tau - RSS_\mu$ is chosen. The algorithm (initially for the randomised case) is summarised in Algorithm 5.

To reduce overfitting and variance, bagging can be used. For this purpose, bootstrap samples of the original dataset are drawn. The causal MARS model is fitted to each of

them. Afterwards, the average of the estimates is taken to obtain the treatment effect of an individual.

In observational data, the model can be built for S propensity strata. The variable $s \in \{1, \ldots, S\}$ indicates the stratum a patient belongs to. The same basis function is used within each stratum, but different coefficients are estimated for each person. The estimation of the coefficients in the different strata is independent. The criterion is defined as $\sum_s n_s dRSS_s$, with n_s equal to the number of patients in each stratum (Powers *et al.* 2018).

Algorithm 5: Causal MARS

Define $\mathcal{F} = \{\{(x_j - c)_+, (c - x_j)_+\} : c \in \{X_{ij}\}, j \in \{1, \ldots, p\}\}$.
Initialise $\mathcal{K} = \{1\}$.
for *d in* $1, \ldots, D$ *(growing the model)* **do**
  for *each pair of functions*
  $\{f, g\} \in \{\{b(x)f^*(x), b(x)g^*(x)\} : b \in \mathcal{K}, \{f^*, g^*\} \in \mathcal{F}\}$ **do**
    i)
$$RSS_\mu = \min_{\beta^1, \beta^0} \sum_{i=1}^{n} \Bigg(y_i - \sum_{b \in \mathcal{K}} (\beta_b^1 b(x_i) I_{\{t_i=1\}} + \beta_b^0 b(x_i) I_{\{t_i=0\}}) - \sum_{h \in \{f,g\}} \beta_h h(x_i) \Bigg)^2$$
    ii)
$$RSS_\tau = \min_{\beta^1, \beta^0} \sum_{i=1}^{n} \Bigg(y_i - \sum_{b \in \mathcal{K}} (\beta_b^1 b(x_i) I_{\{t_i=1\}} + \beta_b^0 b(x_i) I_{\{t_i=0\}}) - \sum_{h \in \{f,g\}} (\beta_h^1 h(x_i) I_{\{t_i=1\}} + \beta_h^0 h(x_i) I_{\{t_i=0\}}) \Bigg)^2$$
    iii)
$$dRSS = RSS_\tau - RSS_\mu$$
  end
  Choose $\{f, g\}$ which maximise $dRSS$ and add them to $\mathcal{K}$.
end
Backward deletion: delete terms step by step. For this purpose, use the same criterion as in the first loop (i - iii). For the estimation of the optimal model size, use the out-of-bag (OOB) error.

5.7 Overview of Methods

The methods explained in the previous sections take different approaches to estimate treatment effects. In the following, they are briefly summarised to facilitate their comparison.

Consider a partition $\mathcal{B}$ and a sample $\mathcal{S}$. The treatment effect is defined by

$$\tau(x) = \mu(1, x) - \mu(0, x). \tag{15}$$

In the binary case this corresponds to the probability of observing $Y = 1$:

$$\mu(1, x) = \mathbb{P}[Y_i = 1 | T_i = 1, X_i = x]$$
$$\mu(0, x) = \mathbb{P}[Y_i = 1 | T_i = 0, X_i = x]$$

1. Model-based (MOB) Recursive Partitioning (GLM tree)

Concept: In each leaf node of a partition $\mathcal{B}$ subgroup specific mean and treatment effects are estimated with a GLM model (with logit link). For this purpose, the objective function in the corresponding leaf is minimised. In GLMs this corresponds to minimising the negative log likelihood:

$$\begin{pmatrix} \hat{\mu} \\ \hat{\tau}^* \end{pmatrix} = \hat{\theta} = \text{argmin}_\theta \sum_{i \in \mathcal{S}} -l((Y, T)_i, \theta).$$

The resulting treatment effect τ^* is expressed by a log odds ratio.

Treatment effect estimation: The log odds ratio is transformed by taking the response function:

$$\mathbb{P}(Y = 1 | T = 0, X) = \frac{\exp(\mu(x))}{1 + \exp(\mu(x))},$$
$$\mathbb{P}(Y = 1 | T = 1, X) = \frac{\exp(\mu(x) + \tau^*(x))}{1 + \exp(\mu(x) + \tau^*(x))}.$$

In order to obtain the treatment effect as in Equation 15, the difference of these probabilities is taken.

2. Causal Tree

Concept: A partition $\mathcal{B}$, that estimates conditional average treatment effects in each leaf, is built. For honest estimations, different samples for building the tree and estimating the treatment effects are taken.

Treatment effect estimation: Within each leaf node the population average outcome in both treatment groups ($t = 0$ or $t = 1$) is calculated, i.e.

$$\mathbb{P}[Y_i = 1 | T_i = t, X_i = x] = \frac{1}{\#(\{i \in \mathcal{S}_t : X_i \in \ell(x, \mathcal{B})\})} \sum_{i \in \mathcal{S}_t : X_i \in \ell(x,\mathcal{B})} Y_i.$$

Afterwards, the difference of these values is taken (see Equation 15).

3. Causal Forest

Concept: A forest of gradient trees is built from which weights $\alpha_i(x)$ are extracted. Observations close to a particular value x are weighted more heavily.

Treatment effect estimation: The treatment effect is calculated by applying the weights:

$$\hat{\tau}(x) = \sum_{i=1}^{n} T_i \alpha_i(x) Y_i - \sum_{i=1}^{n} (1 - T_i) \alpha_i(x) Y_i$$

4. BART

Concept: A model for the outcome Y is estimated:

$$Y = \mu(t, x) + \epsilon, \quad \epsilon \sim N(0, \sigma^2).$$

$\mu(t, x)$ is approximated with a sum-of-trees model

$$\mu(t, x) = \sum_{m=1}^{M} g(t, x; \mathcal{B}_m, A_m)$$

with A_m = the associated set of leaf node parameters of the mth binary tree $\mathcal{B}_m$. A regularisation prior keeps the trees small.

Treatment effect estimation: For binary outcomes, BART is extended by using a probit model. BART calculates individual treatment effects by estimating potential outcomes

for each patient. Therefore, it is similar to the virtual twins approach described in Chapter 2, where the forest is replaced by BART. Afterwards, the difference of these values is taken (see Equation 15).

5. PTO Forest

Concept: The outcome of each observation is transformed by dividing it by the probability of receiving the respective treatment. A random forest G on the transformed outcome is built. G is "pollinated" for the treatment and control group separately. This results in the forests G_0 and G_1.

Treatment effect estimation: The difference between the predictions of the two pollinated forests is taken: $G_1(X_i) - G_0(X_i) = \delta_i$. Optionally, a random forest G^* on δ_i can be fitted and used to predict the treatment effect: $\hat{\tau}(x) = G^*(x)$.

6. Causal MARS

Concept: The MARS algorithm is a flexible regression method that adds a spline basis function in each step.

$$Y = \sum_{d=1}^{D} \beta_d b_d(x) + \epsilon$$

with $b_d(x)$ = basis function and β_d = coefficient.

Treatment effect estimation: A MARS model for each treatment group is fitted. The treatment effect is estimated by taking the difference of the treatment groups.

6 Simulation Study

To compare the methods explained in the previous chapter, eight different datasets are simulated (see Section 6.2 for details on the data generation process). To assess the quality of these estimators, the RMSE (root mean squared error), the bias and the variance are calculated. The bias quantifies the systematic distortion of the estimated treatment effects. The variance indicates how far the estimates are moving around their average value. The RMSE includes both of these measures.

6.1 Performance Measures

The first performance measure applied in the present thesis is the root mean squared error (RMSE). It corresponds to the root of the mean squared error (MSE). The MSE is defined as the squared difference between the estimated and the true treatment effect. Thus, given the estimator $\hat{\tau}$ of τ, the RMSE is defined by

$$\text{RMSE}(\hat{\tau}(x)) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\tau}_i(x) - \tau_i(x))^2},$$

with n = number of observations.

The root is calculated to facilitate interpretability by regarding a performance measure based on the same units as the quantity being estimated. Per definition the RMSE is always non-negative. Values close to zero indicate a reasonable estimation.

The MSE incorporates both bias and variance:

$$\text{MSE}(\hat{\tau}(x)) = \text{Var}(\hat{\tau}(x)) + \text{Bias}(\hat{\tau}(x))^2. \tag{16}$$

For an unbiased estimator, the MSE is equivalent to the variance of the estimator and the RMSE to the standard error. In order to assess which part of the RMSE stems from the bias and which part from the variance, these measures are also applied for the evaluation.

The bias is defined as the difference between the means of the estimated and the true values:

$$\text{Bias}(\hat{\tau}(x)) = \frac{1}{n}\sum_{i=1}^{n}\hat{\tau}_i(x) - \frac{1}{n}\sum_{i=1}^{n}\tau_i(x).$$

Thus, in an estimation procedure, there will be an error that is expressed in the bias. In accordance to the RMSE, values close to zero indicate an appropriate estimate. The variance of $\hat{\tau}(x)$ is defined by

$$\text{Var}(\hat{\tau}(x)) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\tau}_i(x) - \frac{1}{n}\sum_{i=1}^{n}\hat{\tau}_i(x)\right)^2.$$

It represents the average deviation of an estimator from its mean.

When predicting on a test dataset, an irreducible error σ^2 is added to the MSE formula (see Equation 16). It corresponds to the error, or amount of noise, introduced by the data. It cannot be reduced by any model (Hastie *et al.* 2009).

6.2 Simulated Data

Eight different simulation approaches are performed with 300, 600 and 1000 observations, respectively. A brief overview of these simulations is presented in Table 6.1.

Confounders are those variables that contribute to $\mu(x)$ and to $e(x)$. In simulations 4-6 the confounder corresponds to the variable x_2. In simulations 7 and 8 two confounders, variables x_2 and x_{12}, are specified.

As mentioned above, these functions ($\mu(x)$, $e(x)$ and $\tau(x)$) vary in some characteristics to uncover performance distinctions for different scenarios. The key characteristics of the different datasets are summarised in Table 6.2.

Simulation	$\mu(x)$	$e(x)$	$\tau(x)$
1	0.5	0.5	3
2	0.5	0.5	$3 \cdot I_{(x_1>1)}$
3	0.5	0.5	$3x_1$
4	$0.5 + 1.2 \cdot I_{(x_2<0)}$	$\begin{cases} 0.65 & \text{if } x_2 < 0 \\ 0.4 & \text{else} \end{cases}$	3
5	$0.5 + 1.2 \cdot I_{(x_2<0)}$	$\begin{cases} 0.65 & \text{if } x_2 < 0 \\ 0.4 & \text{else} \end{cases}$	$3 \cdot I_{(x_1>1)}$
6	$0.5 + 1.2 \cdot I_{(x_2<0)}$	$\begin{cases} 0.65 & \text{if } x_2 < 0 \\ 0.4 & \text{else} \end{cases}$	$3x_1$
7	$0.5 + 1.2 \cdot I_{(x_2<0)} + 0.5 \cdot I_{(x_{12}=F)}$	$\begin{cases} 0.65 & \text{if } x_2 < 0, x_{12} = F \\ 0.4 & \text{else} \end{cases}$	$3 \cdot I_{(x_1>1)} + 2 \cdot I_{(x_{11}=T)}$
8	$0.5 + 1.2 \cdot I_{(x_2<0)} + 0.5 \cdot I_{(x_{12}=F)}$	$\begin{cases} 0.65 & \text{if } x_2 < 0, x_{12} = F \\ 0.4 & \text{else} \end{cases}$	$3x_1 + 2x_3$

Table 6.1: Functions of eight different simulations

Simulation	Nb. of Confounder	Heterogeneity	Type of Treatment Effect Function	Nb. of Variables in Treatment Effect Function
1	0	No	-	0
2	0	Yes	stepwise	1
3	0	Yes	linear	1
4	1	No	-	0
5	1	Yes	stepwise	1
6	1	Yes	linear	1
7	2	Yes	stepwise	2
8	2	Yes	linear	2

Table 6.2: Overview of characteristics of simulations

Covariates:

The simulation of covariates is identical for all datasets. In total, 20 covariates are generated. Ten of them are binary and ten are numeric. The numeric covariates are samples from a standard normal distribution. The binary covariates are generated similarly: They are zero if the sample from a standard normal distribution is smaller or equal to zero and one if it is greater than zero:

$$x_{num} \sim N(0,1), \qquad x_{bin} = \begin{cases} 0 & \text{if } N(0,1) \leq 0 \\ 1 & \text{if } N(0,1) > 0. \end{cases}$$

Outcome:

In this thesis, the outcome is a binary variable. It is generated from the probability of getting a positive outcome given the covariates and the treatment

$$\mathbb{P}(Y = 1|T, X) = \mu(x) + \tau(x) \cdot I_{t=1},$$

with $\mu(x)$ = mean effect (intercept) and $\tau(x)$ = treatment effect.

This probability is used to draw from the binomial distribution.

6.3 Propensity Score Model

Before estimating the treatment effects, a model to estimate the propensity scores is required. This is a crucial step to ensure that the propensity adjustments are as precise as possible. For this purpose, different models are evaluated in this section.

The propensity score model should include those covariates that are related to the outcome (prognostical covariates) and, if applicable, the treatment (confounder). The variables solely influencing the treatment-selection must not be included in the propensity score model. It could lead to bias and an increased variance in the treatment effect estimate. Causal inference using the propensity scores requires some assumptions presented in Chapter 3. Those are consistency, unconfoundedness and positivity. Additionally, the assumption of no misspecification of the propensity score model has to be fulfilled (Austin & Stuart 2015).

In order to estimate the propensity score model, the following models are compared: generalised linear model (GLM) (with a logit link), classification and regression tree

(CART), random forest, conditional inference tree (ctree) and conditional inference forest (cforest).

GLM: In a generalised linear model, the non-continuous expected response $\mathbb{E}(y)$ is modelled given a linear predictor $\eta_i = x_i^T \beta$. A GLM estimates the coefficients β by maximising the log-likelihood. To guarantee that a prediction is limited to the interval $[0, 1]$, different link functions can be used. For this thesis, the logit link is applied: $\pi_i = h(\eta_i)$ (with $h(\cdot)$ = response function) or $\eta_i = g(\pi_i)$ (with $g(\cdot)$ = link function). The response and link functions are defined by

$$\pi_i = h(\eta_i) = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)} \Leftrightarrow \eta_i = g(\pi_i) = \log\Big(\frac{\pi_i}{1 - \pi_i}\Big),$$

with $\pi_i = \mathbb{P}(y_i = 1|x_i)$ (Fahrmeir *et al.* 2009).

CART and Random Forest: The classification and regression tree (CART) is defined as described in Section 5.2.1. Due to the binary response, classification trees are fitted. A random forest consists of a set of these trees, combined with randomised node optimisation and bagging. Random forests by Breiman (2001) are an ensemble learning method for classification. In a random forest model, single deep trees are averaged to reduce their variance and to avoid overfitting. Each tree is built based on a bootstrap sample (random sampling with replacement) of the same size as the original dataset. Furthermore, at each split point just a random subset of variables is considered as split variable. The split points are chosen based on an information criterion, like the Gini impurity.

CTree and CForest: The conditional inference forest consists of a set of many conditional inference trees, developed by Hothorn *et al.* (2006). It is similar to CART or random forests. The main difference is the choice of split points. In CART / random forests, an information measure is maximised while ctrees / cforests use a permutation-based significance test procedure to select the variables. Hence, ctrees/cforests are computationally more demanding.

The propensity scores for simulations 4-8 are evaluated with the RMSE. Simulations 1-3 are already randomised, so there is no need to estimate the propensity score. Since the propensity scores of simulations 4-6 and 7-8 are similar, they are summarised. Additionally, a distinction between including all the covariates in the model and just

including the confounder is made. The prediction of the propensity score for each scenario and method is repeated 50 times. That means that each model is fitted on 50 datasets from which the propensity scores are estimated. The datasets contain 1000 observations, respectively. The results are presented in Figure 6.1.

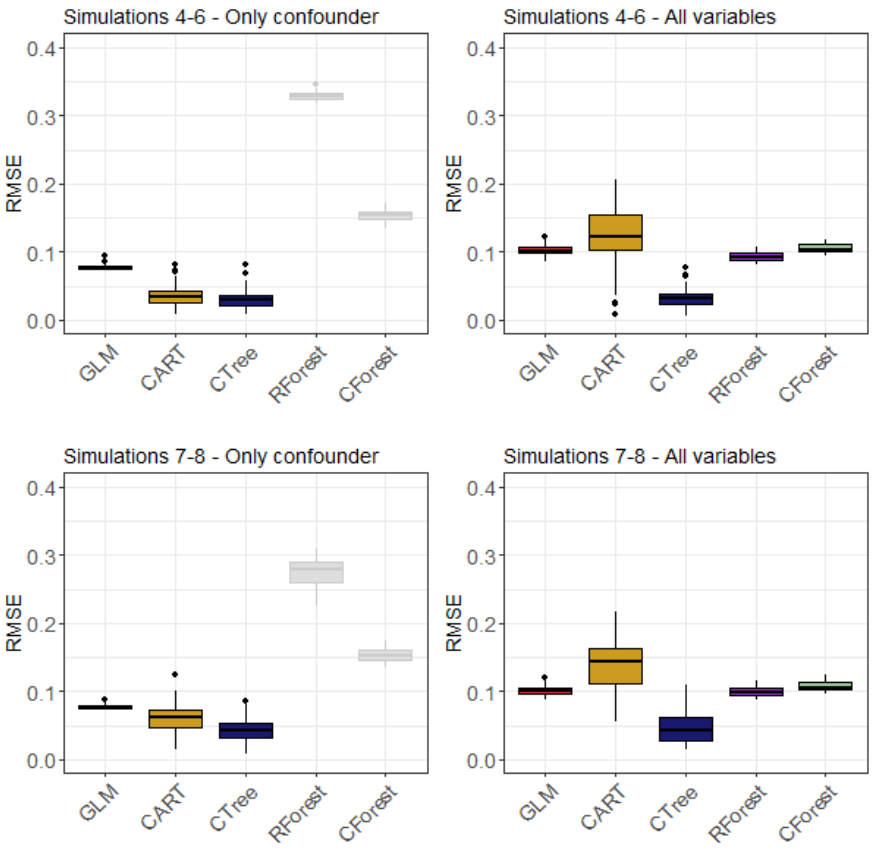

Figure 6.1: RMSE of propensity score models

On the left side, the RMSEs of the models with only confounders included are plotted. The right side shows the RMSEs of the models with all available variables included. The results are distinguished between simulations 4-6 and simulations 7-8, respectively. Naturally, it is not reasonable to fit a random forest or a cforest just with one or two covariates. Therefore, they are printed in grey for the case of only using the confounders. According to the RMSE, ctree is superior for all scenarios. Overall, it has the lowest RMSE. Especially for simulations 7-8 it fits better by just including the confounder instead of all covariates into the model. Thus, in the following, ctree is used to estimate the propensity scores. Moreover, the ctrees are fitted with just the confounders.

6.4 Computational Details

In this section, some details about the implementation and packages used for the previously explained methods are presented. For computations, the statistical software `R` (version 3.5.1) is used (R Core Team 2014). All plots are created with the `R` package `ggplot2` (version 3.0.0) (Wickham 2016).

6.4.1 Propensity Score Model

As illustrated in Section 6.3, the propensity score is estimated by using a causal inference tree (ctree) with only the confounders as covariates. For this purpose, the function `ctree` of the package `party` (version 1.3-1) is applied (Hothorn *et al.* 2006). The cforest is fitted with the same package and the function `cforest`. The random forest is fitted with the function `randomForest` of the package `randomForest` (version 4.6-14) and CART with the function `rpart` of the package `rpart` (version 4.1-13). For the GLM the `glm` function of the package `stats` (version 3.5.1) is utilised. All methods are applied with default hyperparameter settings because tuning would go beyond the scope of this thesis.

6.4.2 Outcome Model

After fitting and choosing the propensity score model, the treatment effects are estimated. The packages utilised to fit the outcome models are described in this section.

6.4.2.1 GLM Tree with IPTW

In order to use a model-based recursive partitioning based on generalised linear models, the `glmtree`-function of the R Package `partykit` (version 1.2-2) was used (Hothorn & Zeileis 2015). This is a new implementation of the general model-based (MOB) recursive partitioning algorithm. The `R` package can be utilised for several different models, like OLS regression or survival regression. Since the response is binary in the present thesis, a generalised linear model (a logit model) is applied. Thus, in each leaf of the tree a GLM (with `family = binomial` and `link = logit`) is fitted.

To fit the model, three different types of variables are available: the response y (the outcome), the regressor t (the treatment) and the partitioning variables X (all covariates).

To get unbiased results from observational data, the observations are weighted by the inverse probability of receiving a treatment (see Section 4.1). In `R`, the argument `weights` can be added. For all other parameters, the default settings are applied.

The model is implemented as follows:

```
glmt <- glmtree(y ~ t | x1 + ... + xp, data = ..., weights = ...,
                family = binomial(link = "logit"))
```

6.4.2.2 GLM Tree with Matching

For the matching procedure, the `R` package `MatchIt` (version 3.0.2) is used. It contains implementations of the suggestions of Ho *et al.* (2007). It is designed for causal inference with dichotomous treatment variables. The new matched dataset is generated with the following function:

```
matcheddata <- matchit(t ~ ., data = ..., method = "nearest",
                       distance = "logit", m.order = "random",
                       caliper = 0.2, discard = "none")
```

The nearest neighbour matching as described in Section 4.2 is used. It is specified by `method = nearest`. By determining the hyperparameter `distance = "logit"`, the matching is performed based on the propensity score. The propensity scores are estimated with a logistic regression in a first step. Matches are chosen step by step for each treated unit, in the order specified by the `m.order` command (here: random). With `caliper = 0.2`, the matched treated and control subjects are always within the standard deviation of 0.2 of the distance measure. Figure 6.2 shows an example of a matched dataset of simulation 4.

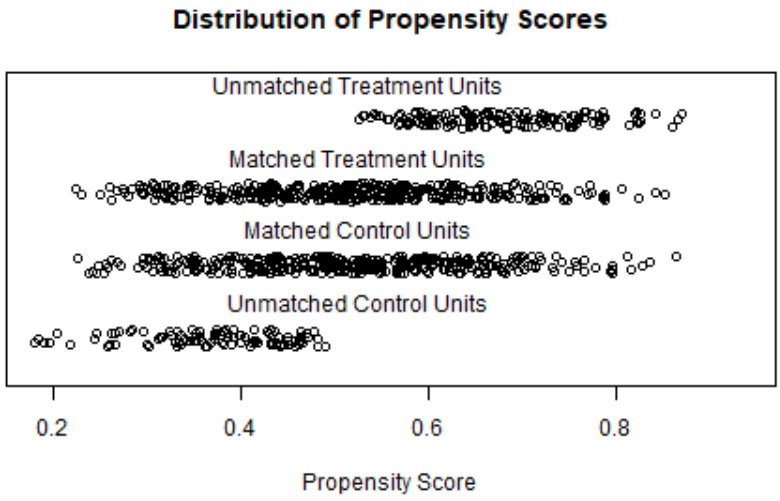

Figure 6.2: Jitterplot of matched dataset

The plot shows a similar propensity score distribution of the treatment and control units after the matching procedure. This helps to reduce confounding. The unmatched observations in both treatment groups are excluded from the analysis. The matched dataset is then used to fit a GLM tree, as defined in the previous section.

6.4.2.3 Causal Tree

Causal trees are implemented in the package `causalTree` (version 0.0) (Athey *et al.* 2016a). To fit honest causal trees, the following code is applied:

```
ct <- causalTree(y ~ ., treatment = t, data = ...,
                 cv.Honest = TRUE, split.Honest = TRUE,
                 split.Rule = "CT", cv.option = "CT",
                 split.Bucket = TRUE, weights = ..., maxdepth = ...)
```

The function expects a formula with response and features and a vector with the binary treatment status of each observation (dummy-coded). To fit the honest version of splitting and cross-validation, `cv.Honest`, as well as `split.Honest` should be set to `TRUE`. A causal tree (instead of a transformed outcome tree, a fit-based tree or a squared t-statistic tree) is achieved by setting the `split.Rule` and the `cv.option` to `CT`. The hyperparameter `split.Bucket = T` ensures that a discrete method for splitting the

tree is applied. Furthermore, the observations in a leaf will be partitioned into buckets. This prevents unnecessary splitting. Without this hyperparameter, the tree might split too often on covariates that have a strong influence on the level of the outcome. Imagine an observation in the treatment group. The expected estimated difference in treatment effects across the left and right leaves fluctuates greatly with the split point. The reason is that it just influences the average of the treatment group. This leads to an increased variance of the estimated treatment effect. As in GLM trees, weights are included by a `weights` parameter.

Pruning the tree by a complexity parameter is not possible since no minimum in the 10-fold cross-validated relative error (xerror) can be determined. Consequently, according to the complexity parameter, the tree should not be pruned at all. To guarantee trees with an adequate size, the maximal depth was fixed beforehand using the hyperparameter `maxdepth`. Thus, the causal tree has a small advantage over the other methods because the number of expected leaves is specified in advance.

6.4.2.4 Causal Forest

The generalised random forests are implemented in the `grf` package (version 0.10.1). This package provides non-parametric methods for least-squares regression, quantile regression, and treatment effect estimation (Tibshirani *et al.* 2018). The causal forest function in this package provides an honest version as well as a tuning function. This function tunes the *minimal node size*, *mtry* (number of variables tried for each split), *alpha* (controls the maximum imbalance of a split) and the *imbalance penalty* (parameter to control the extent of penalisation in imbalanced splits). Theoretically, the fraction of data used to build each tree can be tuned. Since the honest version is applied, the fraction is fixed to 0.5. With the following code, a causal forest is fitted:

```
cf <- causal_forest(X = X, Y = y, W = t, W.hat = ..., Y.hat = ...,
                    min.node.size = ..., sample.fraction = ...,
                    mtry = ..., alpha = ..., imbalance.penalty = ...,
                    tune.parameters = TRUE, honesty = TRUE)
```

The outcome y, as well as the covariates X and the treatment assignment t have to be numeric in this function. For all factor variables 0/1 dummy encoding can be used. For the covariates X, this is implemented in the R-Function `model.matrix`. To improve the

fit, `W.hat` and `Y.hat` functions are included. They estimate the treatment propensities, as well as the expected response using a separate regression forest. With these estimates, local centering is performed and the causal forest is fitted on the centered outcomes. If `W.hat` and `Y.hat` are set to zero, no centering is performed.

The package `grf` is still in beta version. The tuning function does not tune all variables and it is not yet completely documented. The tuning results can be found in the appendix in Table A.1 and Figure A.1. For each forest, the parameters are tuned anew. According to the tuning results, the alpha parameter is relatively small for stepwise functions (simulations 2, 5 and 7). In the case of no heterogeneity (simulations 1 and 4), the minimal node size gets large. For linear functions (simulations 3, 6 and 8), the mtry parameter gets large. The imbalance penalty is quite similar for all simulations.

6.4.2.5 BART

To fit a bayesian additive regression tree (BART), the function `pbart` of the package `BART` (version 1.9) is utilised (McCulloch *et al.* 2018). This function fits BART for a binary response.

In this package, the number of posteriori draws returned (`ndpost`), as well as the number of MCMC iterations used as burn in (`nskip`) have to be defined. The covariates are defined in the parameter `x.train` where also the treatment assignment is included. `y.train` contains the outcome variable. The function returns a matrix with *ndpost* rows (draws from the posterior) and n (number of observations) columns. To obtain one value per person, the columns are averaged. In the binary case, the probit link of the posterior probability $\mathbb{P}(Y = 1|t, x)$ ($\hat{=}\mu^t(x)$) is required. The default setting is used as prior as recommended by Chipman *et al.* (2010). The code to fit a BART model is the following:

```
bart <- pbart(x.train = X_train, y.train = y, nskip = 500, ndpost = 1000)
```

There are also other packages that provide BART computations. One of them is the `dbarts` package. It is much faster and allows more complexity (e.g. random effects) than the `BART` package. Another package is `bartCause`, developed by Jennifer Hill and Vince Dorie. It can handle all relevant things concerning causal inference. Furthermore, it can print out individual treatment effect distributions. However, a problem of these packages is their limitation to a binary response.

6.4.2.6 PTO Forest

The PTO forest algorithm is implemented in the `R` package `causalLearning` (version 1.0.0) (Powers *et al.*). This package contains methods developed by Powers *et al.* (2018). To apply IPTW, the propensity score has to be passed to the function (`pscore`). The default value for the propensity score is 0.5 for all observations. All covariates need to be numeric in this package. For the binary case, 0/1 dummy encoding can be used. The following code fits a PTO forest:

```
PTOf <- PTOforest(x = X, t = t, y = y, pscore = ...)
```

6.4.2.7 Causal MARS

Like the PTO forest, the causal MARS algorithm is implemented in the package `causalLearning` (Powers *et al.*). The bagged version of the causal MARS is computed with the following code:

```
cM <- bagged.causalMARS(x = X, t = t, y = y,
                        propensity = TRUE, stratum = ...,
                        backstep = TRUE, nbag = 20)
```

The function expects a matrix of covariates x, a vector of treatment indicators t and a vector of response values y. Just like in PTO forests, numeric variables are required in this package and in the binary case, a 0/1 dummy encoding can be used. The hyperparameter `nbag` defines the number of models to bag. To adjust for confounding, data is stratified beforehand. Afterwards, the strata is given to the function with the `stratum` hyperparameter. To guarantee the use of propensity score stratification, `propensity` is set to `TRUE`. Each model is pruned based on OOB samples by setting the hyperparameter `backstep`.

6.5 Results

The performance of the eight scenarios is evaluated by refitting the model and predicting 100 times for each method. In each iteration, a new train and a new test dataset are generated. The predictions are calculated for the respective test dataset. The training dataset is 4 times as large as the test dataset. For $n = 1000$ in the train dataset, the test dataset would contain 250 observations. The resulting RMSEs and biases are graphically presented in Figures 6.3 - 6.10. The corresponding variances are displayed in the appendix in Figures A.2 and A.3.

6.5.1 Simulation 1

In simulation 1, no confounders and a similar treatment effect for all observations are assumed. Thus, all methods are used without an adjustment for confounding. Figure 6.3 shows boxplots of the RMSEs for all methods on the left side and of the biases on the right side. The results are divided into the different numbers of observations, respectively. According to the RMSEs, the GLM tree and the causal forest perform best in this scenario. For $n = 300$, the RMSEs of the causal tree and causal MARS are the worst. The difference to the other methods is relatively small. However, their biases are close to zero. Thus, the variances of these two methods are higher than for the remaining ones (see Figure A.2 in the appendix). For all methods, especially for causal MARS, the RMSE gets better for a higher number of observations. PTO forest and BART are the most biased methods in this simulation.

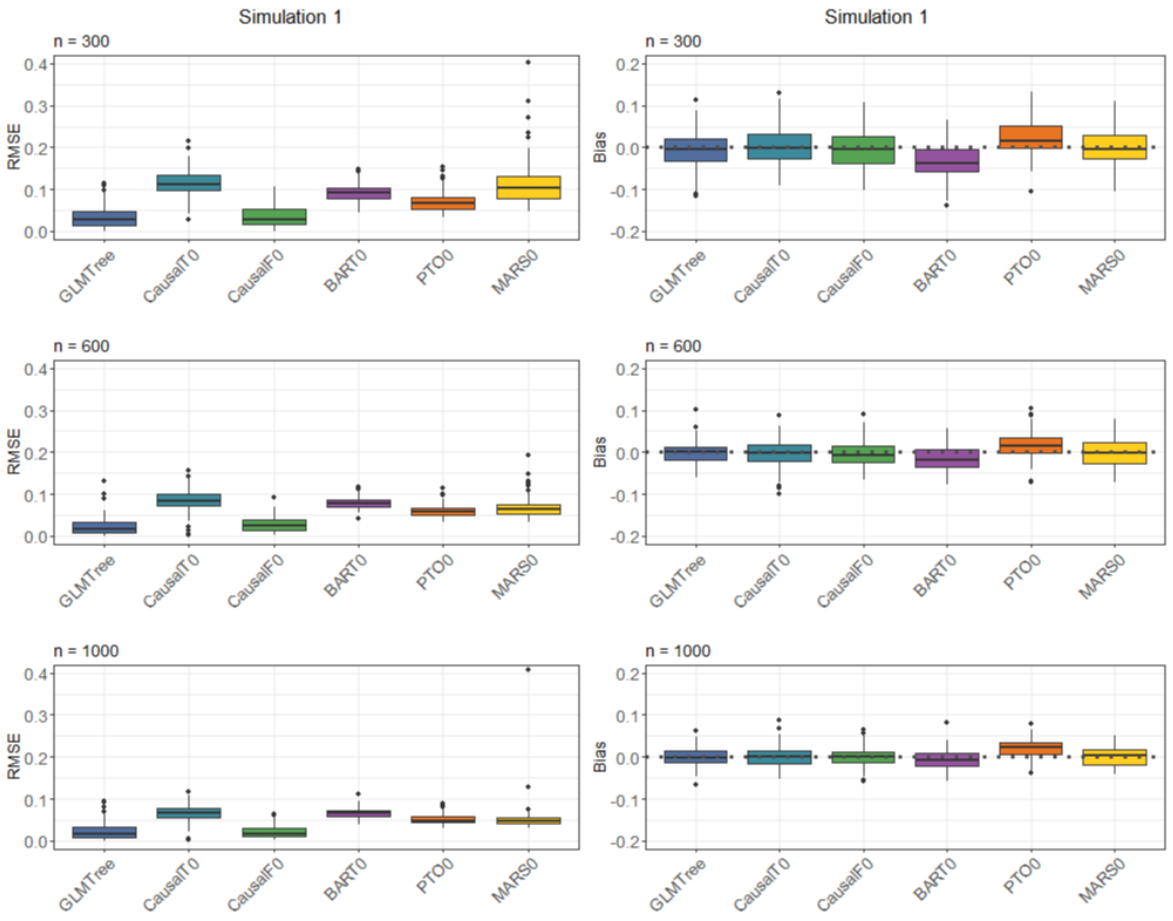

Figure 6.3: RMSE and bias of different methods for simulation 1

6.5.2 Simulation 2

The results for simulation 2 are depicted in Figure 6.4. There are still no confounders but heterogeneous treatment effects, constructed with a stepwise function. Thus, the methods are used without adjustment as before. In general, the methods perform almost similar and none of them is strongly biased. For $n = 300$ and $n = 600$, the causal tree has a large interquartile range in the RMSE. However, for $n = 1000$, it performs best. The other methods seem to be more stable. The causal tree has the highest variance in this scenario for all n (see Figure A.2).

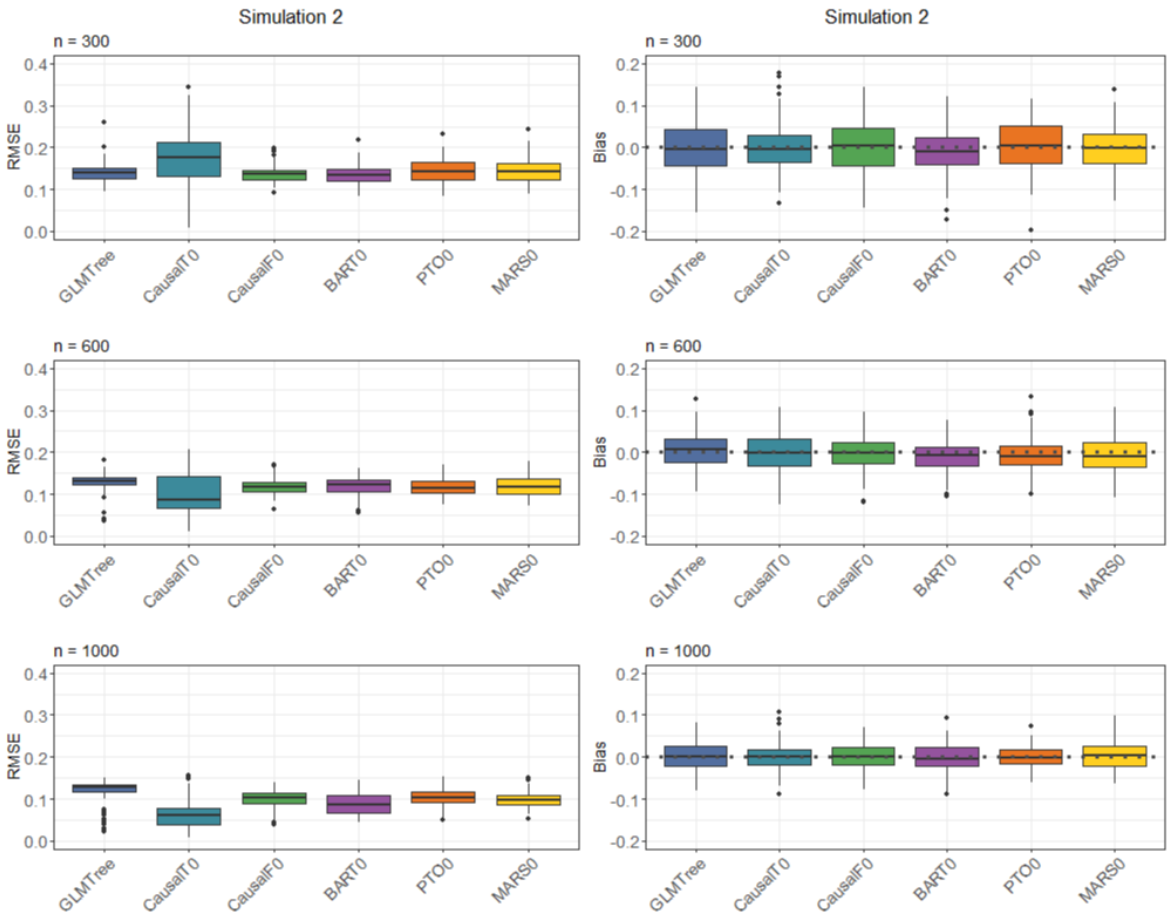

Figure 6.4: RMSE and bias of different methods for simulation 2

6.5.3 Simulation 3

Except for a linear treatment effect function, the data of simulation 3 is similar to simulation 2. Thus, no adjustment for confounding is necessary. As illustrated in Figure 6.5, the RMSE is generally higher than in simulation 2. Especially the causal tree deteriorates extremely. However, all biases are close to zero. This implies a high variance for the results of the causal tree. Also for the other methods, the variances are much higher than in simulation 2 (see Figure A.2). For the GLM tree, the interquartile range of the variance is large for small n. Nevertheless, it gets smaller with an increasing number of observations. BART has the lowest variance for small n. For large n, the variance of the causal forest is the lowest. The most (but not strong) biased method is the PTO forest. The RMSE and variance for the causal tree exceed the y-axis.

Therefore, the values are illustrated with an extended y-axis in Figure A.4 in the appendix.

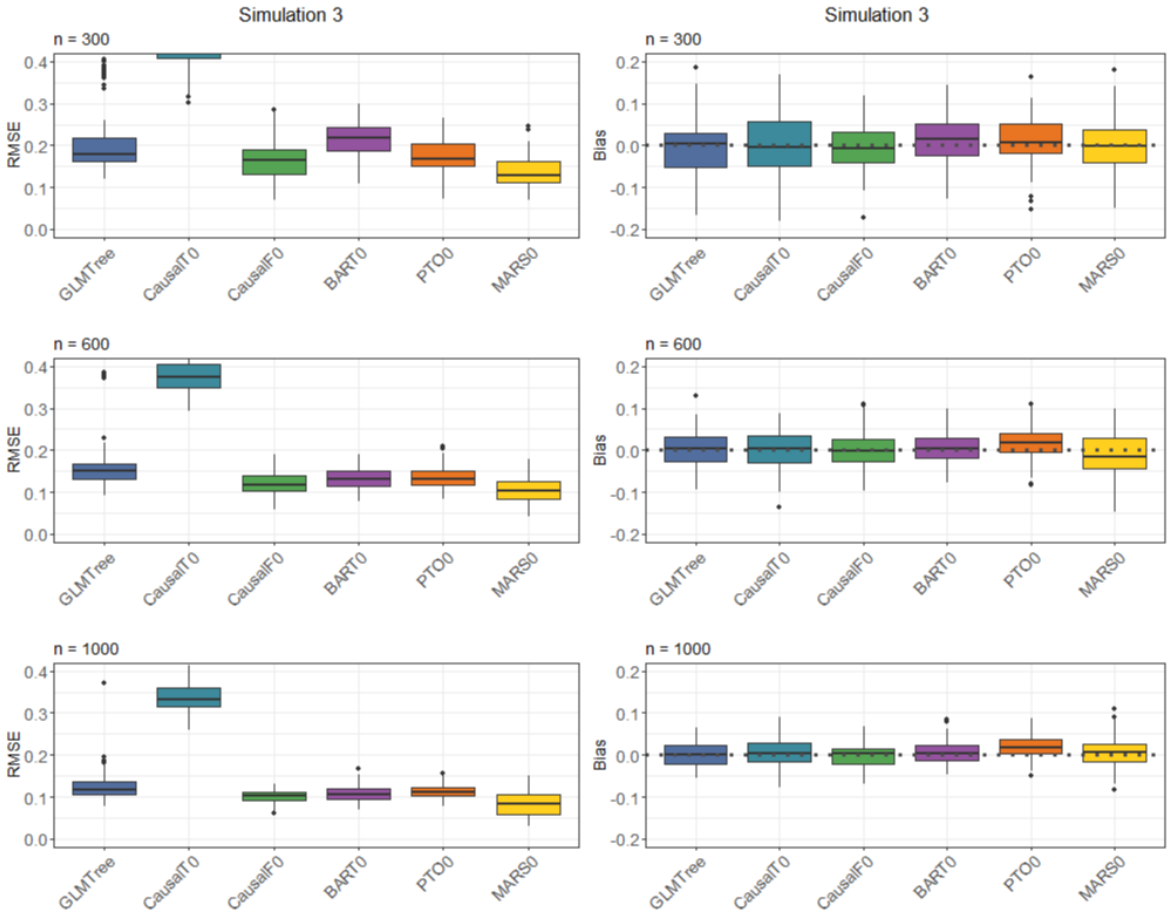

Figure 6.5: RMSE and bias of different methods for simulation 3

6.5.4 Simulation 4

Simulation 4 is the first scenario with confounding. Thus, the comparison between the methods with and without adjustment is meaningful. The methods without adjustment are labelled with a "0" at the end. Hence, it is expected that the methods without a "0" have a lower RMSE and a bias close to zero. For the GLM tree, two different propensity score adjustments are investigated: Matching and IPTW.

In simulation 4, the treatment effects are not heterogeneous. According to Figure 6.6 the RMSE is relatively low for all methods. The approaches with adjustment do not

perform better considering the RMSE except for the causal tree and the causal forest. The same applies for the bias. One exception is the bias of the causal MARS model. It is reduced by propensity score stratification. Additionally, matching and IPTW in the GLM tree reduce the bias. The improvement of the causal MARS model with the propensity score stratification is not visible in the RMSE because of many extreme values in the variance (see Figure A.2). The same applies for the GLM tree, where the IPTW adjustment and matching result in slightly increased variance for large n. The bias of the BART is getting negative with propensity score adjustment. Without the adjustment, the bias is closer to zero.

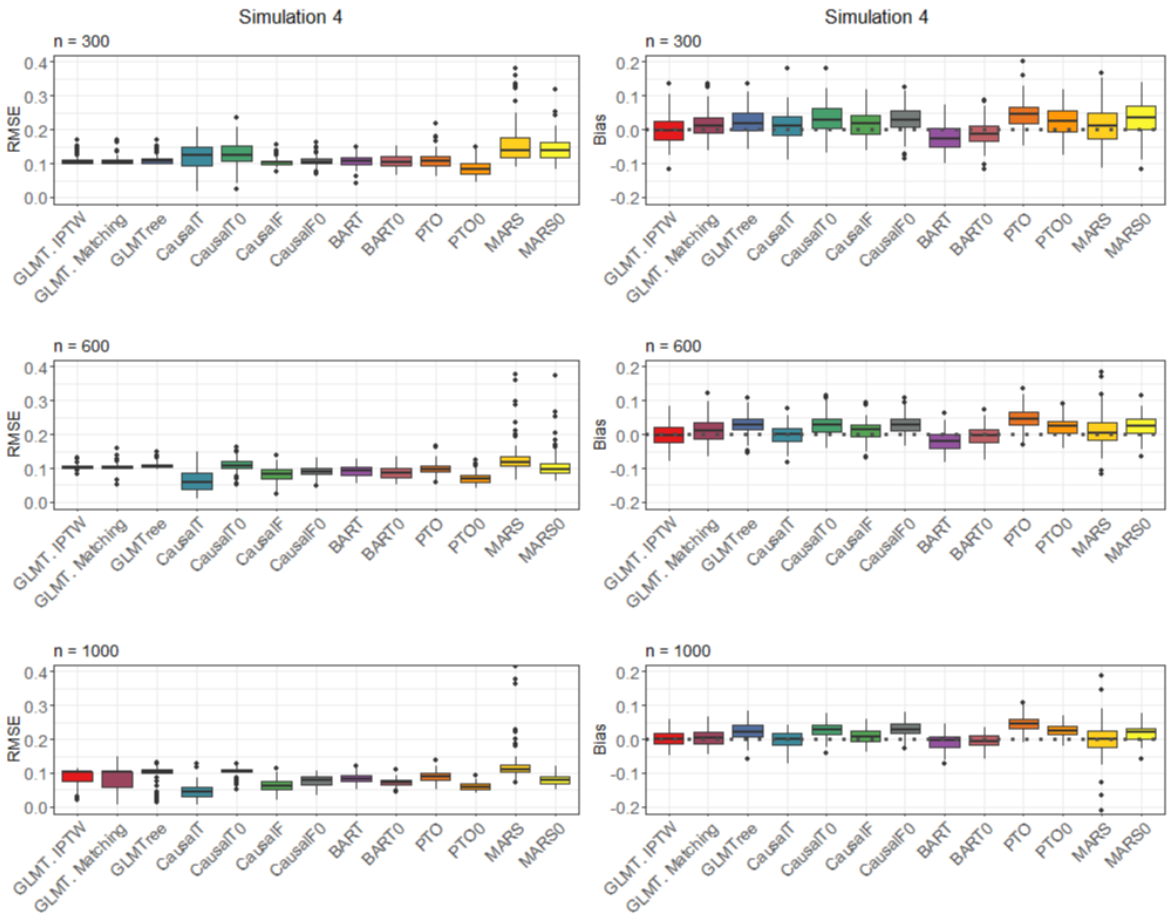

Figure 6.6: RMSE and bias of different methods for simulation 4

6.5.5 Simulation 5

In the case of one confounder and heterogeneous treatment effects, generated with a stepwise function, the causal tree is clearly performing worse compared to the other methods (see Figure 6.7). The RMSE only slightly differs between the values with and without adjustment. Nevertheless, the causal tree, the causal forest, and the causal MARS are clearly less biased with the adjustment. Especially for the causal tree, the bias is getting closer to zero. That indicates a relatively high variance for the causal tree with adjustment (see Figure A.3). Thus, the bias reduction is not visible in the RMSE anymore. As before, the RMSE of the causal MARS contains many outliers.

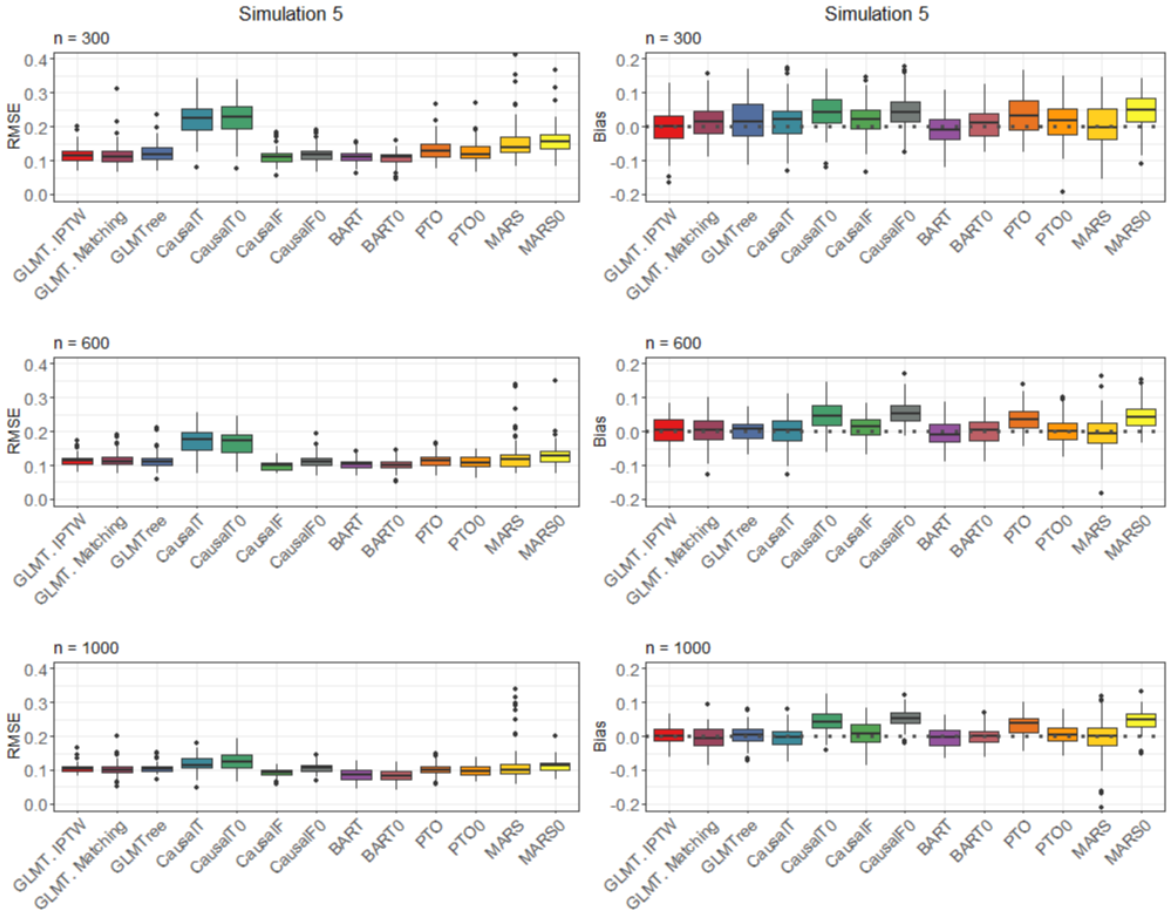

Figure 6.7: RMSE and bias of different methods for simulation 5

6.5.6 Simulation 6

Except for the linear treatment effect function, the dataset in simulation 6 is similar to simulation 5. As illustrated by Figure 6.8 the RMSEs are higher compared to simulation 5, especially for small n. Causal MARS has the lowest RMSE values for $n = 300$ and 600. Correspondingly, it can handle stepwise functions as well as linear functions for small sample sizes. For $n = 1000$, the RMSEs of all methods are similar. One exception are the causal trees that perform worse with the linear function. They boast high variances in the estimates, which is reflected in the RMSE results. For small n, GLM trees have a large interquartile range in the RMSE due to the high variance. As before, the bias is close to zero for almost all methods. Except for the PTO forest, the adjustment is reasonable and results in less bias. In the variance plots (see Figure A.3) more differences between the methods are visible. BART shows the lowest variance for small n, followed by causal forest, PTO forest, causal MARS, GLM tree and causal tree. For $n = 1000$, causal forest has the lowest variance. The variances for the methods with propensity score adaptation are slightly higher than without. Thus, the improvement of the bias is not reflected in the RMSE. As in simulation 3, the RMSE and variance for the causal tree exceed the y-axis. For this reason, the plots are printed with an extended y-axis in Figure A.4 in the appendix.

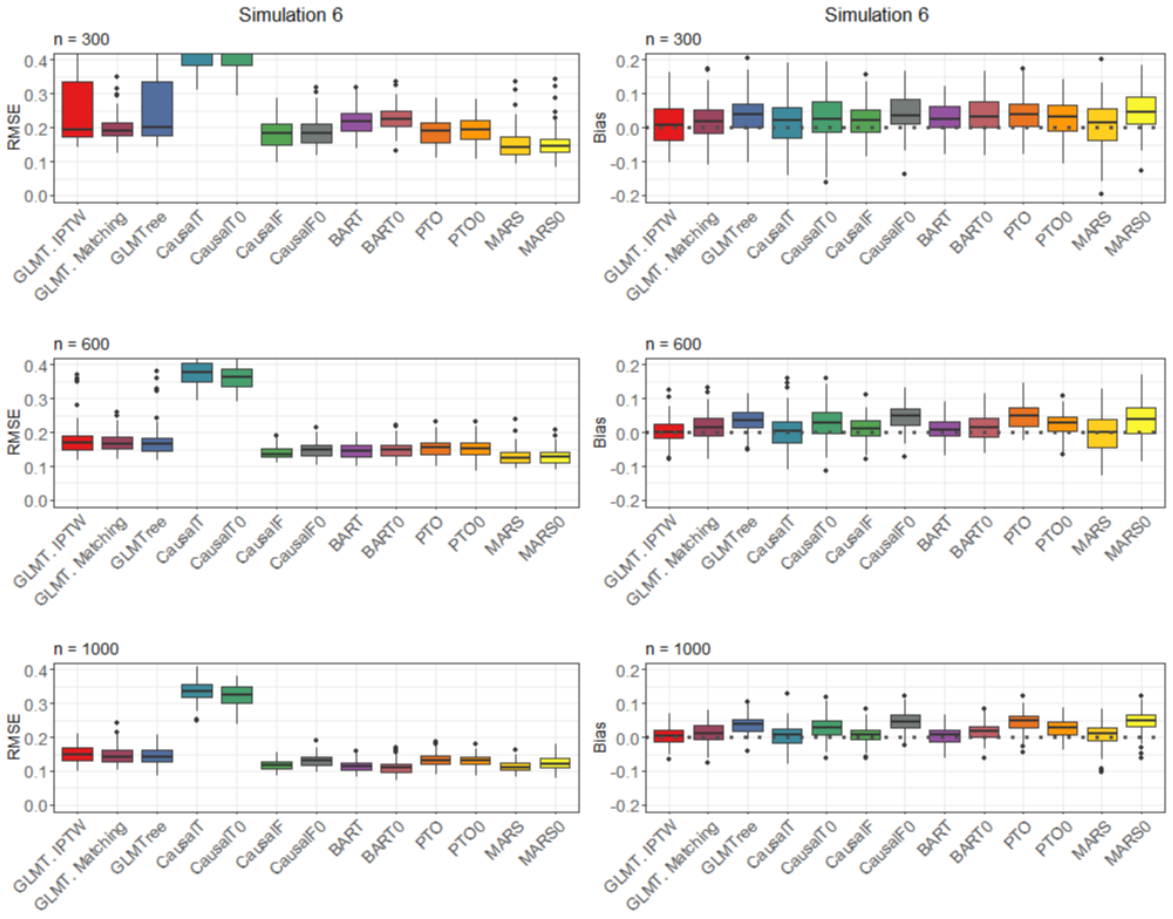

Figure 6.8: RMSE and bias of different methods for simulation 6

6.5.7 Simulation 7

A second confounder is included in the dataset of scenario 7. Additionally, the stepwise treatment effect function is now influenced by two variables. As depicted in Figure 6.9, the causal tree shows the worst performance with respect to the RMSE. This is caused by its high variance (see Figure A.3). Moreover, the causal MARS model has many outliers in the RMSE. The RMSE is the lowest for GLM tree and BART. The methods with and without adjustment hardly differ. The bias is slightly improving with adjustment for all methods except for PTO forest and BART. For large n, the GLM tree is not biased even without the IPTW. This has not been the case with a linear treatment effect function (simulation 6).

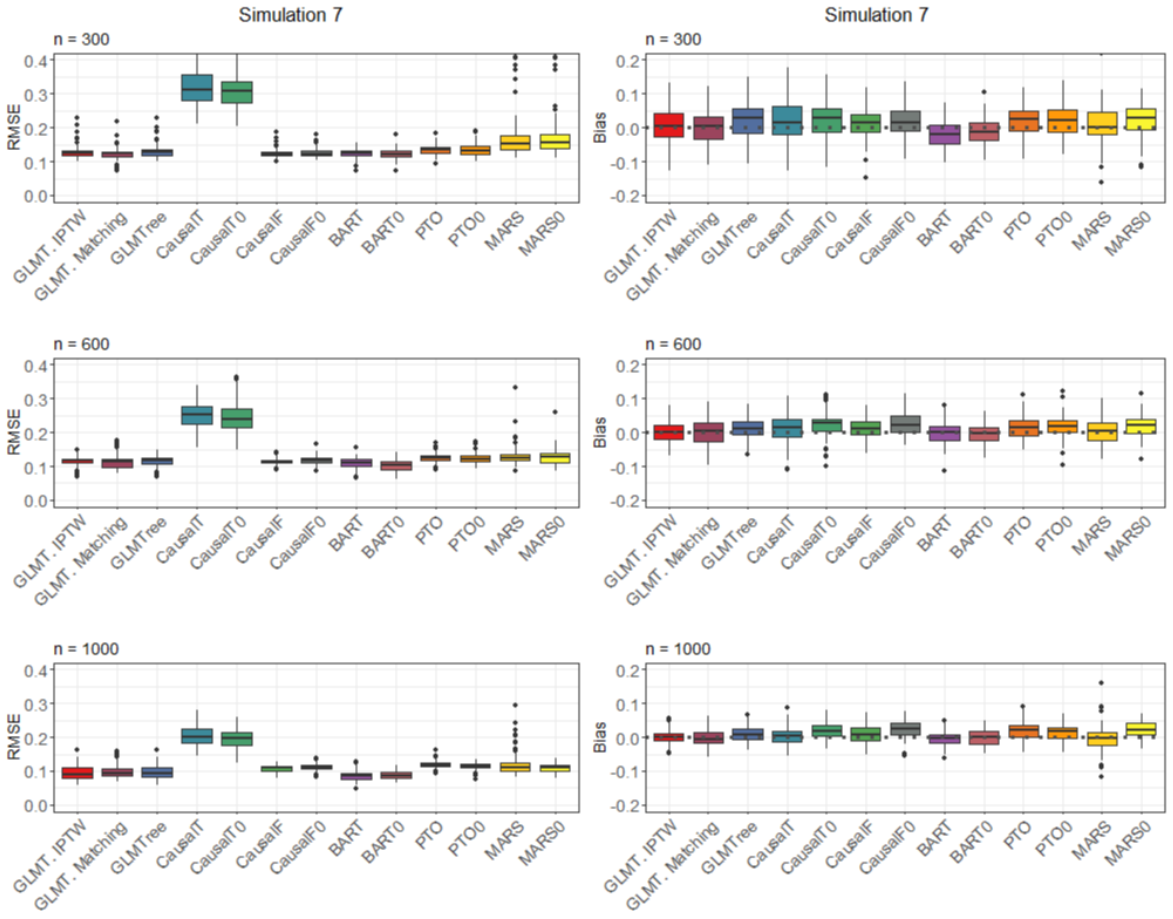

Figure 6.9: RMSE and bias of different methods for simulation 7

6.5.8 Simulation 8

The results of simulation 8 are illustrated in Figure 6.10. In contrast to simulation 7, the treatment effect function is linear and the RMSEs are generally higher. In this scenario, the performance of the distinct methods differs more. For a small sample size, causal MARS has the lowest RMSE, followed by PTO forest, causal forest, BART and GLM tree. The causal tree provides the worst fit. For $n = 1000$, BART's performance improves and is similar to the performance of causal MARS. The adjustment for confounding shows a little improvement in the bias for all methods, except for the PTO forest. The propensity score adjustment for BART is only effective for large n. The variance plot (see Figure A.3) illustrates much higher variances than in simulation 7. For $n = 1000$, the causal forest has the lowest variance, followed by the PTO forest,

BART, causal MARS, GLM tree and causal tree. The high variances in causal trees lead to poor results in the RMSE. Moreover, for small n, the variances of the GLM tree have a very large interquartile range. Additionally, the variances of BART are the lowest for small n. In general, adjustment lowers the bias (except for PTO forest and BART) but increases the variance. Consequently, the RMSE is not improved by the adjustment. As in simulations 3 and 6, the RMSE and variance for the causal tree are not completely visible, because they exceed the y-axis. Thus, they are plotted with an extended y-axis in Figure A.5 in the appendix.

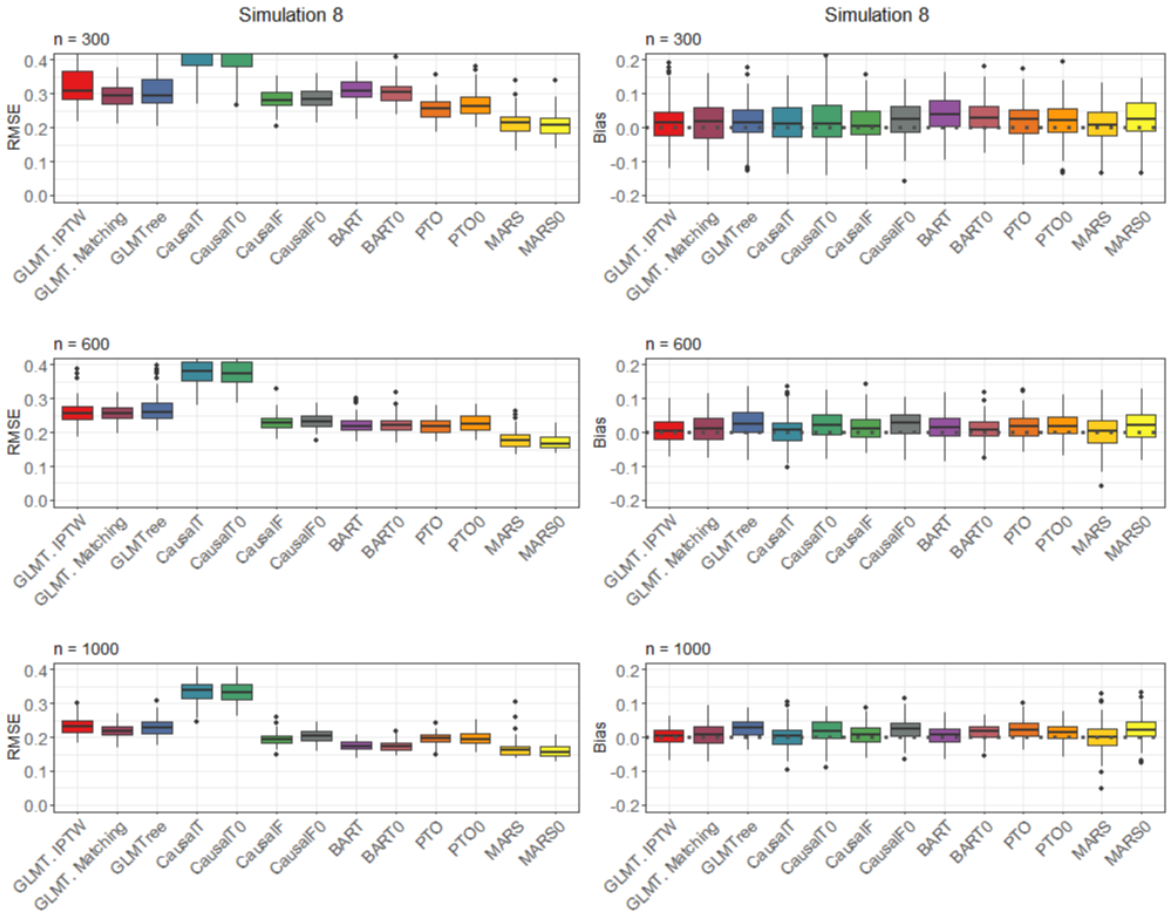

Figure 6.10: RMSE and bias of different methods for simulation 8

6.5.9 Overall Remarks

Some remarks to these results are true for all eight simulations. First, the interquartile range of the RMSE and bias is getting smaller with an increasing number of observations. Second, there is a clear distinction between linear and stepwise treatment effect functions. The variances (see Figures A.2 and A.3) are large for linear treatment effect functions (simulations 3, 6 and 8). Furthermore, the interquartile range of the variance for the GLM trees is extremely wide for small n. The large variances have an impact on the RMSEs. The RMSEs and variances of the causal tree are very large for the simulations with linear treatment effect functions and exceed the y-axis. Therefore, the values are displayed with an extended y-axis in the appendix in Figures A.4 and A.5. Additionally, the adjustments for confounding cause less biased results in most cases, but slightly increase the variance. For large n, causal forest has the lowest variance and for small n, BART has the lowest variance. This is most apparent for a linear treatment effect function.

6.5.10 Visualisation of the Estimated Treatment Effect

For a visualisation of the approximation to the true treatment effect function, the true and the estimated treatment effect values in dependence on the variable x_1 are plotted (see Figures A.6 - A.19). The variable x_1 is the covariate influencing the treatment effect in simulations 2-8. In simulations 7 and 8, variable x_{11} is an additional confounder. In simulation 1, the treatment effect is constant for all observations. Simulations 1-3 are generated without and 4-8 with adjustment for confounding. The GLM tree cannot handle linear treatment effect functions, especially for small n. Due to its nature, the GLM tree can only estimate stepwise functions. Additionally, the estimations of the causal tree vary widely around the true values for all simulations. All other methods approximate the true treatment effect function better.

6.5.11 Running Time

Another important aspect in the evaluation of methods is the running time of the algorithms. Table A.2 shows the average running time in seconds for the different methods. The table is split into the different numbers of observations. Figures A.20

and A.21 visualise these values. For simulations 4,5,7 and 8 the running time of causal MARS with adjustment is not completely visible. Thus, these plots are printed with an extended y-axis in Figure A.22 in the appendix. In general, the algorithms are taking longer with an increasing number of observations. Particularly the running time of the GLM tree strongly increases with large n and a linear treatment effect function. The GLM tree fitted with a matched dataset is slightly faster. The causal tree is one of the fastest methods in all simulations. However, according to the RMSEs this method performs poorly in the case of confounding. Without the local centering approach, the causal forest is one of the fastest methods. With local centering, the algorithm takes much longer. The slowest method is causal MARS, especially with propensity score adjustment. Compared to the other methods, the running time of PTO forest and BART are average. The running time of PTO forest is not affected by an adjustment for confounding. For BART, the running time with propensity score adjustment is getting slightly worse. The time of the propensity adjusted BART model is in fact even higher than illustrated here. The reason is that the propensity scores themselves are estimated by a BART model. Hence, the running time almost doubles. However, the propensity score can be also estimated with a faster method.

6.5.12 Number of Nodes in the GLM Tree

An IPTW weighted GLM tree is a new method to estimate personalised treatment effects from observational data. To further examine the performance of the GLM trees, the expected and the true number of leaves in each scenario are counted. Figures A.23 and A.24 in the appendix show the frequencies of the number of nodes for all simulations. Simulations 1-3 are fitted without and simulation 4-8 with adjustment for confounding. The expected value is marked by the blue dotted line in each plot. According to these plots, the number of nodes meets expectations in almost all cases for simulation 1. Most of the trees in simulation 2 and 4 have only one node, whereby two are expected. This might be due to the manual pruning after the splitting procedure. An increasing number of observations leads to slightly more trees with two nodes. This means the expected value is reached more often. For a large number of expected nodes, (see simulation 5 and 7), the resulting trees become too small. Obviously, no expected number of nodes can be specified for simulation 3,6 and 8 with a linear treatment effect function.

6.6 Further Analyses of GLM Trees

In contrast to the other presented methods, the performance of IPTW weighted GLM trees has never been investigated before. Hence, additional analyses are presented in this section to assess the effectiveness of weights in GLM trees. For this purpose, some characteristics of the previous simulated datasets are modified. As before, the simulations are iterated 100 times.

6.6.1 Varying Propensity Scores

As a first step, the propensity score is changed. In the previous analysis, it was fixed to 0.65 if the covariate x_2 was smaller than 0 (and in simulations 7 and 8 if additionally x_{12} was equal to FALSE) and 0.4 otherwise. The application of IPTW in GLM trees showed a little improvement in these simulations. The weights are expected to be more effective for more imbalanced data. In this context, more imbalance corresponds to larger distance of the propensity score to 0.5.

To evaluate the correctness of this assumption, the distance to 0.5 is varied. Two different datasets, as described in Table 6.3, are simulated.

Simulation	$\mu(x)$	$e(x)$	$\tau(x)$
1	$0.5 + 1.2 \cdot I_{(x_2<0)}$	$\begin{cases} 0.5+d & \text{if } x_2 < 0 \\ 0.5-d & \text{else} \end{cases}$	$3 \cdot I_{(x_1>1)}$
2	$0.5 + 1.2 \cdot I_{(x_2<0)}$	$\begin{cases} 0.5+d & \text{if } x_2 < 0 \\ 0.5-d & \text{else} \end{cases}$	$3 \cdot x_1$

Table 6.3: Simulations with varying propensity score, with $d = 0.1, 0.2, 0.3$

The datasets are similar to simulation 5 and 6 that were described in Section 6.2. Both have one confounder. While one dataset has a linear treatment effect function, the other one has a stepwise treatment effect function. The only difference to the analysis of the previous section is the varying distance d to 0.5. For d, the values 0.1, 0.2, and 0.3 are chosen.

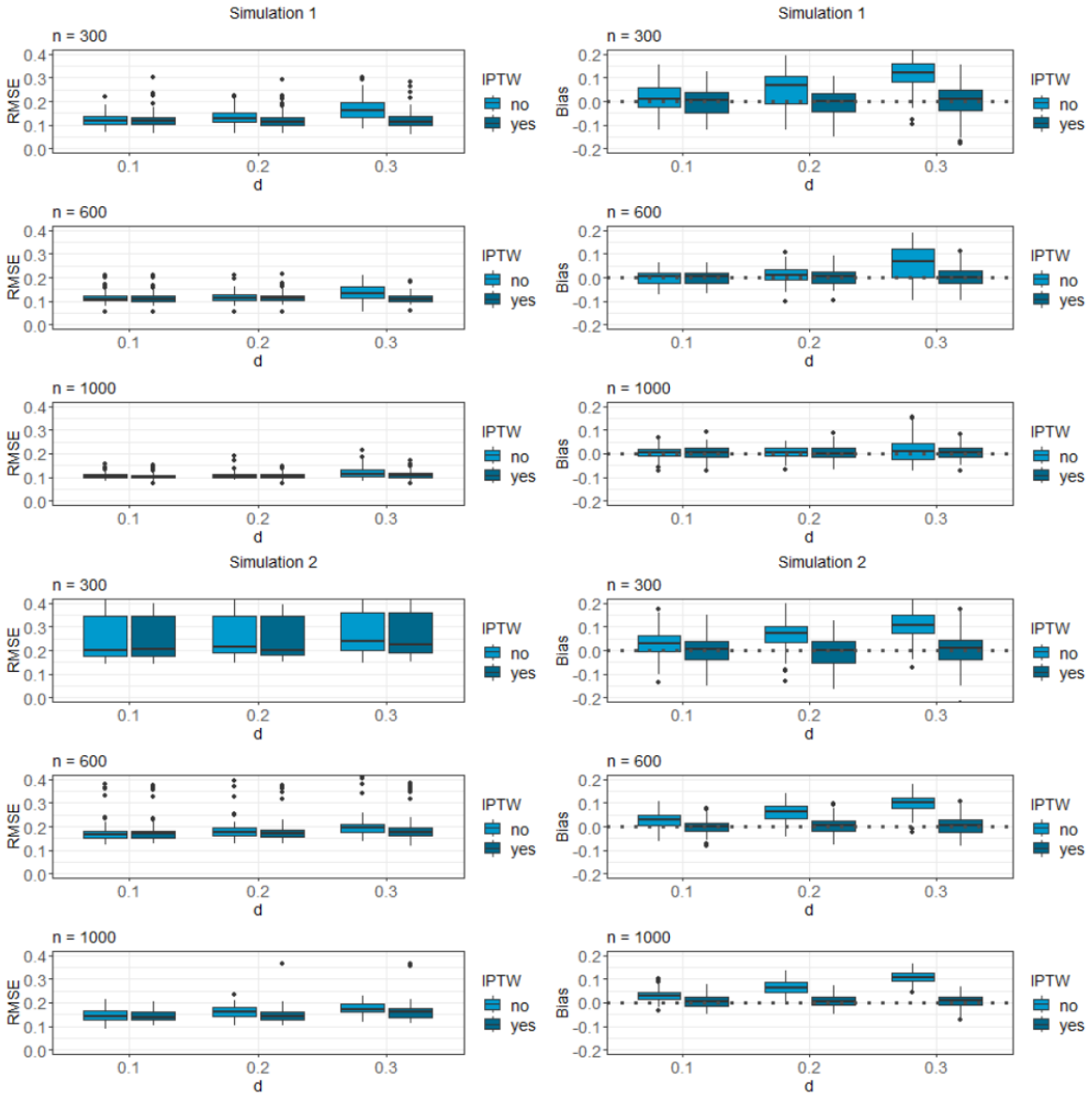

Figure 6.11: RMSE and bias of GLM trees for different propensity scores

Figure 6.11 shows the RMSE and bias of the GLM trees with varying propensity scores. The boxplots with $d = 0.1$ illustrate weighted and unweighted GLM trees with propensity scores equal to 0.6 and 0.4. The weights lead to a lower bias and thus lower RMSEs in both simulations. Moreover, with larger values of d, more biased models result. In simulation 1, a large n causes a relatively small bias for all values of d, even without weighting. In simulation 2, the variances and hence the RMSEs are higher than in the first simulation. Unlike simulation 1, the bias of the trees without IPTW is not

improving with an increasing number of observations. Thus, in this case, weighting is more essential than in simulation 1. The interquartile range of the RMSE in simulation 2 is large for small n. This is caused by the variance, where the interquartile is also large for small n. Furthermore, the variances are slightly larger for weighted than for unweighted trees. The variance is plotted in Figure A.25 in the appendix.

In accordance to Figure 6.12, an increasing number of observations results in a longer running time. In contrast, an increasing propensity score does not affect the running time. Especially in simulation 1 the running time for different values of d is quite similar. Simulation 2 identifies differences between the weighted and unweighted trees. The GLM trees adjusted by IPTW need more time for fitting. Moreover, the running time in simulation 2 is generally higher than in simulation 1. Therefore, the fit of GLM trees takes longer with a linear than with a stepwise treatment effect function.

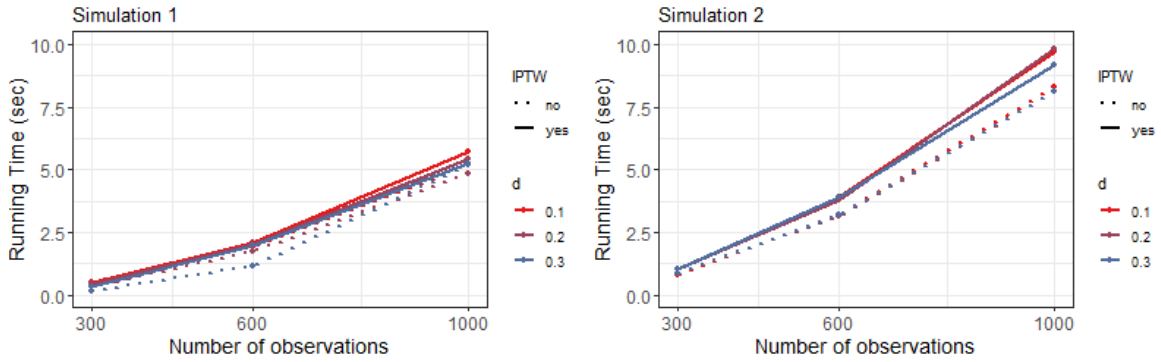

Figure 6.12: Running time of GLM trees for different propensity scores

6.6.2 IPTW with Varying Coefficient

To further examine the influence of the coefficients in the mean effect function $\mu(x)$ on the effectiveness of weights in GLM trees, the previous simulations are repeated with a varying coefficient c. The created datasets are illustrated in Table 6.4. For the propensity score, the distance d to 0.5 is fixed to 0.1. For the coefficient c, the values 1, 2 and 4 are tested.

Simulation	$\mu(x)$	$e(x)$	$\tau(x)$
1	$0.5 + c \cdot I_{(x_2<0)}$	$\begin{cases} 0.6 & \text{if } x_2 < 0, x_{12} = F \\ 0.4 & \text{else} \end{cases}$	$3 \cdot I_{(x_1>1)}$
2	$0.5 + c \cdot I_{(x_2<0)}$	$\begin{cases} 0.6 & \text{if } x_2 < 0, x_{12} = F \\ 0.4 & \text{else} \end{cases}$	$3 \cdot x_1$

Table 6.4: Simulations with varying coefficient $c = 1, 2, 4$

The results of these simulations are displayed in Figure 6.13. Generally, the values of the RMSE are higher for the linear treatment effect function (simulation 2). This corresponds to the results of the previous section. Furthermore, in simulation 2, the RMSEs get larger with an increasing coefficient value c. In both simulations, the RMSEs are similar for weighted and unweighted analyses. For a large value of c, the difference gets more pronounced. In simulation 1, the bias is reduced by weighting for small n and all values of c. For large n, the biases are close to zero, even without weighting. An exception to this is $c = 4$, where the difference between the biases with and without weighting does not change with an increasing n. The biases in simulation 2 are similar to those of simulation 1. However, the difference between the biases with and without weighting remains with an increasing n. Thus, weighting is more essential for linear than for stepwise treatment effect functions. The variances of both simulations, shown in Figure A.26, are getting smaller with an increasing c. This is most visible in simulation 2, where the variances are very large for small c. Furthermore, the weighting slightly increases the variances.

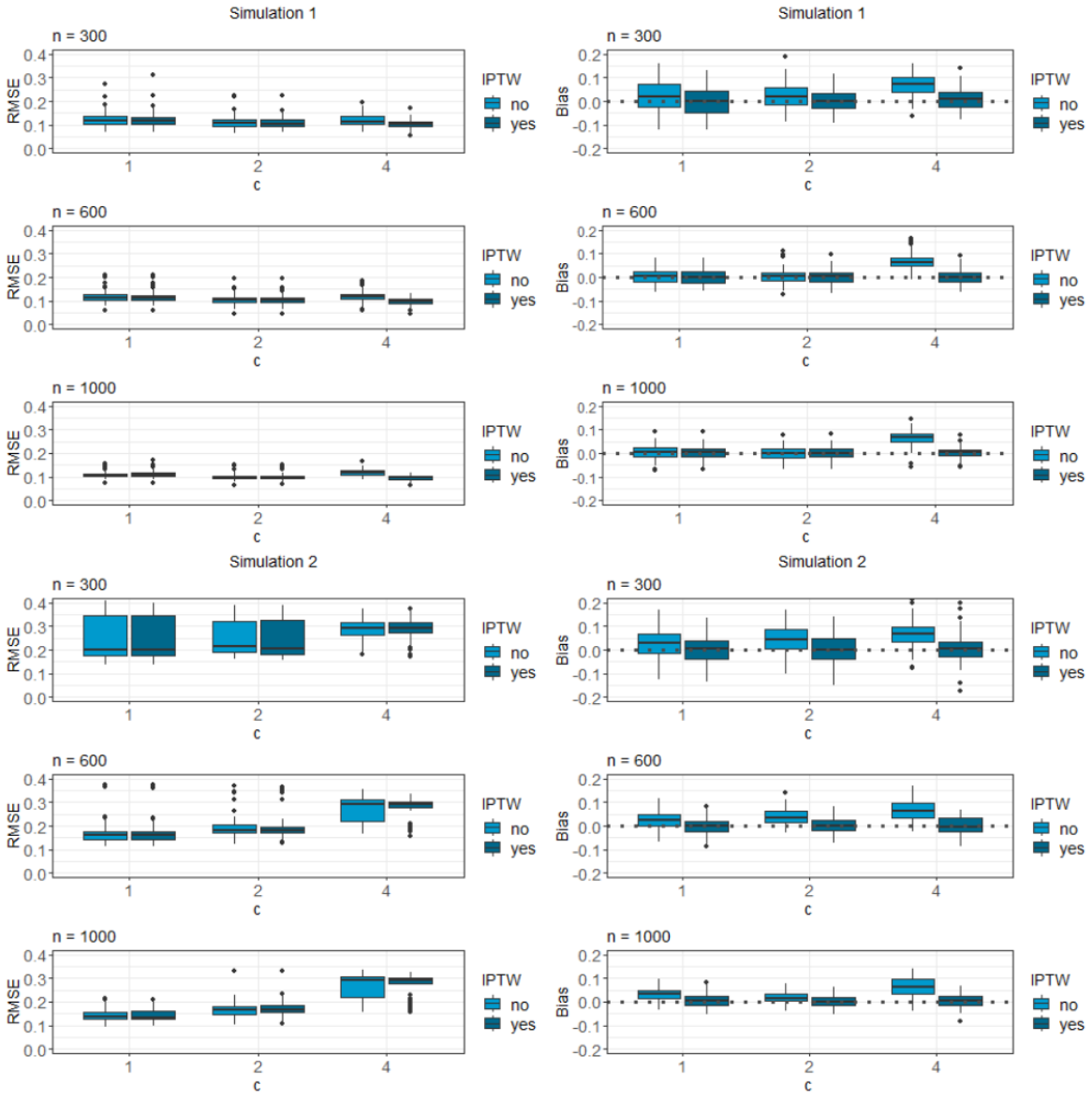

Figure 6.13: RMSE and bias of GLM trees for different coefficients

According to Figure 6.14, the running time increases with the number of observations in the dataset. Additionally, in simulation 1, a large value for c results in a higher running time. In simulation 2, this difference no longer exists.

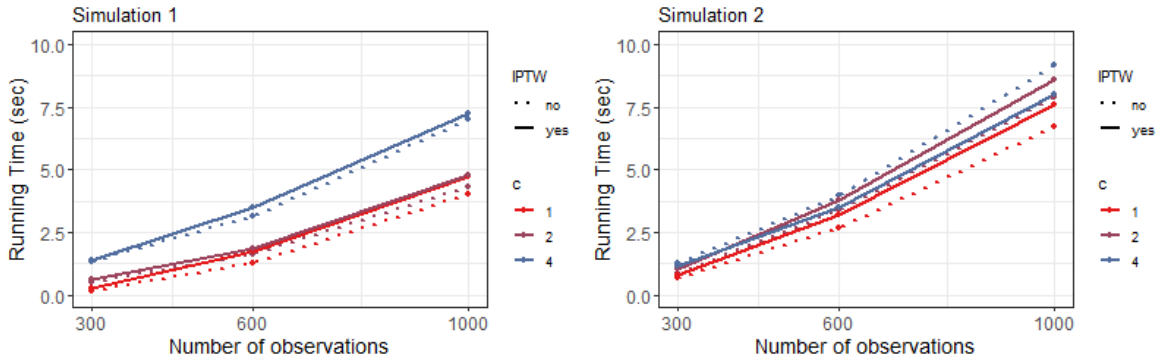

Figure 6.14: Running time of GLM trees for different coefficients

6.6.3 IPTW with Varying Treatment Effect

As a last step, the value of the treatment effect is modified. For this purpose, the same datasets considered before are used. d is fixed to 0.1 and c is fixed to 1. The treatment effect TE takes the values 0.5, 2 and 5. The datasets are specified in Table 6.5

Simulation	$\mu(x)$	$e(x)$	$\tau(x)$
1	$0.5 + 1 \cdot I_{(x_2<0)}$	$\begin{cases} 0.6 & \text{if } x_2 < 0, x_{12} = F \\ 0.4 & \text{else} \end{cases}$	$TE \cdot I_{(x_1>1)}$
2	$0.5 + 1 \cdot I_{(x_2<0)}$	$\begin{cases} 0.6 & \text{if } x_2 < 0, x_{12} = F \\ 0.4 & \text{else} \end{cases}$	$TE \cdot x_1$

Table 6.5: Simulations with varying treatment effect $TE = 0.5, 2, 5$

The plots in Figure 6.15 show an increasing RMSE with an increasing TE. This is caused by the increasing variance, printed in Figure A.27. Moreover, the interquartile range of the variance is large for a large value of the treatment effect and small n. The RMSEs are similar for weighted and unweighted trees. By contrast, the biases decrease with weighting. In simulation 1, the biases are not influenced by the value of the treatment effect. As before, they approach zero with an increasing n, even without weighting. In simulation 2, the biases increase slightly with an increasing value of the treatment effect. Additionally, weighting is more essential, especially for a large value of the treatment effect.

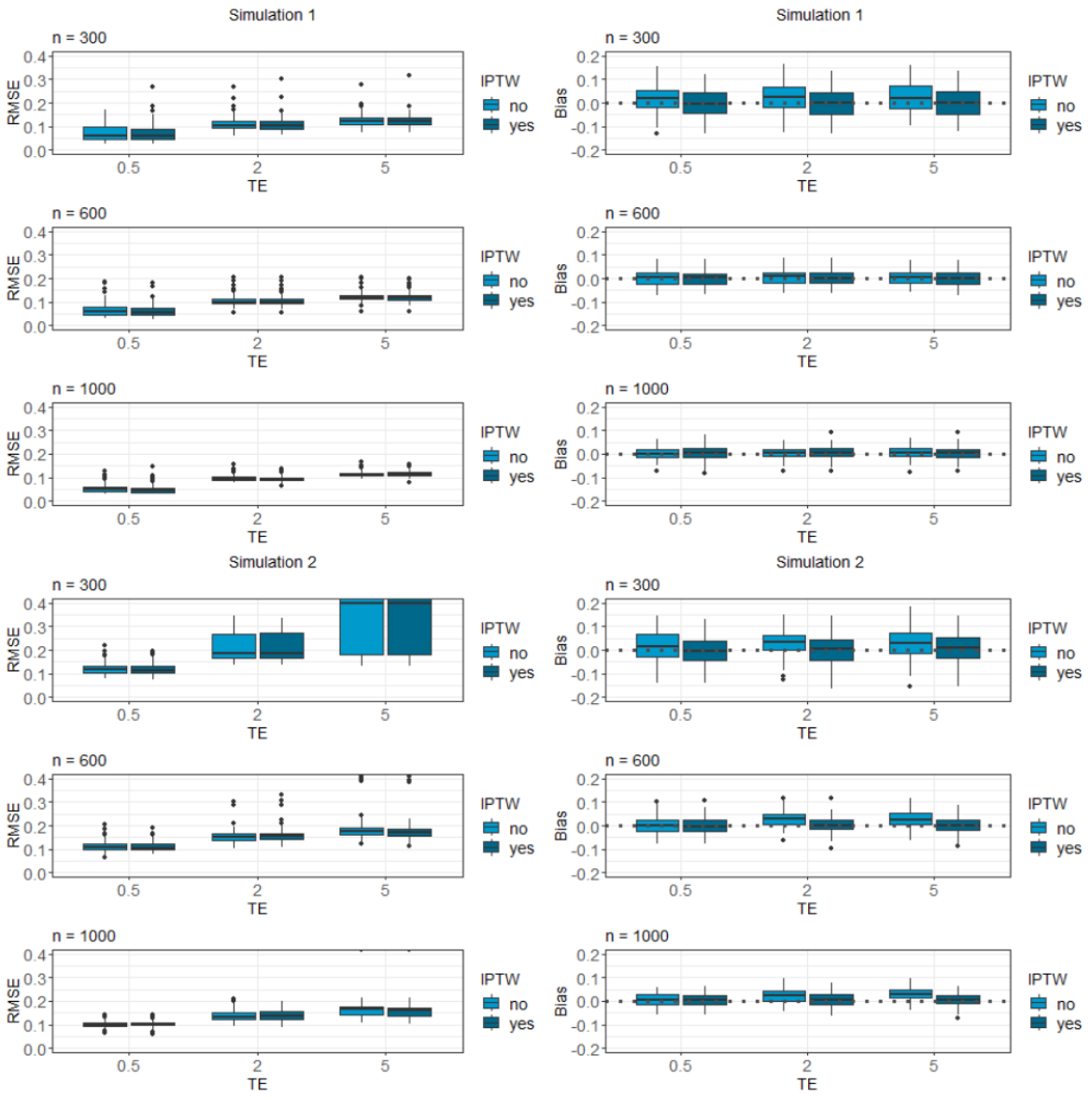

Figure 6.15: RMSE and bias of GLM trees for different treatment effects

The running time, shown in Figure 6.16, strongly increases with large values of TE. In particular for the weighted trees in simulation 2, the difference is remarkable. The larger n and the values of TE, the longer it takes the model to be fitted. Moreover, IPTW adjustment extends the running time. In simulation 1, the running time is almost similar between all scenarios, except for the weighted trees with $TE = 5$ and $n = 1000$, where the running time is higher.

The running times for $TE = 2$ and 5 are not completely visible in simulation 2. Thus, they are visualised with an extended y-axis in the appendix (see Figure A.28).

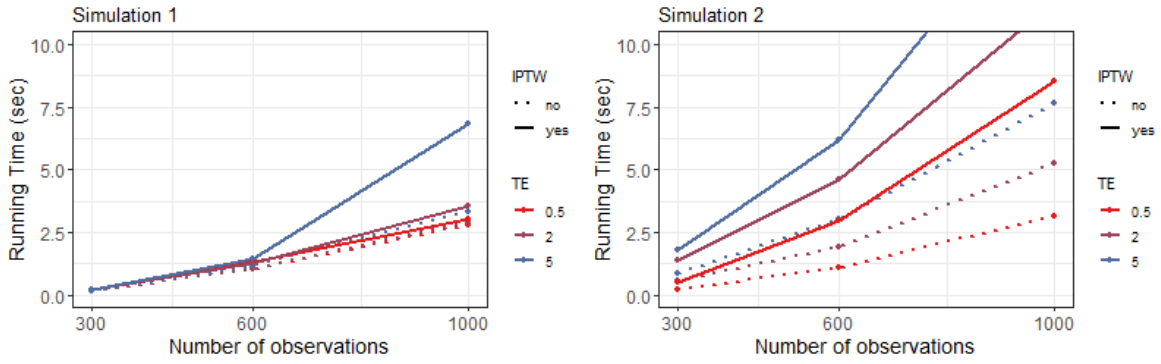

Figure 6.16: Running time of GLM trees for different treatment effects

7 Discussion and Outlook

There is a large interest in estimating personalised treatment effects in the case of non-randomised datasets. For this purpose, the performance of some tree-based methods and one regression spline method were tested by conducting a simulation study. Datasets differing regarding the number of confounders and the type of treatment effect functions were simulated. The methods showed different performances in distinct scenarios. The results of the simulation study are briefly summarised in the following.

7.1 Summary of Results

Overall, linear treatment effect functions led to higher variances in the estimates than stepwise treatment effect functions. Furthermore, with linear treatment effect functions most differences in the methods became apparent. The trees, i.e. the GLM tree and the causal tree, performed well in the cases without confounder. In addition, the GLM tree was stable in the case of confounding except for datasets with a linear treatment effect function. However, in these cases the bias could be reduced by IPTW and matching, whereby IPTW is slightly better than matching. Nevertheless, they still had a large variance leading to a high RMSE. The causal tree was very sensitive to confounding. Even the inclusion of weights did not substantially improve the fit. The main problem with causal trees was not the bias, but the increasing variance. The bias by contrast could be reduced with IPTW. As mentioned, the causal tree had an advantage by fixing the maximal depth beforehand. Nevertheless, it performed poorly regarding the RMSEs. The causal forest had the lowest variance of all methods for a large sample size. Its bias was reduced by the local centering approach. This was also slightly reflected in the RMSE. With an increasing number of observations, the interquartile range of the bias was getting smaller and approached zero. This also applied to all other methods. BART had the smallest variance for small n. The bias approached zero with an increasing number of observations. Additionally, the RMSE decreased with an increasing n. Thus, with large n, BART was a well performing method. The propensity score adjustment did not essentially improve the results. The PTO forest performed well in all scenarios. Overall, it had a relatively low variance. However, it was slightly biased, especially for linear treatment effect functions. In contrast to other methods, the propensity score adjustment slightly worsened the fit. Hence, the PTO forest should be applied

without adjustment. By contrast, the bias of the causal MARS model was reduced with propensity score stratification. Nevertheless, the variance of the estimates was relatively high for a linear treatment effect function. In the case of a stepwise treatment effect function, the variance occasionally was extremely high.

In these analyses, the causal tree, the causal forest and the GLM trees had a slight advantage compared to the other methods. For causal trees, the maximal depth of the trees was defined beforehand. The causal forest was the only method that was tuned and the GLM trees were manually pruned after the tree building phase. The reason for tuning the GLM trees was to ensure that the models fitted in each leaf are reasonable and interpretable. In `R`, errors occurred in the case of insufficient numbers of observations in each leaf. These are counted for each simulation in the main analysis and further analyses parts and included in the electronic appendix.

The results of this thesis are comparable to the results of Wendling *et al.* (2018). In their paper, BART, causal forests and causal MARS are compared. Additionally, they included causal boosting and regularised logistic regression (LASSO + Ridge) in their analysis. The BART model, as well as causal boosting performed best in their evaluation. Causal forest and causal MARS were competitive, except for a low variance of the treatment effect. Causal boosting however, was computationally much more demanding than BART. Especially for BART and causal boosting, the propensity score adjustment did not essentially improve the fit.

Obviously, no method is preferable for all situations. The recommended method depends on the dataset. In simulation 1, with no confounding and homogeneous treatment effects, the recommended methods are either the GLM tree or the causal forest. These two methods have the lowest RMSE. In simulation 2, with heterogeneous treatment effects and a stepwise treatment effect function, the causal tree is a reasonable option. In the case of a linear treatment effect function (simulation 3), all methods perform similarly. An exception is the causal tree, resulting in a large RMSE. Looking at simulation 4, the first simulation with confounding and no heterogeneous treatment effects, the IPTW adjusted causal tree provides the lowest RMSE for large n. For a small number of observations, all other methods perform better, except for causal MARS. The method with the lowest RMSE is the PTO forest without adjustment. For simulations 5 and 7 (stepwise treatment effect function and confounding), all methods perform similarly. One exception is the causal tree, which has large variances and consequently large

RMSE values. BART is slightly superior to the other methods. It has the lowest RMSE for large n and it performs well without propensity score adjustment. For a linear treatment effect function and confounding (simulations 6 and 8) and small n, causal MARS has the lowest RMSE. However, it has a long running time. For a large sample size, BART performs similarly and no propensity score adjustment is necessary. Apart from this, it is required to apply all methods with an adjustment for confounding, except for the PTO forest. As described in Section 6.6, weights in GLM trees always reduce the bias. Thus, weights never worsen the fit. In the worst case, they are only unnecessary since the bias is already close to zero without any weights. For extreme propensity scores, weights are reasonable to include. This applies to large coefficients in the mean effect function as well. However, the running time for weighted GLM trees is extremely high for large n, especially for a linear treatment effect function. Additionally, a large value of the treatment effect and IPTW adjustment increases the running time.

7.2 Outlook

In this thesis, not all potential data structures could be covered. In the main analysis part, the focus was on the number of confounders, the type of the treatment effect functions and the number of variables in the treatment and mean effect functions. All datasets had the same coefficients, number of covariates and distribution of covariates. Additionally, the treatment effect was fixed. The coefficients of the mean effect functions, different propensity scores and a varying value of the treatment effect were tested for GLM trees. In a further study, this could be applied to other methods.

Additionally, only confounders were considered in the present thesis. But there might be more complex data structures in real life, such as the presence of a collider or a mediator. A collider is a covariate that is influenced by the treatment as well as by the outcome variable. A mediator is a variable that is directly connecting the treatment and the outcome variable: the treatment influences the mediator and the mediator the outcome. Furthermore, in this simulation study, the covariates are all independent. A further investigation of the influence of dependency structures between the covariates on the treatment effect estimation could be reasonable.

Moreover, most of the hyperparameters used in this thesis were set to their default values (except those of the causal forest). In further analysis, these parameters should

be tuned.

For the matching procedure, only the nearest neighbour method is used. It might be of interest whether other matching methods improve the fit. However, matching was criticised by some researchers: problems can arise with matching, that need to be solved. Imbalance, inefficiency, model dependence, and bias can increase (King & Nielsen 2018).

Furthermore, with an adjustment for confounding the results in most methods showed a reduction of the bias and at the same time a slightly increased variance. In the IPTW adjustment one reason for the high variances might be the large weights. To circumvent this problem, weights might be trimmed to the 90th percentile. That means all weights with values above the 90th percentile of the weights are set to the 90th percentile. This approach can be tested in a further study.

There are also other promising approaches that target confounding in the dataset, e.g. the targeted maximum likelihood estimation (TMLE), which is a doubly robust method. But also other methods mentioned in Chapter 2 are thinkable, such as the synthetic forests developed by Ishwaran & Malley (2014). Lu *et al.* (2018) compare different random forest methods to estimate personalised treatment effects. They show that synthetic random forests generally perform best among all methods. They even outperform the BART model. Thus, this is a promising approach which can be further investigated in continued work.

A Appendix

A.1 Parameter Tuning of Causal Forest

Simulation	Parameter	Min.	Median	Mean	Max.
1	Minimal Node Size	1.00	4.00	11.74	62.00
	mtry	1.00	7.00	8.61	21.00
	alpha	0.00	0.14	0.13	0.25
	Imbalance Penalty	0.00	0.66	0.85	5.18
2	Minimal Node Size	1.00	3.00	3.57	18.00
	mtry	1.00	17.00	15.17	21.00
	alpha	0.00	0.07	0.08	0.23
	Imbalance Penalty	0.01	0.70	0.86	6.53
3	Minimal Node Size	1.00	2.00	1.97	7.00
	mtry	14.00	18.00	17.95	21.00
	alpha	0.01	0.13	0.12	0.24
	Imbalance Penalty	0.11	0.67	0.71	2.20
4	Minimal Node Size	1.00	3.00	5.01	36.00
	mtry	1.00	18.00	16.85	21.00
	alpha	0.00	0.14	0.13	0.25
	Imbalance Penalty	0.02	0.67	0.81	4.75
5	Minimal Node Size	1.00	3.00	3.42	19.00
	mtry	1.00	18.00	16.39	21.00
	alpha	0.00	0.11	0.11	0.25
	Imbalance Penalty	0.00	0.75	0.90	3.31
6	Minimal Node Size	1.00	2.00	1.95	12.00
	mtry	14.00	18.00	18.06	21.00
	alpha	0.01	0.12	0.12	0.23
	Imbalance Penalty	0.04	0.68	0.73	2.29
7	Minimal Node Size	1.00	2.00	3.49	36.00
	mtry	1.00	18.00	16.92	21.00
	alpha	0.01	0.13	0.13	0.25
	Imbalance Penalty	0.04	0.70	0.84	3.37
8	Minimal Node Size	1.00	2.00	2.02	6.00
	mtry	15.00	18.00	18.37	21.00
	alpha	0.02	0.12	0.12	0.23
	Imbalance Penalty	0.06	0.69	0.75	2.56

Table A.1: Summary of results of hyperparameter tuning of causal forest

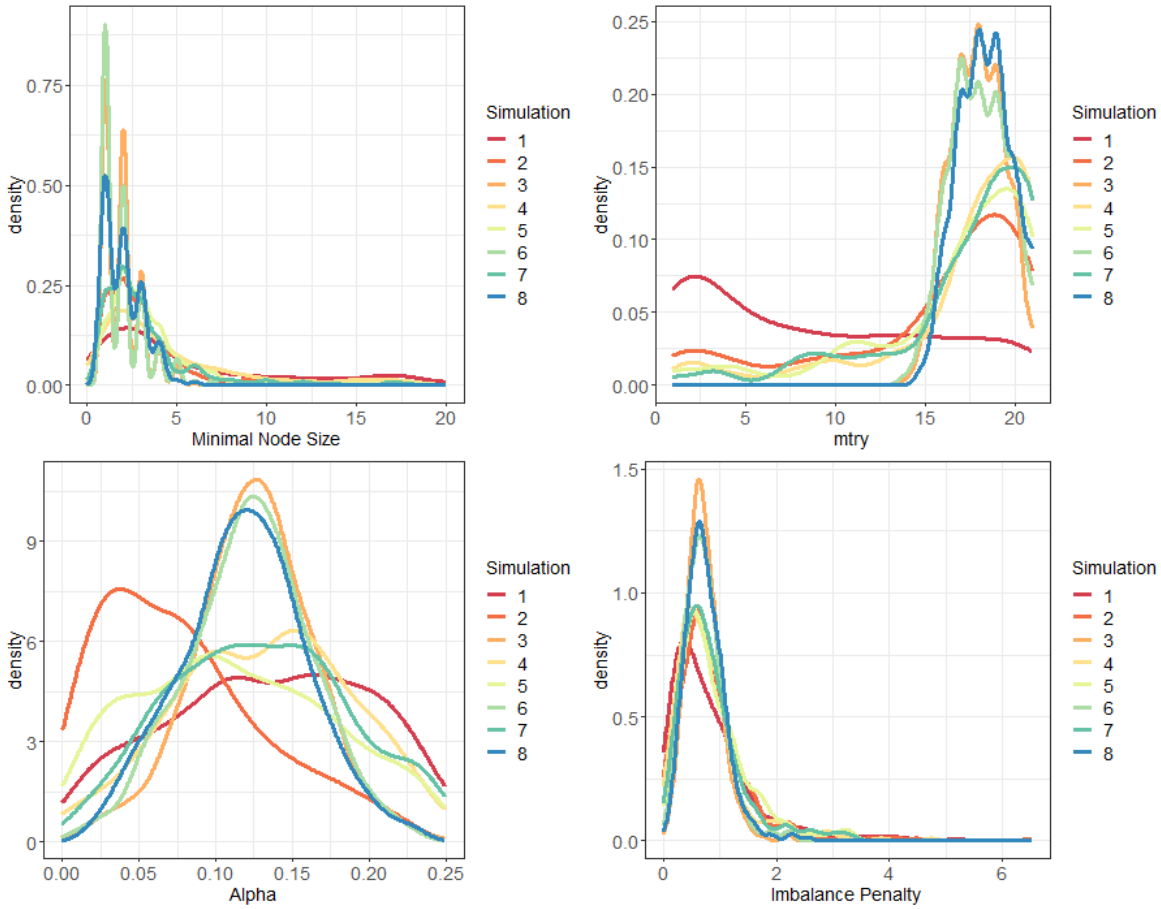

Figure A.1: Results of hyperparameter tuning of causal forest

A.2 Variance of Simulations

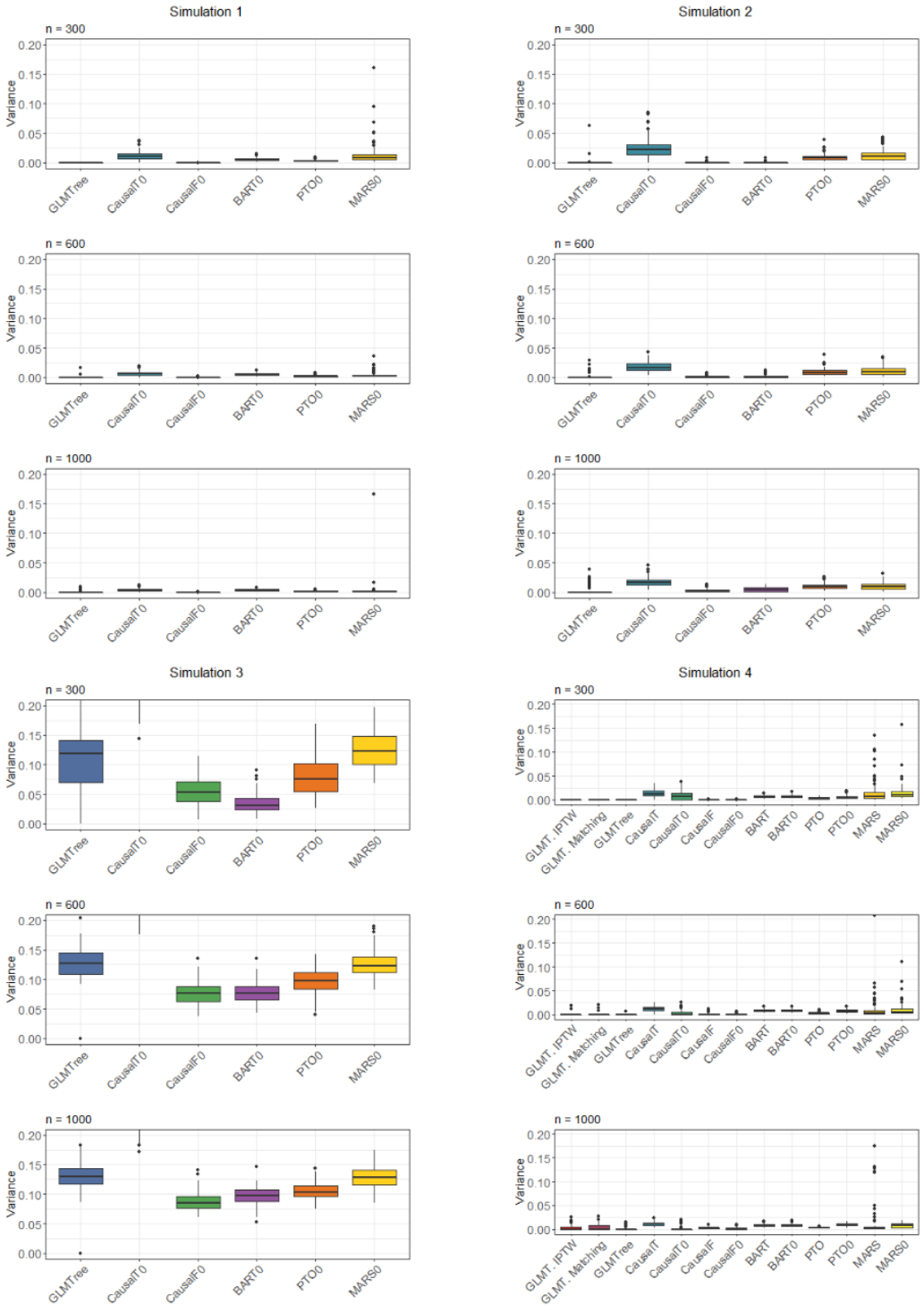

Figure A.2: Variance of different methods for simulations 1 - 4

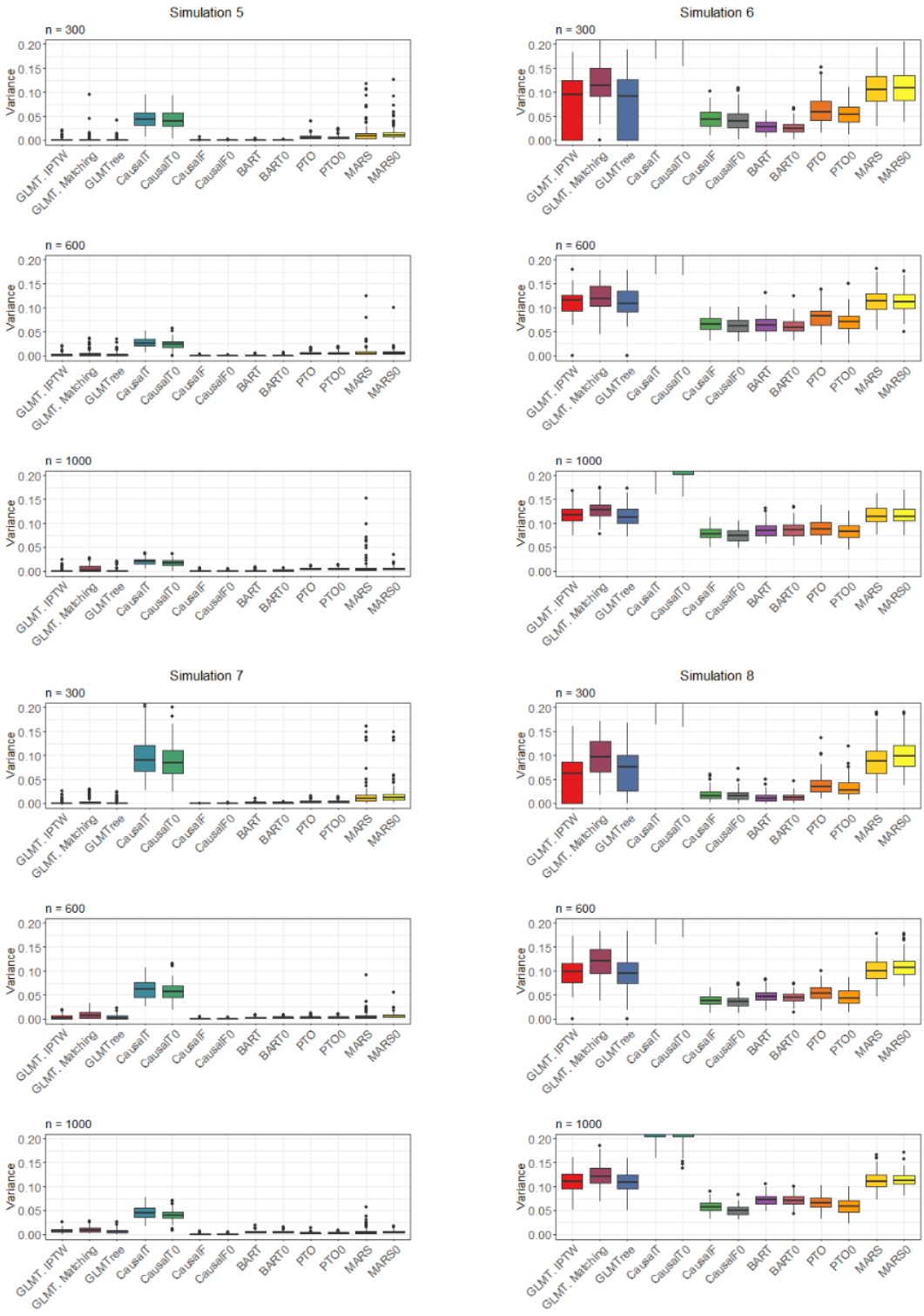

Figure A.3: Variance of different methods for simulations 5 - 8

A.3 Complete Plots of RMSE and Variance

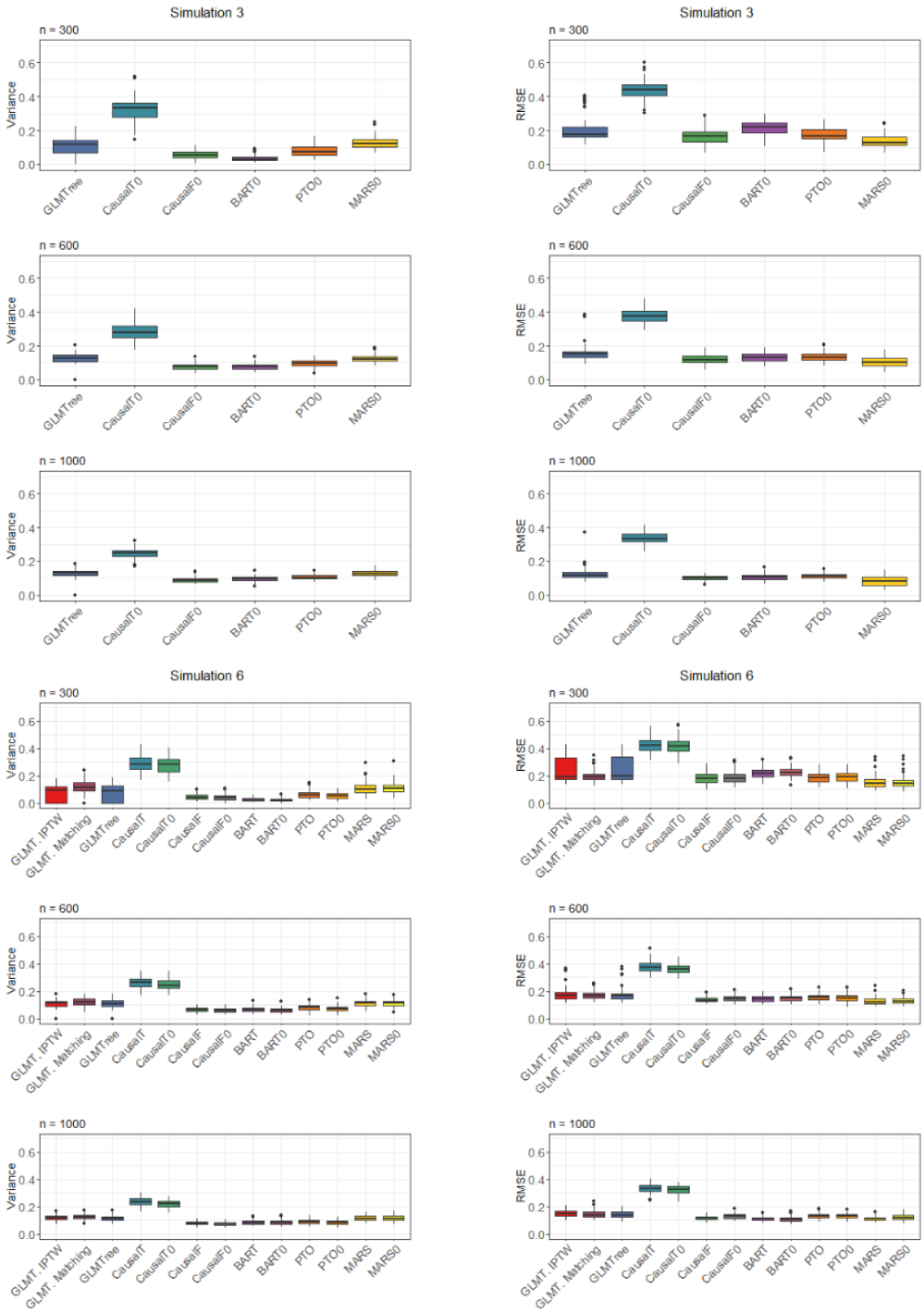

Figure A.4: RMSE and variance of different methods with extended y-axis for simulations 3 and 6

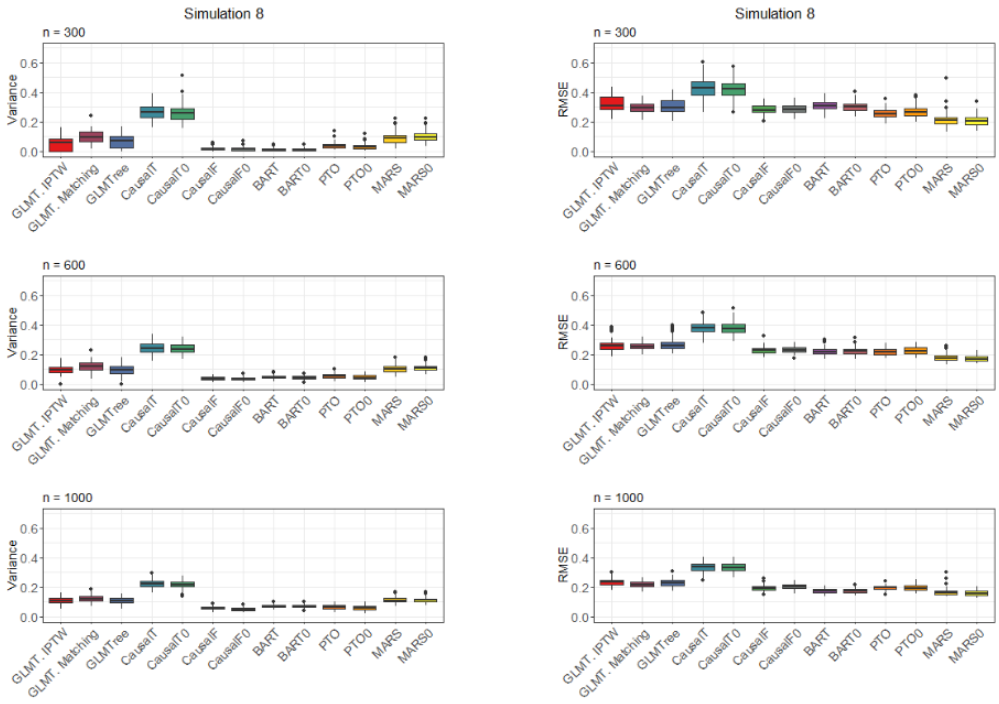

Figure A.5: RMSE and variance of different methods with extended y-axis for simulation 8

A.4 Estimated Treatment Effects

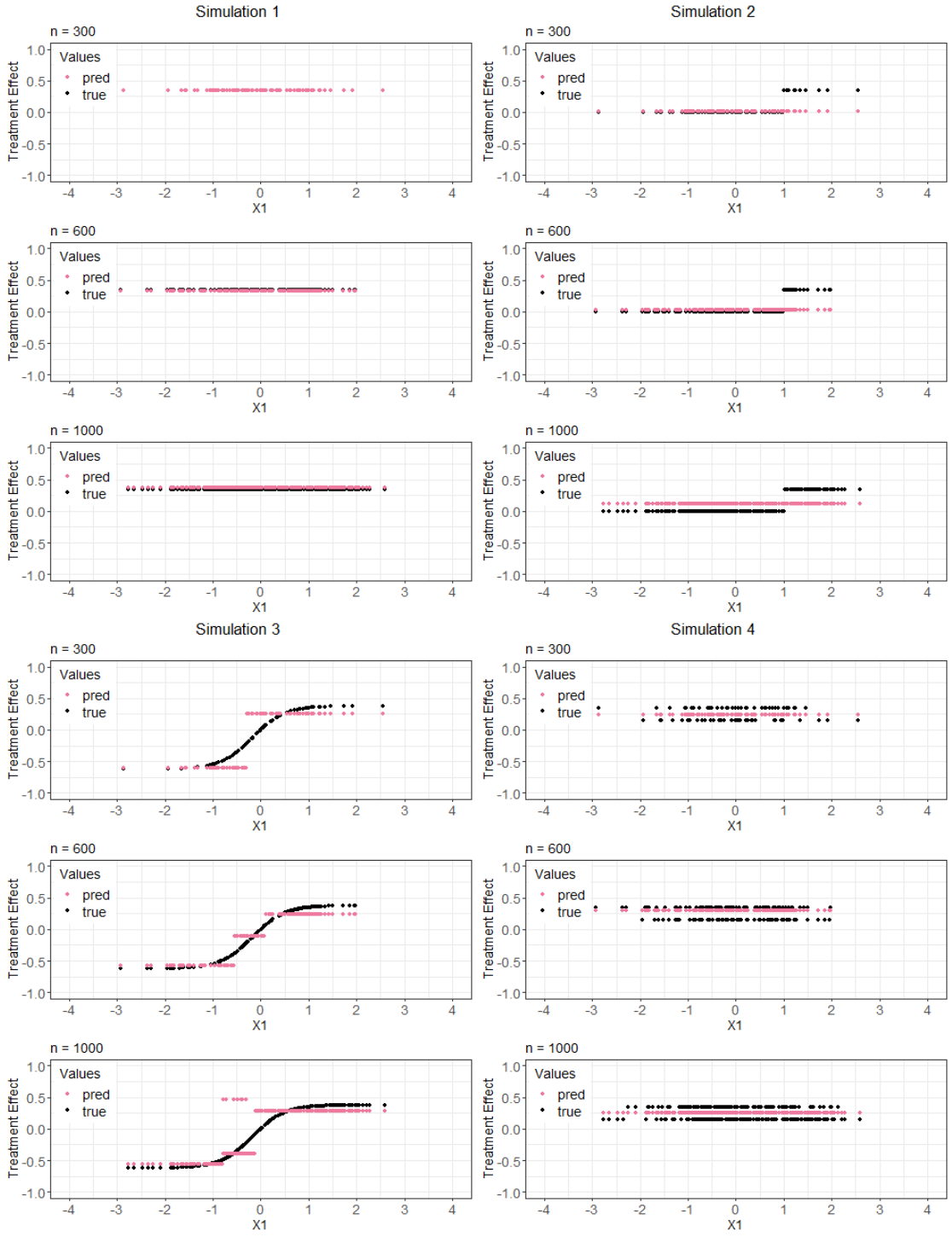

Figure A.6: Prediction of treatment effect function for GLM tree with IPTW for simulations 1-4

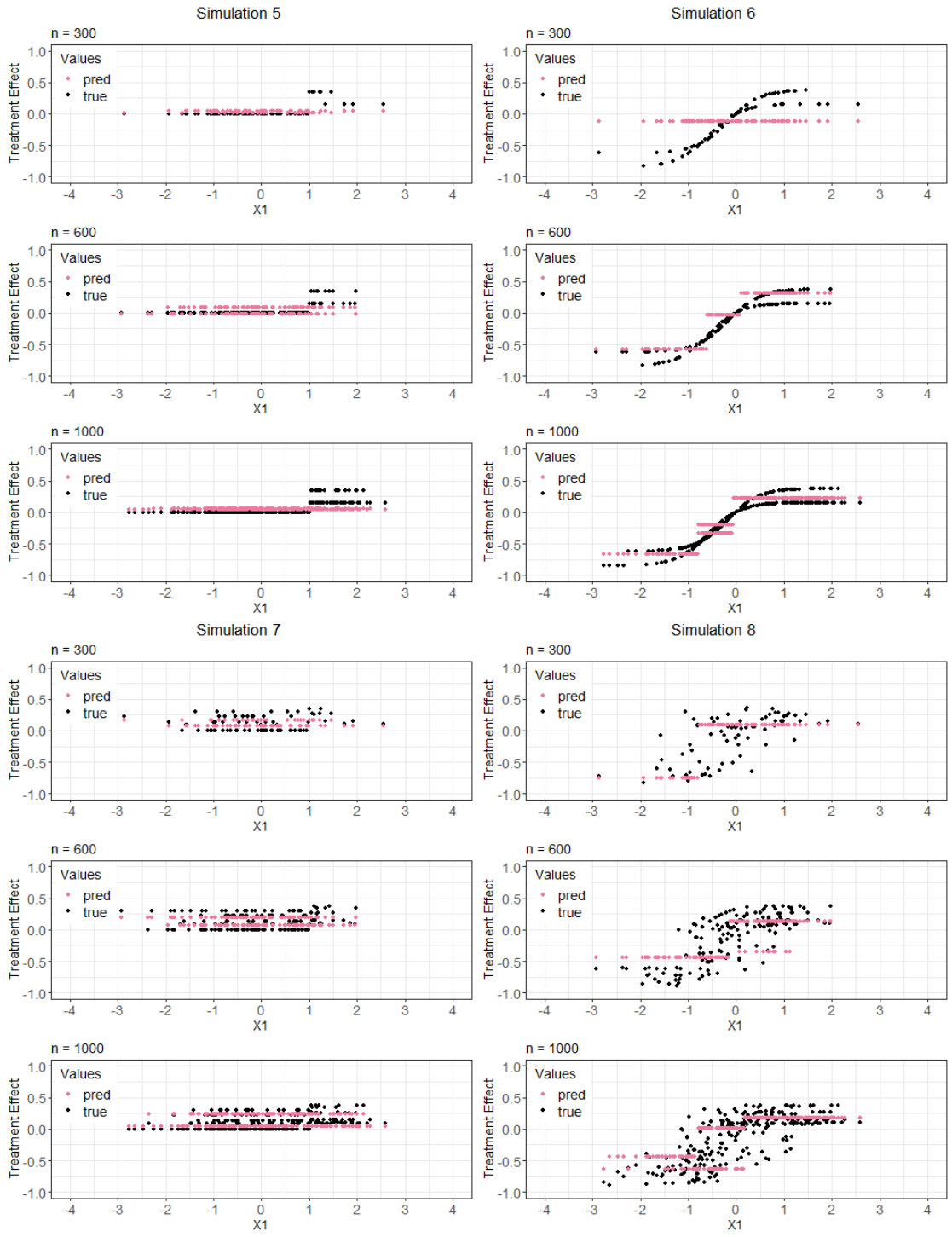

Figure A.7: Prediction of treatment effect function for GLM tree with IPTW for simulations 5-8

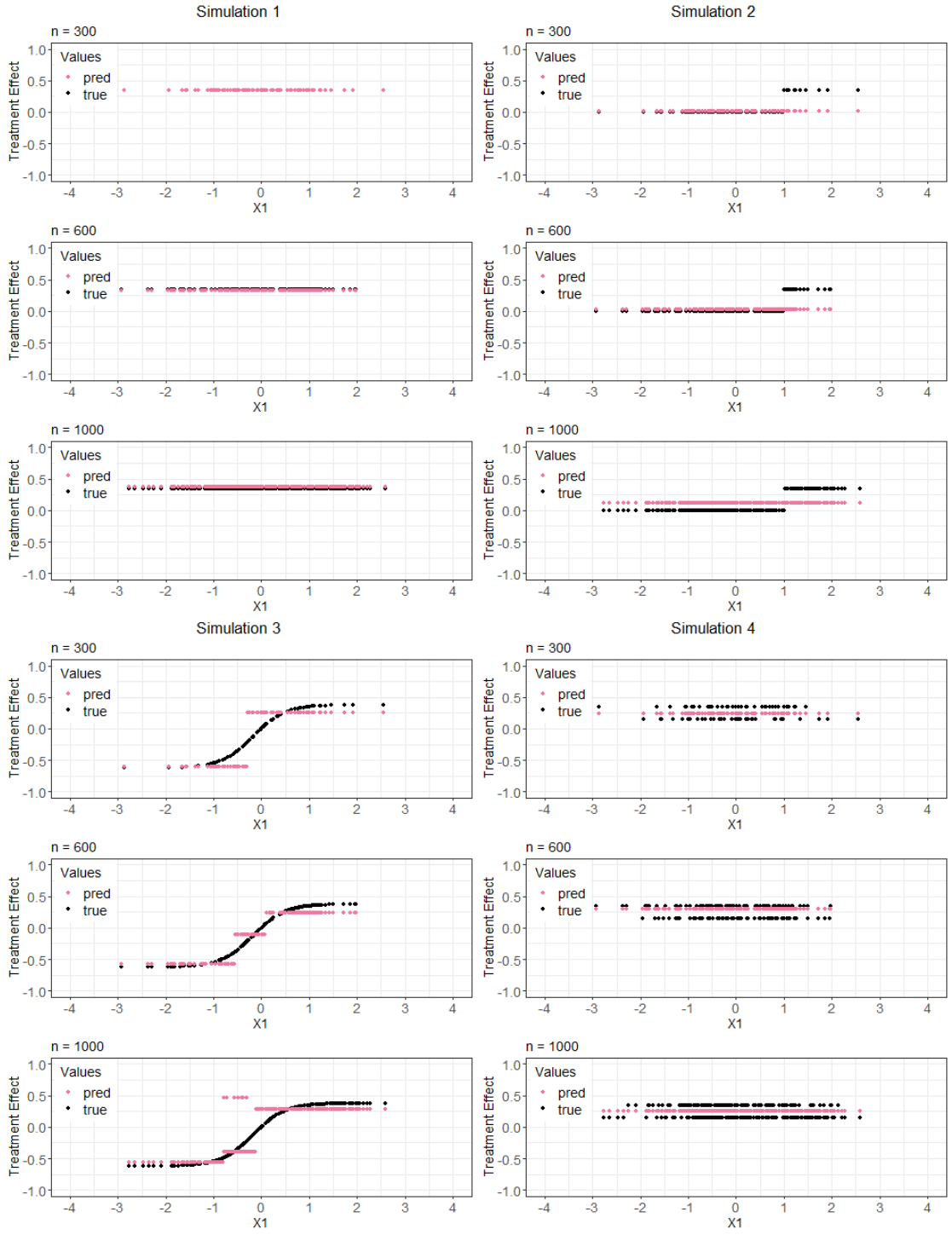

Figure A.8: Prediction of treatment effect function for GLM tree with matching for simulations 1-4

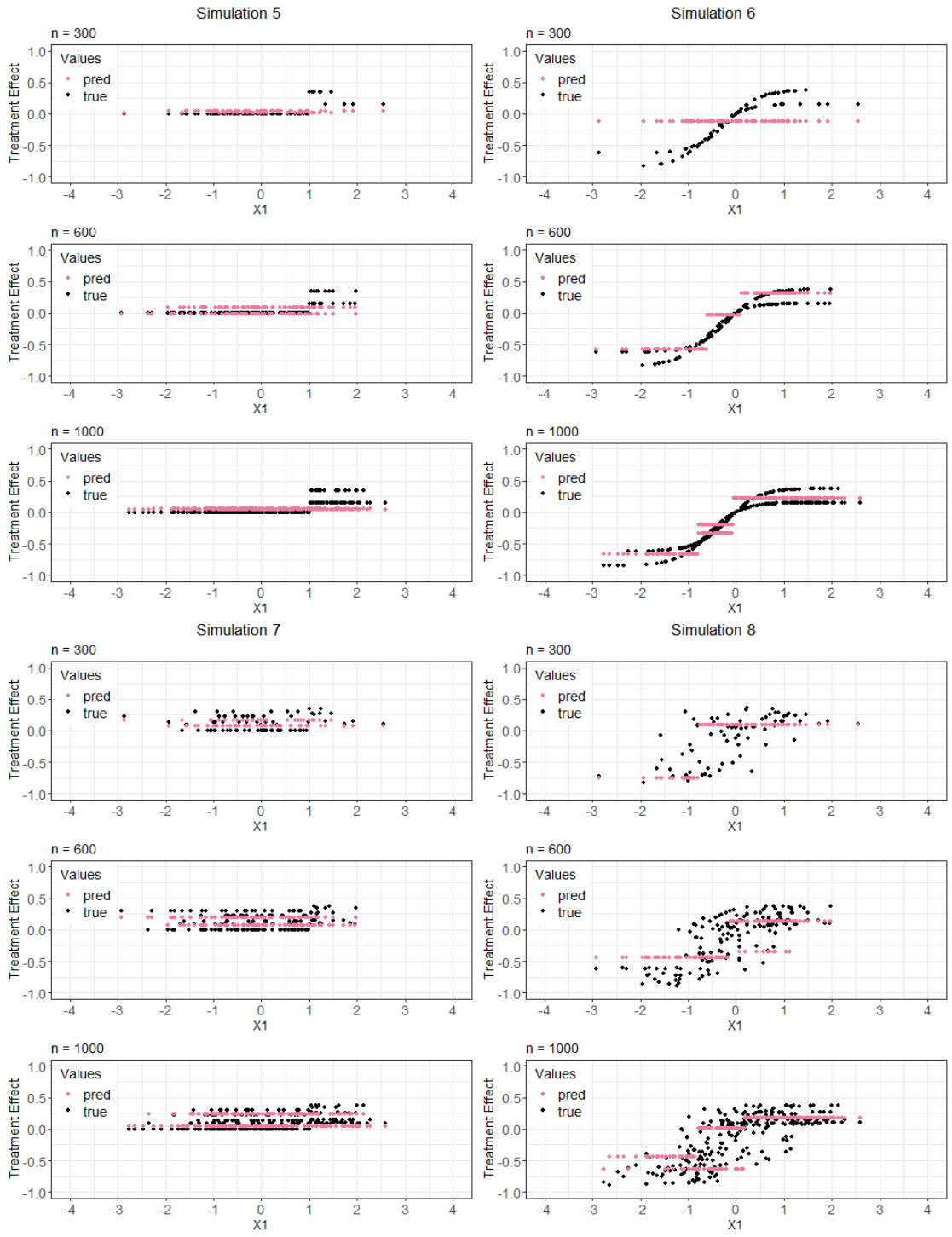

Figure A.9: Prediction of treatment effect function for GLM tree with matching for simulations 5-8

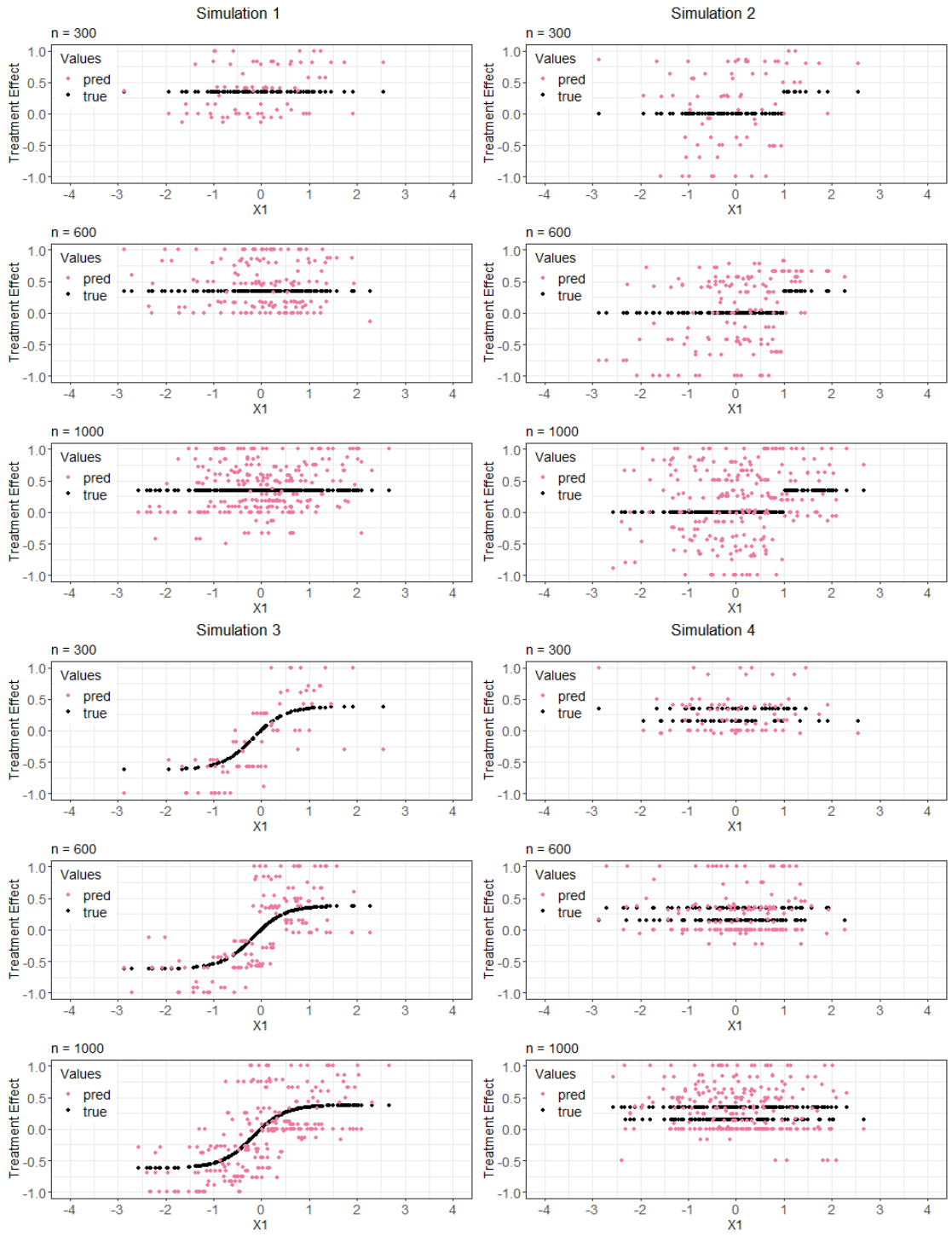

Figure A.10: Prediction of treatment effect function for causal tree for simulations 1-4

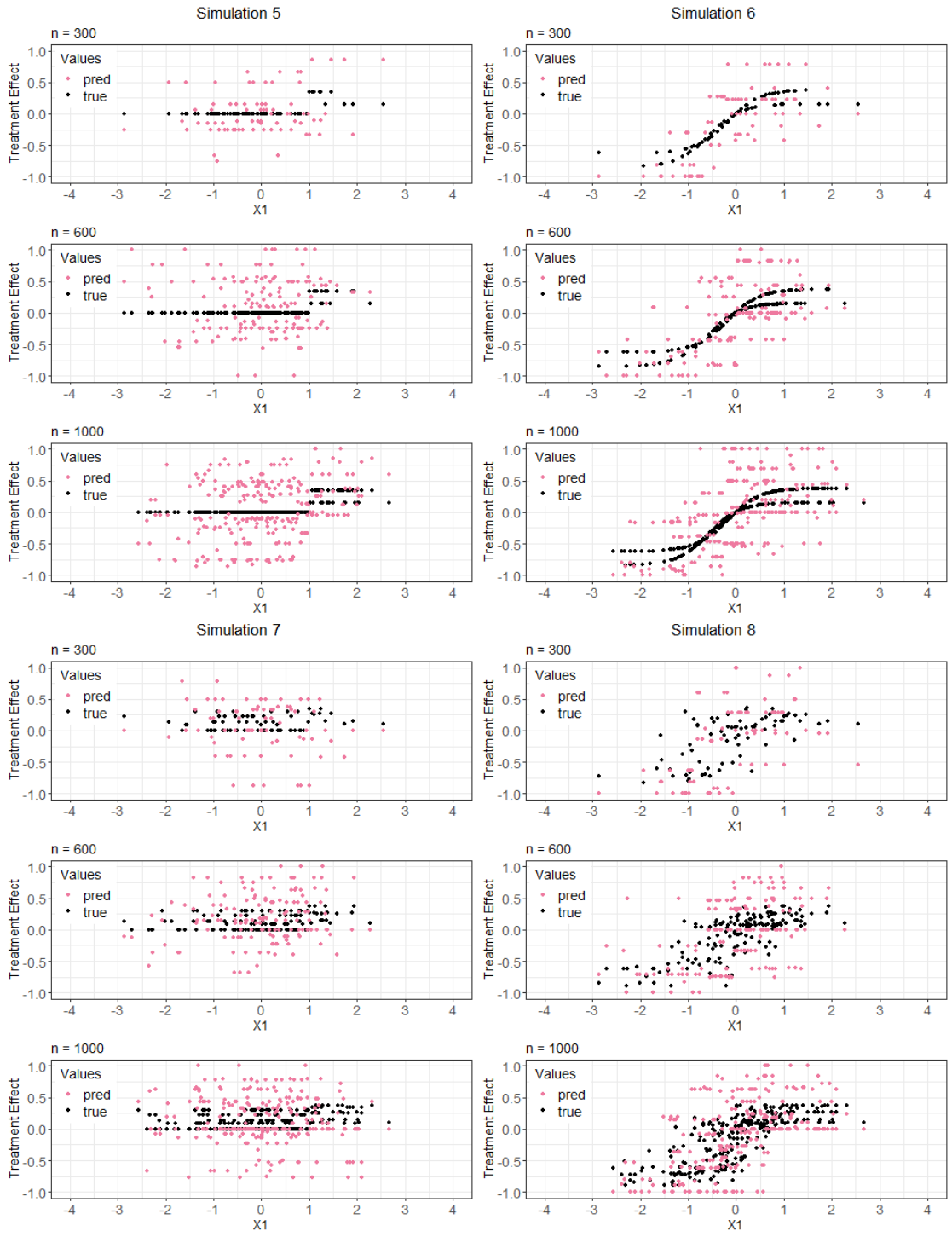

Figure A.11: Prediction of treatment effect function for causal tree for simulations 5-8

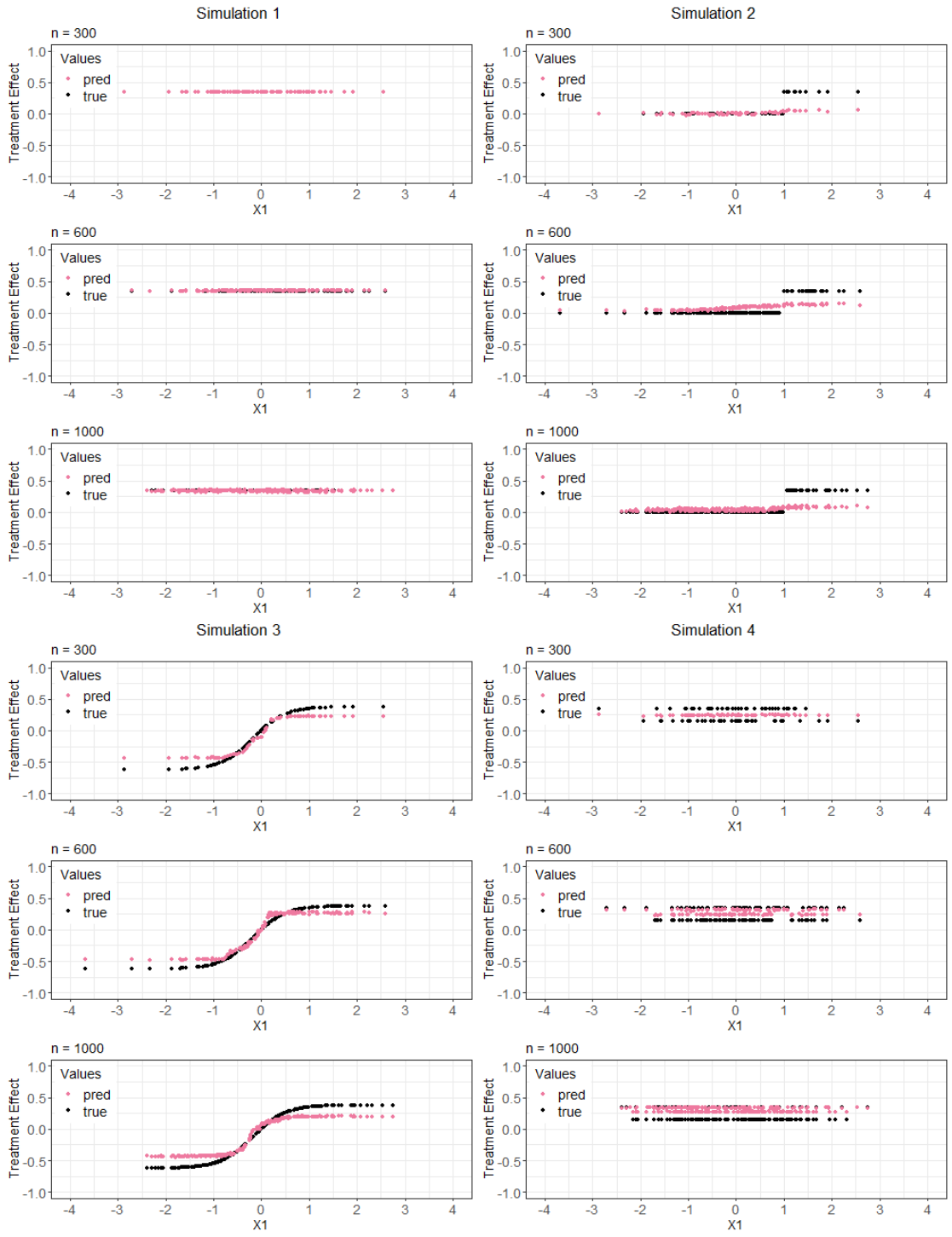

Figure A.12: Prediction of treatment effect function for causal forest for simulations 1-4

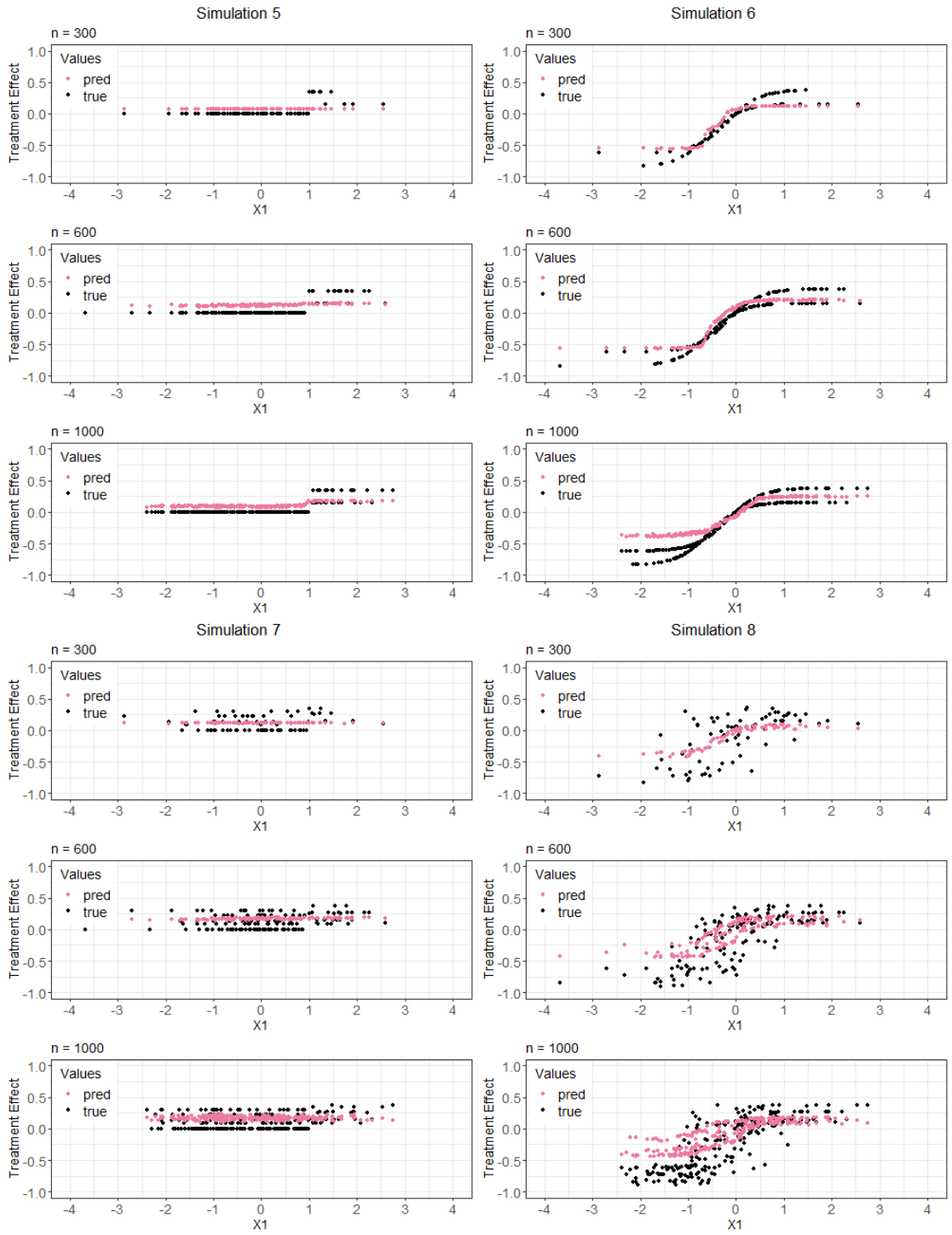

Figure A.13: Prediction of treatment effect function for causal forest for simulations 5-8

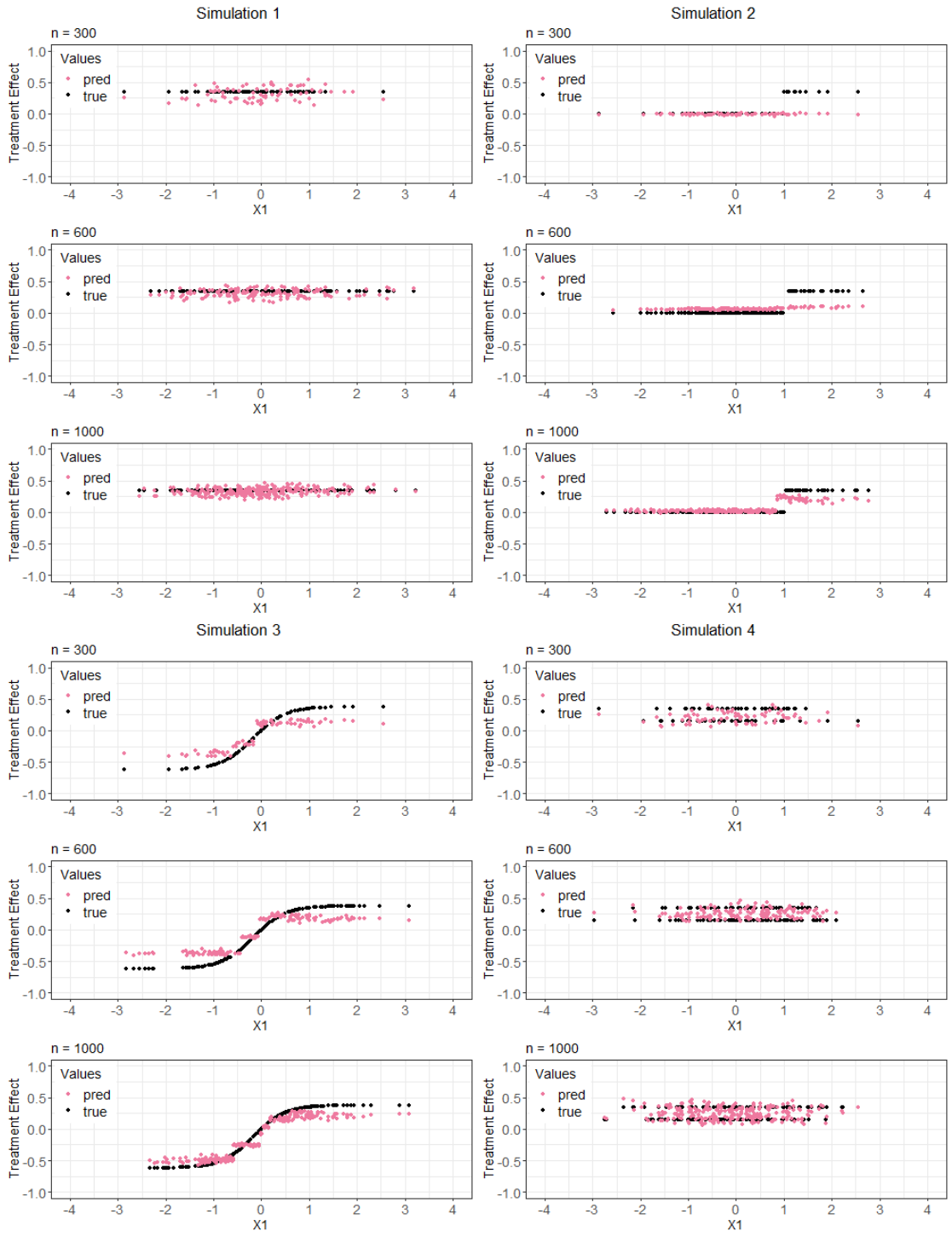

Figure A.14: Prediction of treatment effect function for BART for simulations 1-4

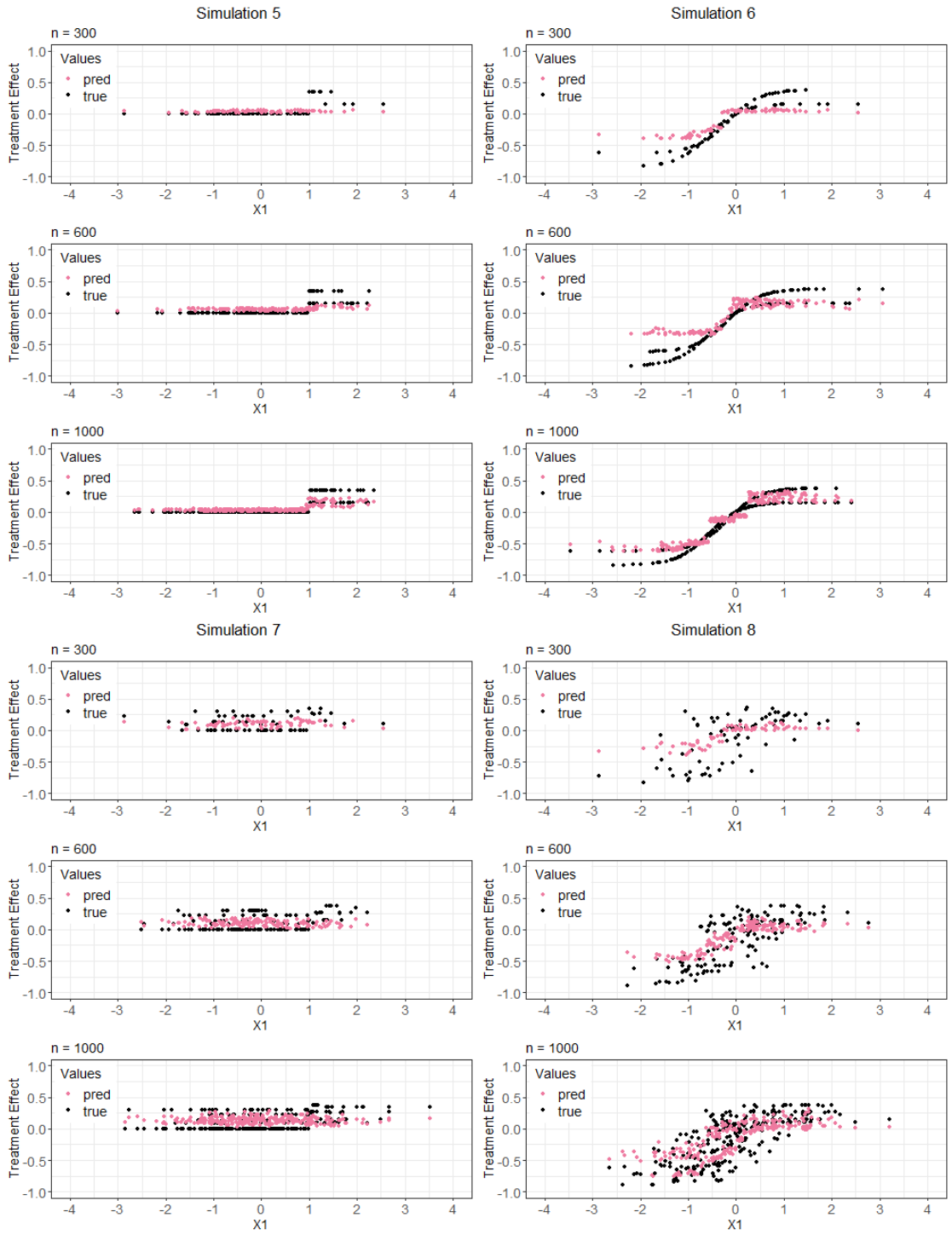

Figure A.15: Prediction of treatment effect function for BART for simulations 5-8

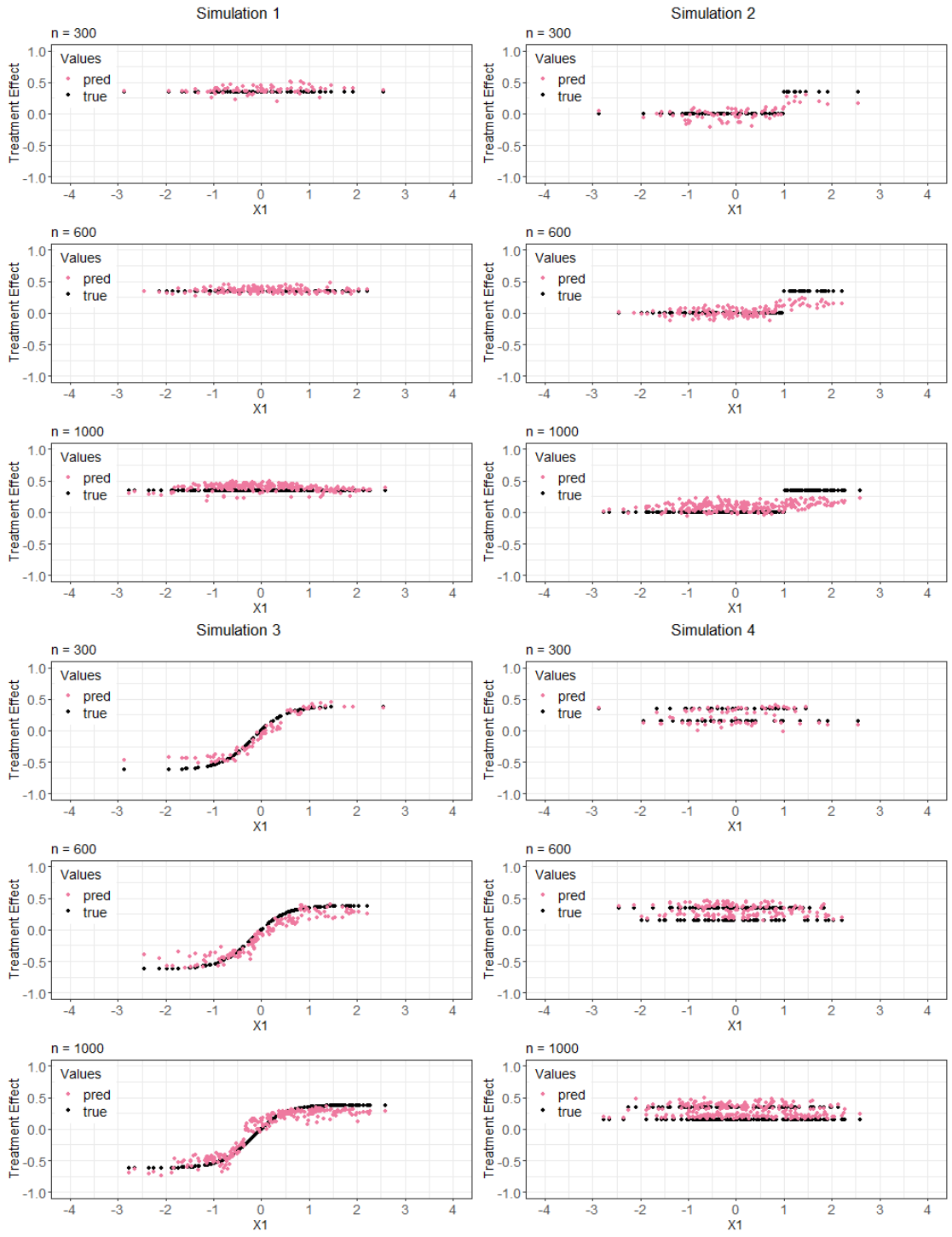

Figure A.16: Prediction of treatment effect function for PTO forest for simulations 1-4

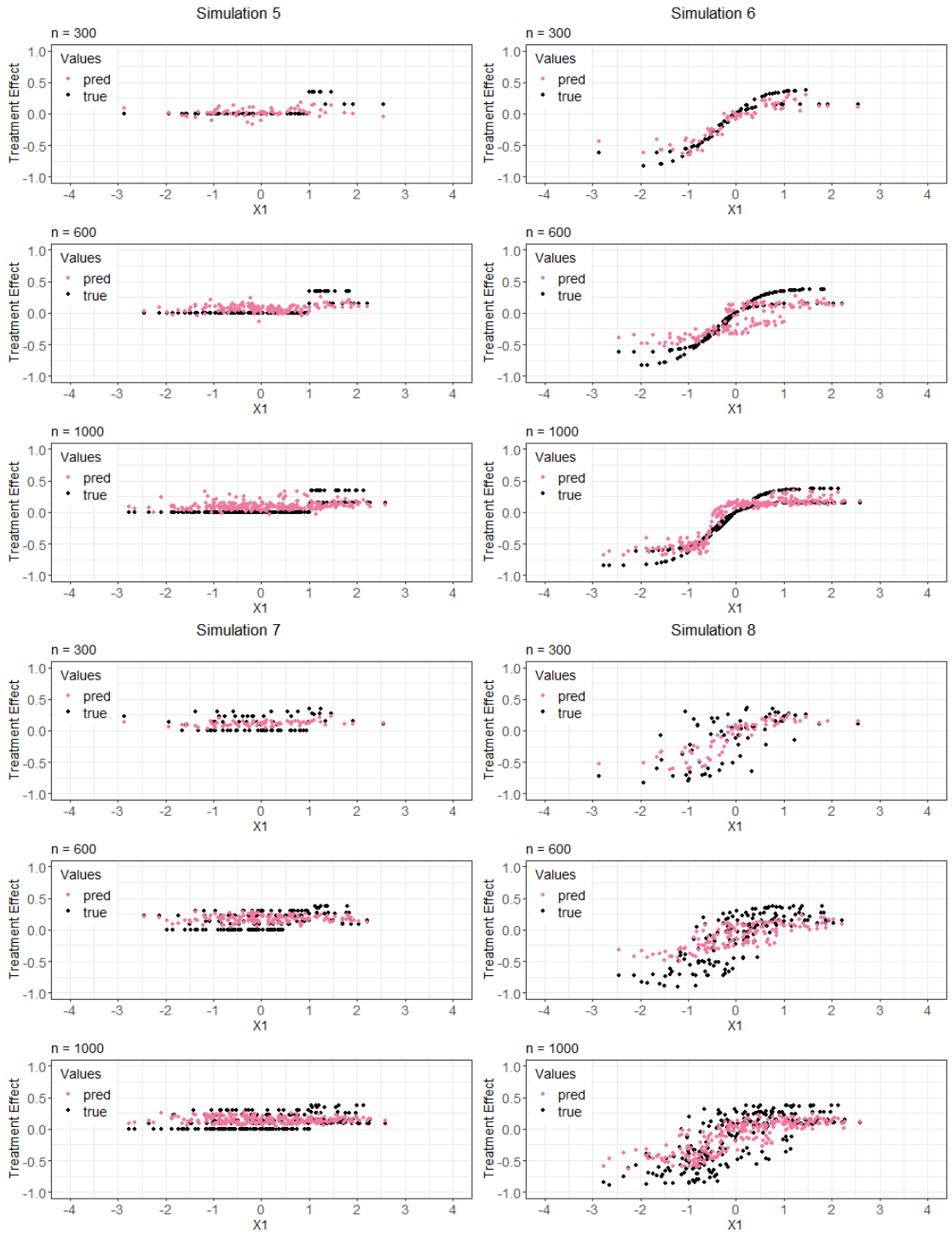

Figure A.17: Prediction of treatment effect function for PTO forest for simulations 5-8

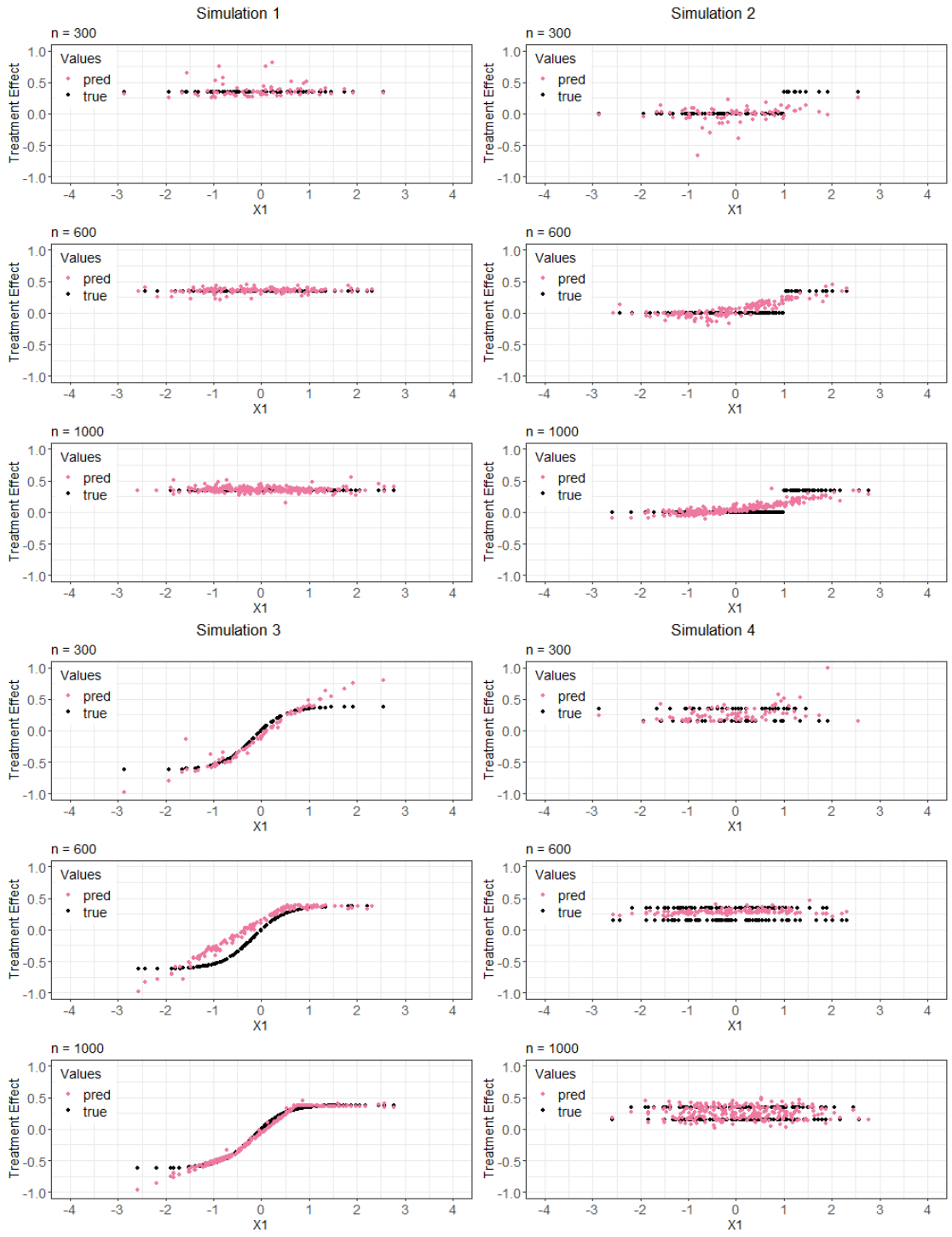

Figure A.18: Prediction of treatment effect function for causal MARS for simulations 1-4

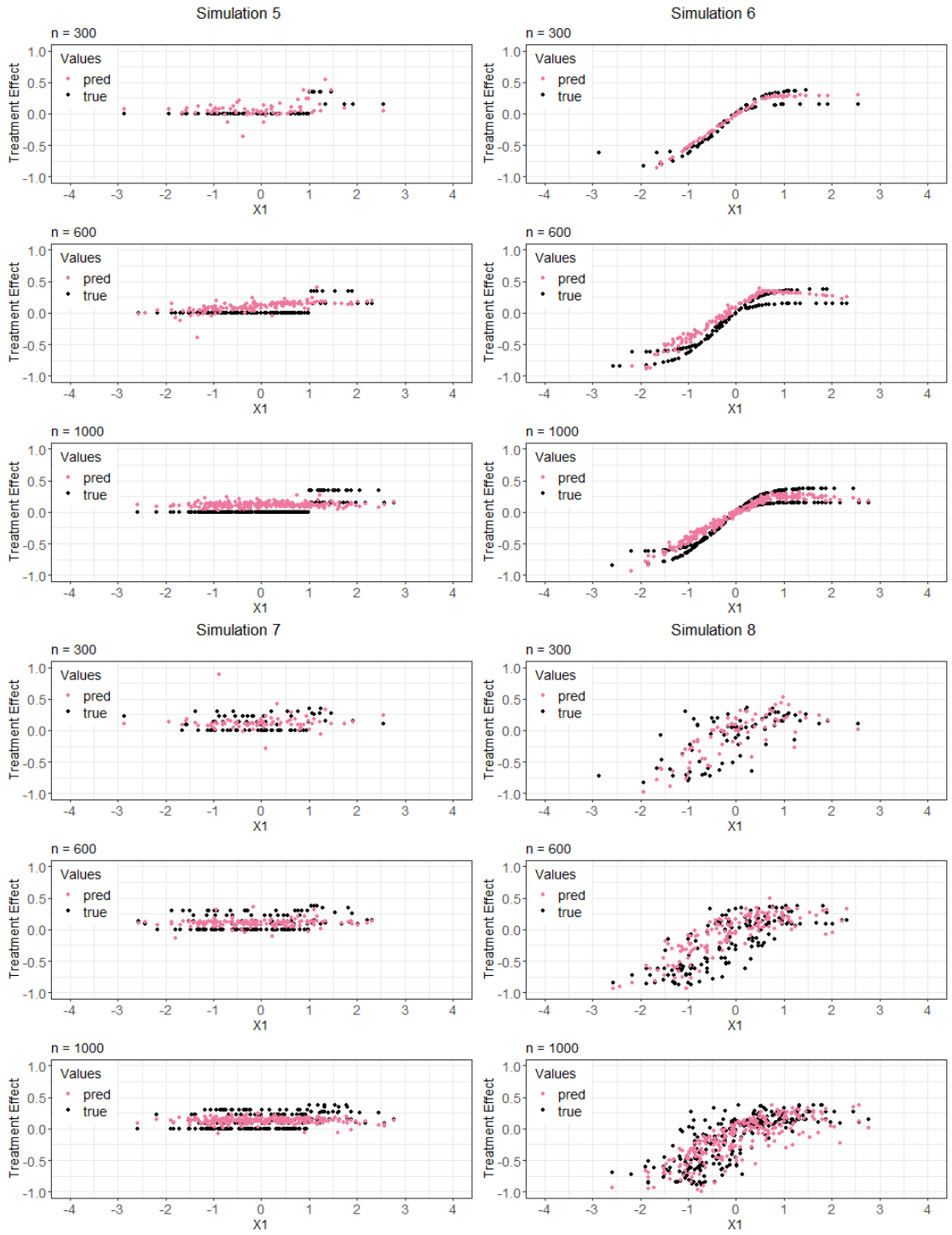

Figure A.19: Prediction of treatment effect function for causal MARS for simulations 5-8

A.5 Running Time

n = 300

Method	Sim 1	Sim 2	Sim 3	Sim 4	Sim 5	Sim 6	Sim 7	Sim 8
GLMTree IPTW				0.70	0.45	1.31	0.52	1.69
GLMTree Matching				**0.34**	0.40	0.58	**0.18**	0.49
GLMTree	0.26	0.50	1.98	1.58	0.71	1.58	0.58	1.94
CausalT				**0.34**	**0.29**	**0.32**	0.31	**0.32**
CausalT0	0.30	0.31	**0.35**	0.34	0.32	0.33	0.31	0.34
CausalF				2.15	3.34	3.58	3.44	2.01
CausalF0	**0.23**	**0.28**	0.36	**0.23**	**0.25**	**0.25**	**0.17**	**0.24**
BART				2.95	3.60	3.99	3.88	3.29
BART0	2.32	3.56	3.55	2.96	3.24	3.00	2.96	2.42
PTO				1.42	1.42	1.52	1.51	1.63
PTO0	1.34	1.50	1.36	1.42	1.44	1.52	1.50	1.40
MARS				20.22	18.94	18.81	15.53	16.75
MARS0	14.65	8.71	8.88	9.34	9.50	9.41	9.15	7.68

n = 600

Method	Sim 1	Sim 2	Sim 3	Sim 4	Sim 5	Sim 6	Sim 7	Sim 8
GLMTree IPTW				3.04	2.40	4.48	1.90	5.34
GLMTree Matching				1.95	2.43	2.11	0.94	2.00
GLMTree	**0.26**	4.15	4.69	5.05	3.09	5.89	2.44	6.88
CausalT				**0.38**	**0.34**	**0.53**	**0.44**	**0.51**
CausalT0	0.36	**0.38**	**0.48**	**0.37**	**0.39**	**0.48**	**0.43**	**0.56**
CausalF				6.13	6.08	5.99	5.19	3.56
CausalF0	0.50	0.85	0.82	0.55	0.64	0.59	0.49	0.61
BART				4.69	5.00	6.33	5.56	6.28
BART0	4.10	4.42	4.67	4.44	4.13	4.68	4.49	4.37
PTO				2.65	2.76	2.85	2.85	2.71
PTO0	3.84	3.02	2.54	2.71	2.78	2.80	2.78	2.66
MARS				26.39	34.57	24.44	26.87	29.14
MARS0	19.56	11.86	12.01	12.41	12.62	12.48	10.62	10.32

n = 1000								
Method	Sim 1	Sim 2	Sim 3	Sim 4	Sim 5	Sim 6	Sim 7	Sim 8
GLMTree IPTW				8.38	6.63	12.41	6.33	18.83
GLMTree Matching				15.51	3.36	6.14	2.55	5.49
GLMTree	**0.44**	9.91	8.15	11.51	8.52	14.49	7.14	15.56
CausalT				**0.53**	**0.50**	**0.83**	**0.65**	**0.88**
CausalT0	0.45	**0.51**	**0.82**	**0.45**	**0.51**	**0.77**	**0.55**	**0.78**
CausalF				9.12	9.63	9.48	5.51	5.68
CausalF0	0.93	1.82	1.62	1.07	1.47	1.11	1.04	1.08
BART				8.68	9.15	8.31	6.40	7.63
BART0	7.99	6.57	7.98	7.11	6.01	9.68	6.99	7.88
PTO				4.61	4.69	4.94	4.92	4.57
PTO0	4.83	5.14	4.69	4.63	4.89	4.84	4.47	4.39
MARS				34.76	37.34	24.21	26.67	33.11
MARS0	20.59	15.22	25.26	16.44	16.76	16.72	14.16	14.71

Table A.2: Running time of methods in seconds with $n = 300$, 600 and 1000. Bold numbers indicate the best running time for each simulation and number of observations. For simulations 4-8 the best running time for methods with and without adjustment are highlighted, respectively. The running times without adjustment for confounding are printed in grey for simulations 4-8.

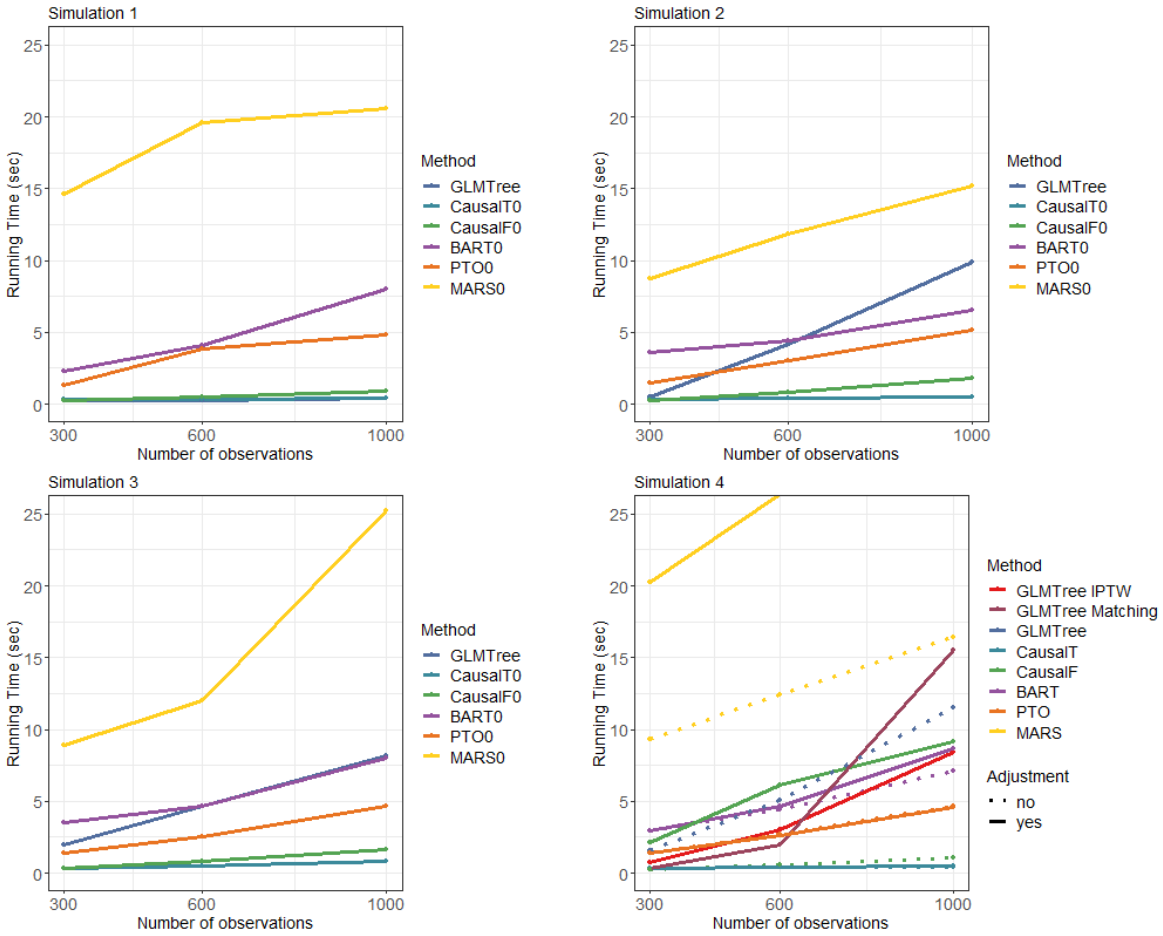

Figure A.20: Running time of methods for simulations 1-4

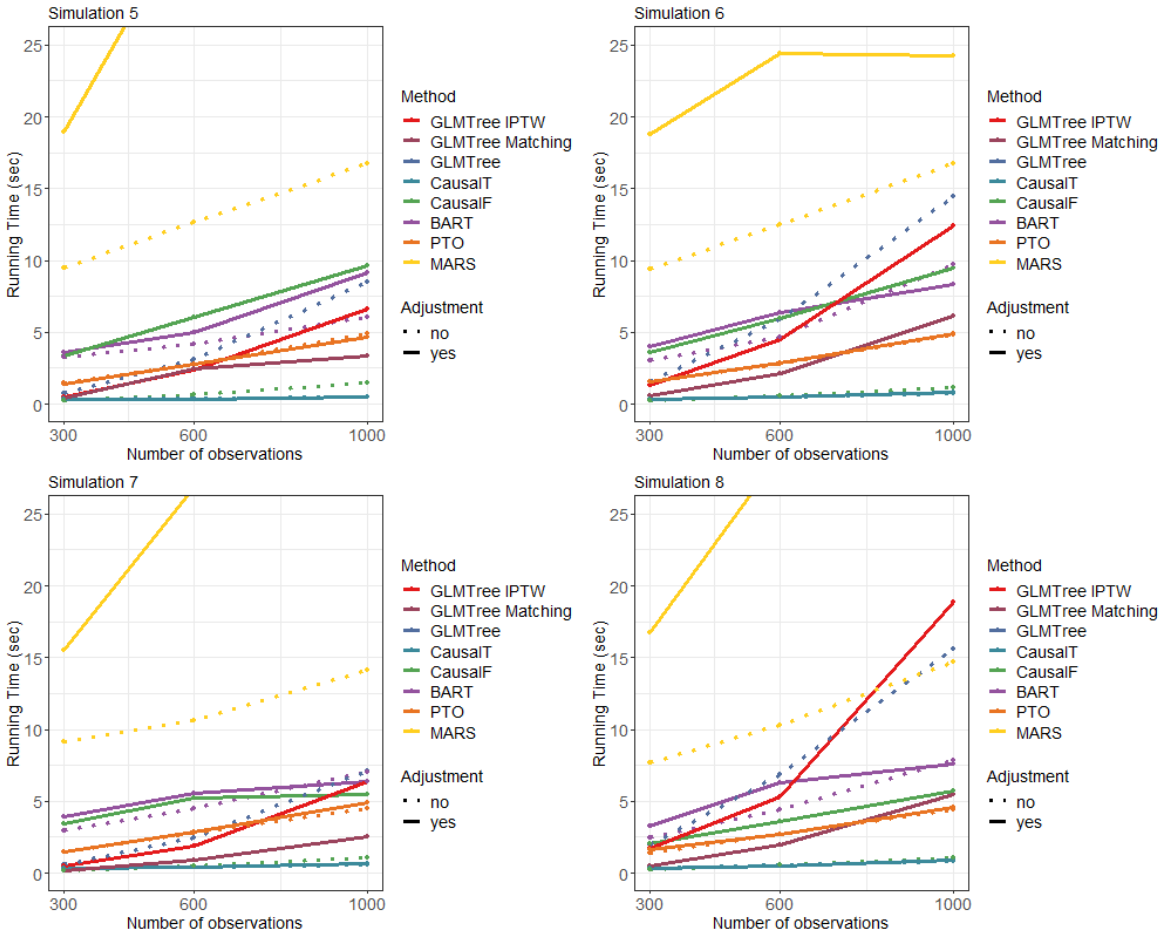

Figure A.21: Running time of methods for simulations 5-8

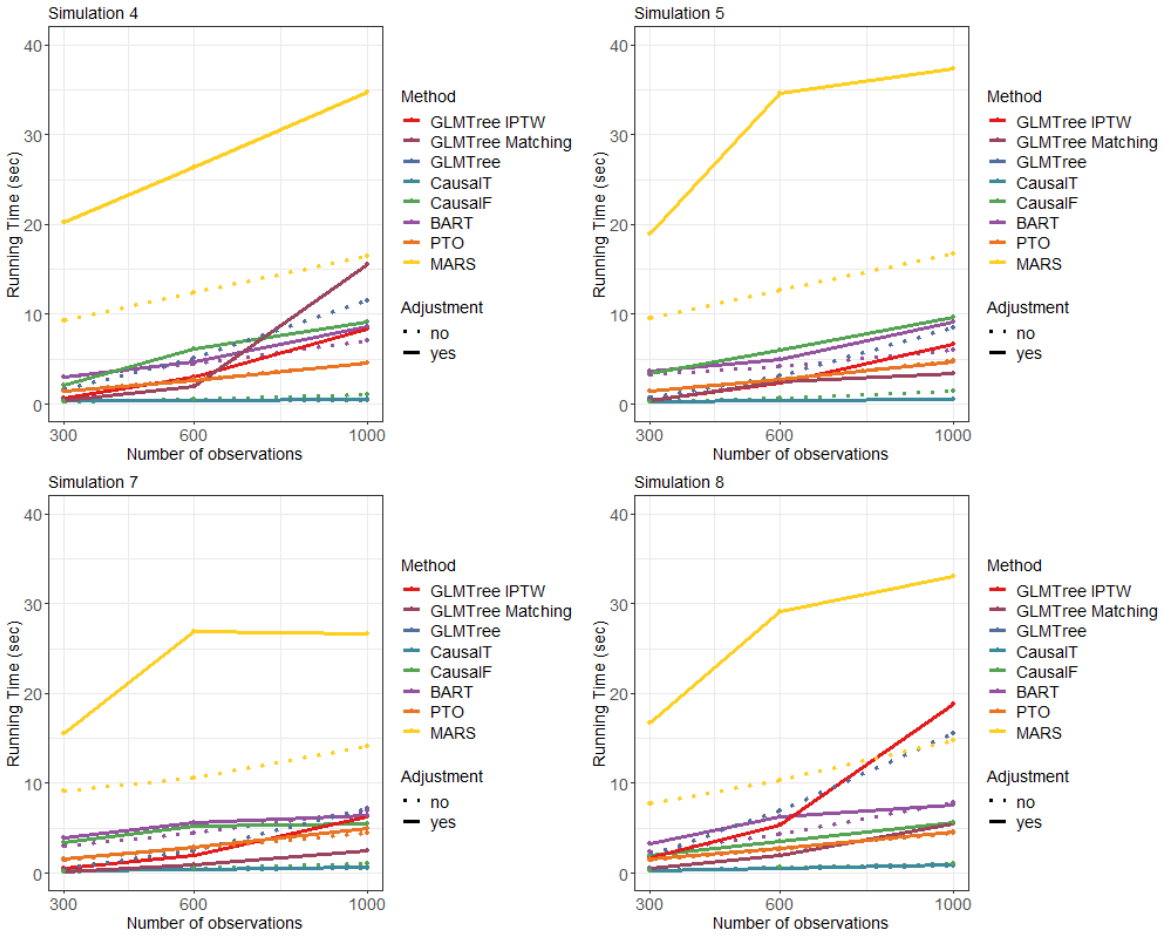

Figure A.22: Running time of methods with extended y-axis for simulations 4, 5, 7 and 8

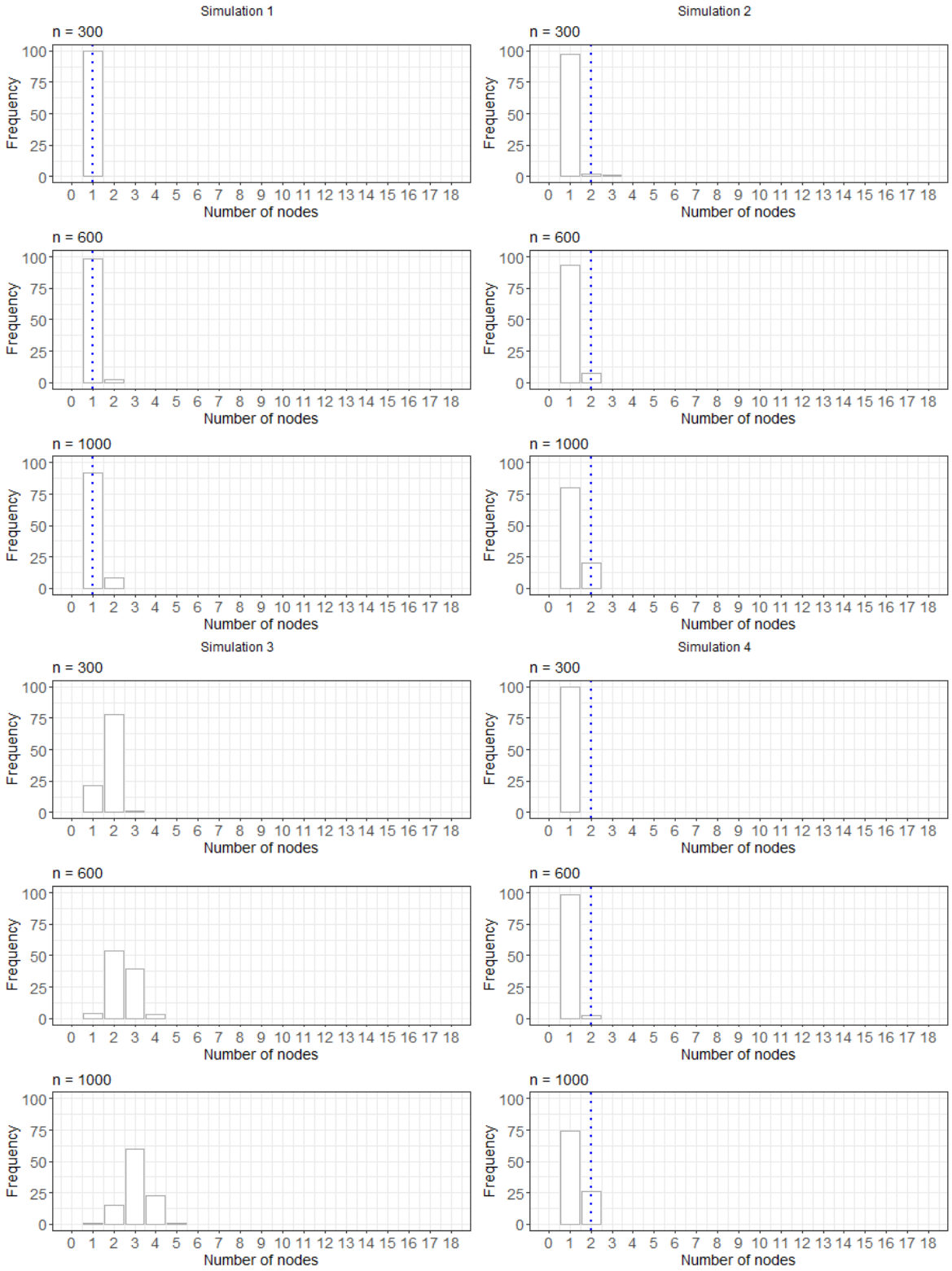

Figure A.23: Number of nodes in GLM tree for simulations 1-4

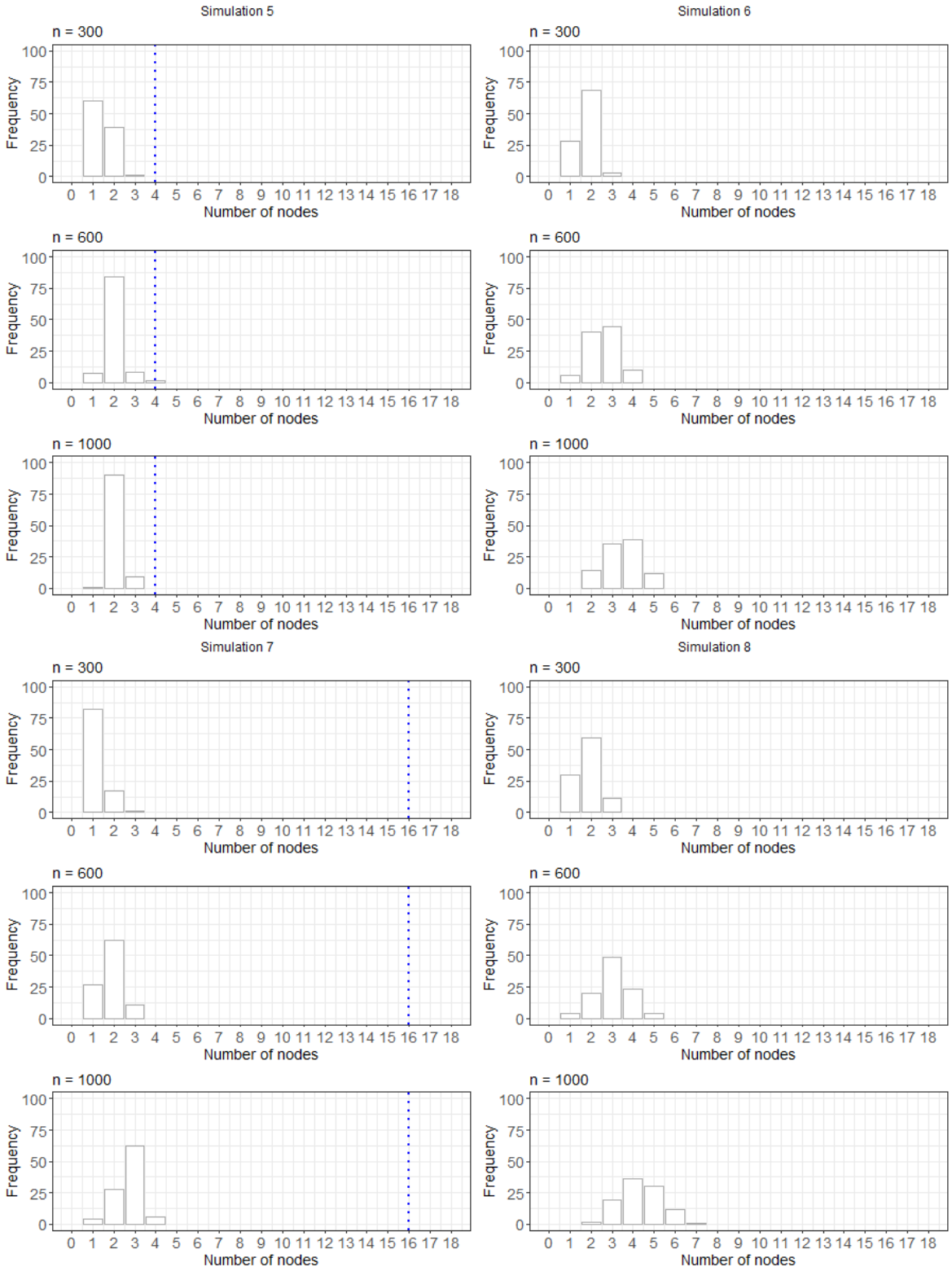

Figure A.24: Number of nodes in GLM tree for simulations 5-8

A.6 Further Analyses of GLM Trees

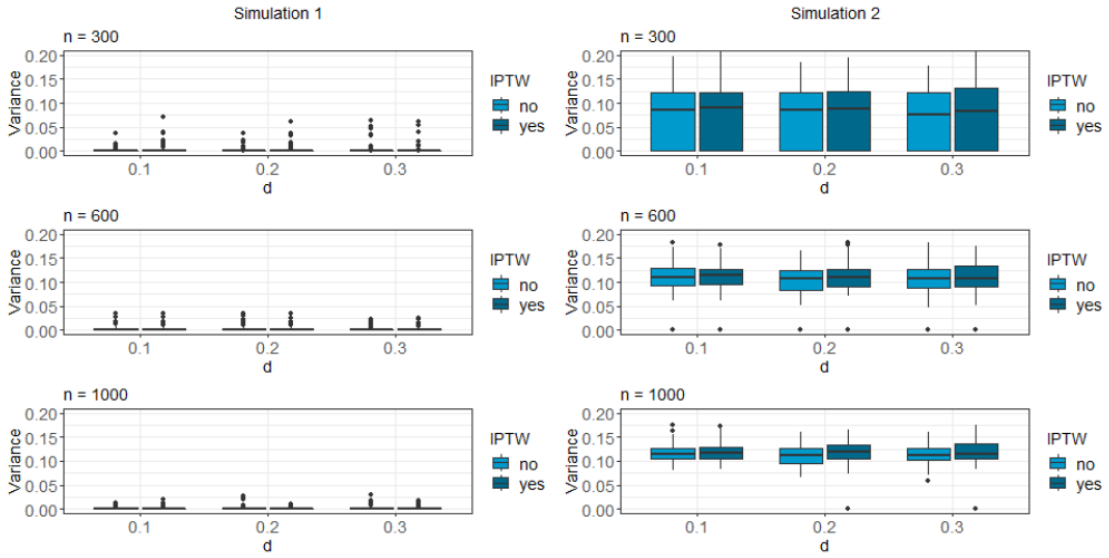

Figure A.25: Variance of GLM trees with varying propensity score

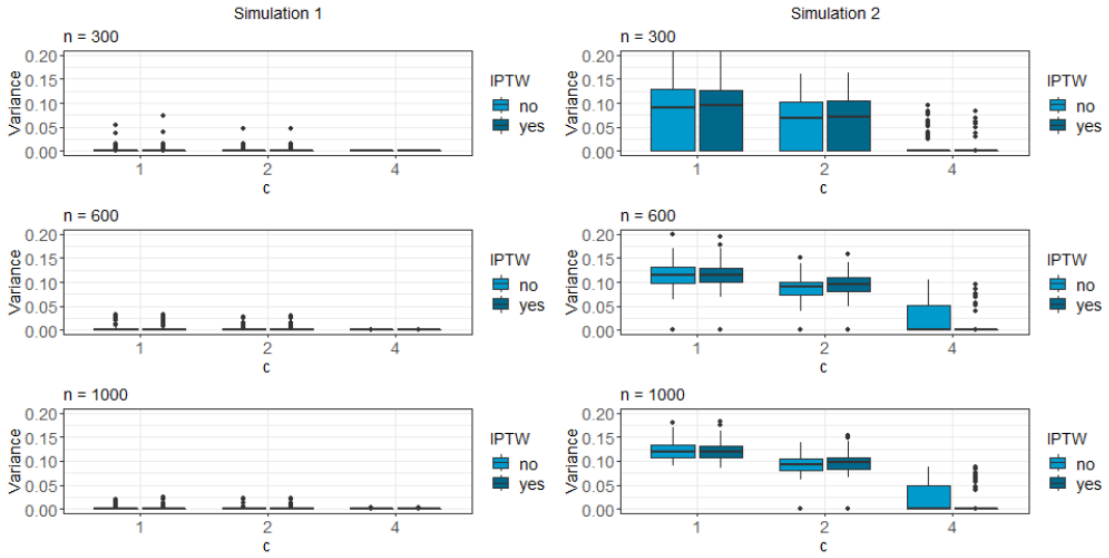

Figure A.26: Variance of GLM trees with varying coefficients

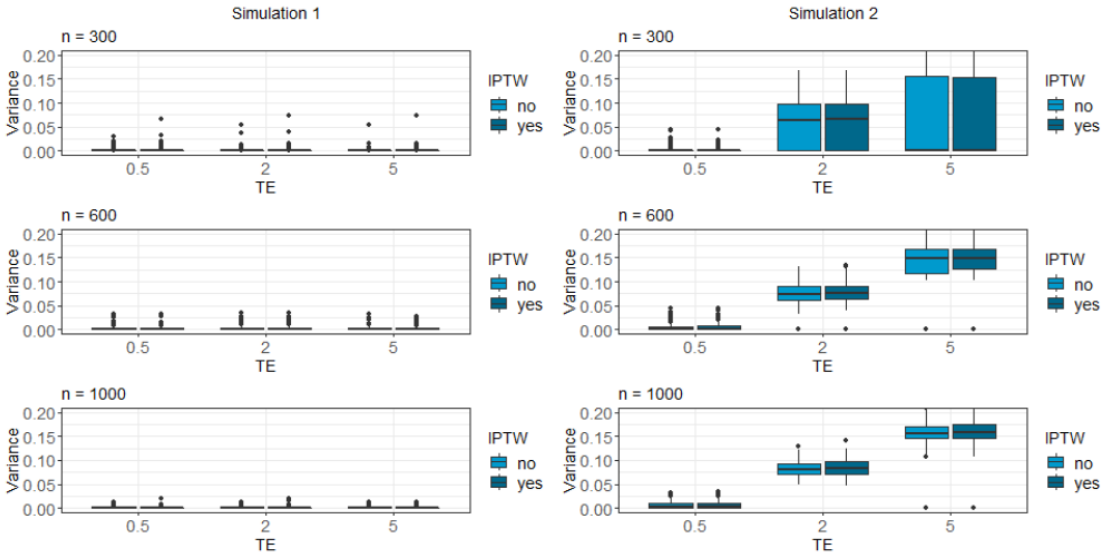

Figure A.27: Variance of GLM trees with varying treatment effects

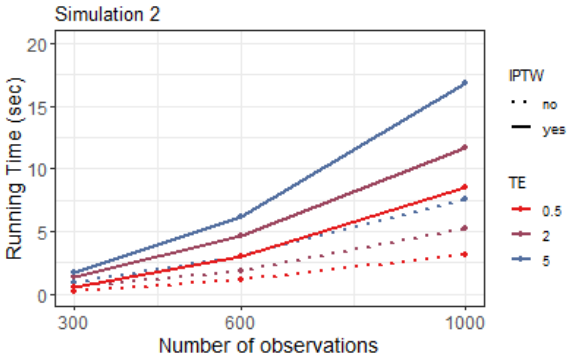

Figure A.28: Running time of GLM trees for different treatment effects with extended y-axis for simulation 2

B Electronic Appendix

The electronic appendix comprises a *Make_File*, a *README*, two folders (*Code* and *Results*) and the thesis as PDF. The *Make_File* runs the code step by step in the correct order and saves the results. The *README*-file explains the structure of the two folders (*Code* and *Results*).

The two folders are split into the sections of the present thesis: the choice of the *Propensity Score Model*, the *Main Simulation* and the *Further Analyses* part.

- The folder *Code* comprises:
  - *Propensity Score Model*: Contains all R-Scripts for the propensity model choice. On the one hand, the script *PS_model.R* is included. It calculates the RMSE for different scenarios with different methods and saves the resulting RMSEs. On the other hand, it comprises the script *PS_model_Plot.R* that plots the resulting RMSEs.
  - *Main Simulation*: Includes the R-Scripts for *Functions*, *Simulation* and *Plots* of the main analysis, i.e. comparing different methods.
    - The *Functions* folder consists on the one hand of the functions for the data generation (covariates, response and treatment) and the different Scenarios (1-8) (folder *Data*). On the other hand it contains the functions to calculate the predictions and performance of the methods (folder *Methods*). For each method, a function *calculate_prediction_"Method"* exists that fits the model and predicts values for a test dataset. The function *calculate_performance* calls these prediction functions, iterates it multiple times and calculates the RMSE, bias and variance. Furthermore, the running time is measured.
    - The *Simulation* R-Scripts call the functions and save the calculated values for each method separately.
    - The *Plots* folder contains R-Scripts for plotting the generated RMSE, bias, variance and running time for all methods. Furthermore, a R-Script that combines the plots for an appropriate presentation in the thesis is included. Additionally, it contains a folder *Methods* that comprises the scripts for plotting method specific features, like the treatment effects

and number of nodes/error messages for the GLM trees.

  - *Further Analyses*: Includes all R-Scripts that evaluates the IPTW for GLM trees. As in the main simulation part, it consists of a *Functions*, a *Simulation* and a *Plots* folder. The folder *Functions* contains the scenarios as well as the functions to calculate the predictions and the performances. The folders *Simulation* and *Plots* are split into the sections *VaryingPScore*, *VaryingCoefficient* and *VaryingTE*, respectively. Thus, the *Simulation* folder includes the R-Scripts to run simulations with a varying propensity score, coefficient and treatment effect. The *Plots* folder contains the R-Scripts for plotting the RMSE, bias, variance and running time and number of error messages of these simulations.

- The folder *Results* is as well divided into the sections *Propensity Score Model*, *Main Simulation* and *Further Analyses*, where each includes the generated data and plots. Next to the plots shown in the thesis, the number of nodes for matched datasets are saved. Furthermore, the plots illustrating the number of error messages for the GLM trees are included.

C References

Abrahams, E. (2008). Right Drug—Right Patient—Right Time: Personalized Medicine Coalition. *Clinical and Translational Science*, **1**, 11–12.

Athey, S. & Imbens, G. (2015). Recursive Partitioning for Heterogeneous Causal Effects. *arXiv:1504.01132 [econ, stat]*.

Athey, S. & Imbens, G.W. (2017). The State of Applied Econometrics: Causality and Policy Evaluation. *Journal of Economic Perspectives*, **31**, 3–32.

Athey, S., Imbens, G. & Kong, Y. (2016a). *CausalTree: Recursive partitioning causal trees*.

Athey, S., Imbens, G., Kong, Y. & Ramachandra, V. (2016b). An Introduction to Recursive Partitioning for Heterogeneous Causal Effects Estimation Using causalTree package. 17.

Athey, S., Tibshirani, J. & Wager, S. (2018). Generalized Random Forests. *arXiv:1610.01271 [econ, stat]*.

Austin, P.C. & Stuart, E.A. (2015). Moving towards best practice when using inverse probability of treatment weighting (IPTW) using the propensity score to estimate causal treatment effects in observational studies. *Statistics in Medicine*, **34**, 3661–3679.

Breiman, L. (2001). Random Forests. *Machine Learning*, **45**, 5–32.

Breiman, L., Friedman, J., Stone, C.J. & Olshen, R.A. (1984). *Classification and Regression Trees*. Taylor & Francis.

Chipman, H.A., George, E.I. & McCulloch, R.E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, **4**, 266–298.

Elze, M.C., Gregson, J., Baber, U., Williamson, E., Sartori, S., Mehran, R., Nichols, M., Stone, G.W. & Pocock, S.J. (2017). Comparison of Propensity Score Methods and Covariate Adjustment: Evaluation in 4 Cardiovascular Studies. *Journal of the American College of Cardiology*, **69**, 345–357.

Fahrmeir, L., Kneib, T. & Lang, S. (2009). *Regression: Modelle, Methoden und*

Anwendungen, 2nd edn. Springer-Verlag, Berlin Heidelberg.

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

Friedman, J.H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics*, **19**, 1–67.

Green, D.P. & Kern, H.L. (2012). Modeling Heterogeneous Treatment Effects in Survey Experiments with Bayesian Additive Regression Trees. *Public Opinion Quarterly*, **76**, 491–511.

Hahn, P.R., Murray, J. & Carvalho, C. (2017). Bayesian regression tree models for causal inference: Regularization, confounding and heterogeneity. 38.

Haoda Fu, J.Z. and D.E.F. (2016). Estimating optimal treatment regimes via subgroup identification in randomized control trials and observational studies. *Statistics in Medicine*, **35**, 3285–3302.

Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, 2nd edn. Springer-Verlag, New York.

Hernan, M. & Robins, J. (2018). Causal Inference Book.

Hill, J.L. (2011). Bayesian Nonparametric Modeling for Causal Inference. *Journal of Computational and Graphical Statistics*, **20**, 217–240.

Ho, D.E., Imai, K., King, G. & Stuart, E.A. (2007). Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference. *Political Analysis*, **15**, 199–236.

Ho, D.E., Imai, K., King, G. & Stuart, E.A. (2011). MatchIt: Nonparametric Preprocessing for Parametric Causal Inference. *Journal of Statistical Software*, **42**.

Holland, P.W. (1986). Statistics and Causal Inference. *Journal of the American Statistical Association*, **81**, 945–960.

Hothorn, T. & Zeileis, A. (2015). partykit: A modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research*, **16**, 3905–3909.

Hothorn, T., Hornik, K. & Zeileis, A. (2006). Unbiased Recursive Partitioning: A

Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, **15**, 651–674.

Imai, K. & Ratkovic, M. (2013). Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics*, **7**, 443–470.

Imbens, G.W. & Wooldridge, J.M. (2009). Recent Developments in the Econometrics of Program Evaluation. *Journal of Economic Literature*, **47**, 5–86.

Ishwaran, H. & Malley, J.D. (2014). Synthetic learning machines. *BioData Mining*, **7**.

King, G. & Nielsen, R. (2018). Why Propensity Scores Should Not Be Used for Matching. 34.

Knaus, M., Lechner, M. & Strittmatter, A. (2017). Heterogeneous Employment Effects of Job Search Programmes: A Machine Learning Approach. *SSRN Electronic Journal*.

Knaus, M.C., Lechner, M. & Strittmatter, A. (2018). Machine Learning Estimation of Heterogeneous Causal Effects: Empirical Monte Carlo Evidence. 114.

Lendle, S.D., Fireman, B. & Laan, M.J. van der. (2013). Targeted maximum likelihood estimation in safety analysis. *Journal of Clinical Epidemiology*, **66**, S91–S98.

Lu, M., Sadiq, S., Feaster, D.J. & Ishwaran, H. (2018). Estimating Individual Treatment Effect in Observational Data Using Random Forest Methods. *Journal of Computational and Graphical Statistics*, **27**, 209–219.

Luque-Fernandez, M.A., Schomaker, M., Rachet, B. & Schnitzer, M.E. (2018). Targeted maximum likelihood estimation for a binary treatment: A tutorial. *Statistics in Medicine*, **37**, 2530–2546.

McCulloch, R., Sparapani, R., Gramacy, R., Spanbauer, C. & Pratola, M. (2018). *BART: Bayesian additive regression trees*.

Powers, S., Qian, J., Hastie, T. & Tibshirani, R. *CausalLearning: Methods for heterogeneous treatment effect estimation*.

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

Qian, M. & Murphy, S.A. (2011). Performance guarantees for individualized treatment

rules. *The Annals of Statistics*, **39**, 1180–1210.

R Core Team. (2014). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rosenbaum, P.R. & Rubin, D.B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, **70**, 41–55.

Rubin, D.B. (1980). Randomization Analysis of Experimental Data: The Fisher Randomization Test Comment. *Journal of the American Statistical Association*, **75**, 591–593.

Schwab, P., Linhardt, L. & Karlen, W. (2018). Perfect Match: A Simple Method for Learning Representations For Counterfactual Inference With Neural Networks. *arXiv:1810.00656 [cs, stat]*.

Seibold, H., Zeileis, A. & Hothorn, T. (2016). Model-Based Recursive Partitioning for Subgroup Analyses. *The International Journal of Biostatistics*, **12**, 45–63.

Su, X., Tsai, C.-L., Wang, H., Nickerson, D.M. & Li, B. (2009). Subgroup Analysis via Recursive Partitioning. *SSRN Electronic Journal*.

Tian, L., Alizadeh, A.A., Gentles, A.J. & Tibshirani, R. (2014). A Simple Method for Estimating Interactions Between a Treatment and a Large Number of Covariates. *Journal of the American Statistical Association*, **109**, 1517–1532.

Tibshirani, J., Athey, S., Wager, S., Friedberg, R., Miner, L. & Wright, M. (2018). *Grf: Generalized random forests (beta)*.

Wager, S. & Athey, S. (2018). Estimation and Inference of Heterogeneous Treatment Effects using Random Forests. *Journal of the American Statistical Association*, **113**, 1228–1242.

Wendling, T., Jung, K., Callahan, A., Schuler, A., Shah, N.H. & Gallego, B. (2018). Comparing methods for estimation of heterogeneous treatment effects using observational data from health care databases. *Statistics in Medicine*, **37**, 3309–3324.

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York.

Xie, Y., Brand, J.E. & Jann, B. (2012). Estimating Heterogeneous Treatment Effects

with Observational Data. *Sociological Methodology*, **42**, 314–347.

Zeileis, A. & Hornik, K. (2007). Generalized M-fluctuation tests for parameter instability. *Statistica Neerlandica*, **61**, 488–508.

Zeileis, A. & Hothorn, T. Parties, Models, Mobsters: A New Implementation of Model-Based Recursive Partitioning in R. 39.

Zeileis, A., Hothorn, T. & Hornik, K. (2008). Model-Based Recursive Partitioning. *Journal of Computational and Graphical Statistics*, **17**, 492–514.

Zhang, B., Tsiatis, A.A., Davidian, M., Zhang, M. & Laber, E. (2012a). Estimating optimal treatment regimes from a classification perspective: Treatment regimes and classification. *Stat*, **1**, 103–114.

Zhang, B., Tsiatis, A.A., Laber, E.B. & Davidian, M. (2012b). A Robust Method for Estimating Optimal Treatment Regimes. *Biometrics*, **68**, 1010–1018.

Zhao, Y., Zeng, D., Rush, A.J. & Kosorok, M.R. (2012). Estimating Individualized Treatment Rules Using Outcome Weighted Learning. *Journal of the American Statistical Association*, **107**, 1106–1118.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources. I furthermore declare that this thesis has not been submitted to any other board of examiners yet.

Signature	Date