ESSAYS ON USING MACHINE LEARNING FOR CAUSAL INFERENCE

DISSERTATION

zur Erlangung des akademischen Grades
doctor rerum politicarum
(Doktor der Wirtschaftswissenschaft)

eingereicht an der

Wirtschaftswissenschaftlichen Fakultät
der Humboldt-Universität zu Berlin

von

Daniel Jacob

Präsidentin der Humboldt-Universität zu Berlin:
Prof. Dr.-Ing. Dr. Sabine Kunst

Dekan der Wirtschaftswissenschaftlichen Fakultät:
Prof. Dr. Daniel Klapper

Gutachter:
1. Prof. Dr. Stefan Lessmann
2. Prof. Dr. Wolfgang Karl Härdle

Tag des Kolloquiums: 22.10.2021

Abstract

Economic sectors in marketing, education, health care, the public sector, and many others now collect larger datasets and more granular data than never ever before. To use such data effectively, modern econometricians need to expand and rethink their toolbox. One field where such a transformation has already started is causal inference. The fact that only a low-dimensional parameter (the treatment effect) is of primary interest enables the incorporation of black-box machine learning methods to control for confounding bias or to explore heterogeneity in treatment effects. Parametric models can, however, not just be replaced. Certain pitfalls like loss in efficiency, overfitting, parameter identification, variable selection, efficient treatment allocation, and effectiveness need to be addressed.

This thesis aims to explore these and further issues, provide solutions, and develop new methods on how machine learning can be used to estimate causal parameters. I categorize novel methods to estimate heterogeneous treatment effects and provide a practitioner's guide for implementation. The parameter of interest is the conditional average treatment effect (CATE). It can be shown that an ensemble of methods is preferable to relying on one method. A special focus, with respect to the CATE, is set on the comparison of such methods and the role of sample splitting and cross-fitting to restore efficiency. Huge differences in the estimated parameter accuracy can occur if the sampling uncertainty is not correctly accounted for. One feature of the CATE is a coarser representation through quantiles. Estimating groups of the CATE leads to more robust estimates with respect to the sampling uncertainty and the resulting high variance.

This thesis not only develops and explores methods to estimate treatment effect heterogeneity but also to identify confounding variables as well as observations that should receive treatment. For these two tasks, this thesis proposes the outcome-adaptive random forest for automatic variable selection, as well as supervised randomization for a cost-efficient selection of the target group. Insights into important variables and those that are not true confounders are very helpful for policy evaluation and in the medical sector when randomized control trials are not possible.

Keywords: Causal Inference, Machine Learning, Heterogeneity, Sample Splitting, Variable Selection

Zusammenfassung

Industrien in der Kommunikation, Bildung, Gesundheit, dem öffentlichen Sektor und vielen anderen Bereichen sammeln größere und granuliertere Datensätze in einem noch nie dagewesenen Umfang. Um diese Daten am effektivsten zu nutzen, muss die moderne Ökonometrie ihren Werkzeugkasten an Modellen erweitern und neu denken. Das Feld, in dem diese Transformation am besten beobachtet werden kann, ist die kausale Inferenz. Da meist nicht alle, sondern nur ein Parameter (der Effekt von einem Treatment) von speziellem Interesse ist, macht es möglich sogenannte „black-Box“ Methoden zur Kontrolle von Selektionsverzerrung oder zum Auffinden von Heterogenität in Effekten von einem Treatment zu verwenden. Ein Ersetzen von parametrischen Modellen durch Machine Learning kann jedoch nicht ohne weiteres stattfinden. Stolpersteine wie sinkende Effizienz, Overfitting, die Schätzung des Parameters, Auswahl der wichtigen Variablen, Effiziente Treatment Verteilung und die generelle Effektivität müssen unbedingt beobachtet und untersucht werden.

Diese Dissertation verfolgt die Absicht diese und weitere Probleme zu untersuchen, Lösungen zu präsentieren und neue Methoden zu entwickeln Machine Learning zu benutzen, um kausale Parameter zu schätzen. Dafür werden in der Dissertation zuerst verschiedene neuartige Methoden, welche als Ziel haben heterogene Treatment Effekte zu messen, eingeordnet. Im zweiten Schritt werden, basierend auf diesen Methoden, Richtlinien für ihre Anwendung in der Praxis aufgestellt. Der Parameter von Interesse ist der „conditional average treatment effect“ (CATE). Es kann gezeigt werden, dass ein Vergleich mehrerer Methoden gegenüber der Verwendung einer einzelnen Methode vorzuziehen ist. Ein spezieller Fokus liegt dabei auf dem Aufteilen und Gewichten der Stichprobe, um den Verlust in Effizienz wettzumachen. Ein unzulängliches Kontrollieren für die Variation durch verschiedene Teilstichproben führt zu großen Unterschieden in der Präzision der geschätzten Parameter. Wird der CATE durch Bilden von Quantilen in Gruppen unterteilt, führt dies zu robusteren Ergebnissen in Bezug auf die Varianz.

Diese Dissertation entwickelt und untersucht nicht nur Methoden für die Schätzung der Heterogenität in Treatment Effekten, sondern auch für das Identifizieren von richtigen Störvariablen und einer effektiven Verteilung des Treatments an gewisse Beobachtungen. Um beide Probleme zu lösen, schlägt diese Dissertation sowohl die „outcome-adaptive random forest“ Methode vor, welche automatisiert Variablen klassifiziert, als auch „supervised randomization“ für eine kosteneffiziente Selektion der Zielgruppe. Einblicke in wichtige Variablen und solche, welche keine Störung verursachen, ist besonders in der Evaluierung von Politikmaßnahmen aber auch im medizinischen Sektor wichtig, insbesondere dann, wenn kein randomisiertes Experiment möglich ist.

Schlagworte: Kausale Inferenz, Machine Learning, Heterogenität, Stichprobenaufteilung, Variablen Selektion

Acknowledgments

I wish to express my deepest gratitude to my supervisor, Prof. Stefan Lessmann, whose excellent teaching and motivating ideas inspired me to start a PhD in machine learning. With him a short question ended in long and fruitful discussions about new research topics. I want to express my sincere gratitude to my second supervisor, Prof. Wolfgang Karl Härdle, for continuous support of my doctoral studies. His guidance throughout the academic journey has helped me reach this stage in my life.

I am indebted to my coauthors, especially Dr. Johannes Haupt and Dr. Robin Gubela who have gifted me their knowledge and time on countless occasions. I am deeply grateful to my colleagues and my fellow PhD students, among them Georg Keilbar, Marius Sterling, Elizaveta Zinovyeva, and many others, for always picking up the pen when discussing ideas in front of a whiteboard. I would also like to thank Dr. Quingliang Fan, whom I met during my research stay at Xiamen University, for great discussions and ideas. Advice and assistance given by Gabriel Okasa and Michael Knaus from the University St. Gallen has been a great help in writing part of this thesis.

I am deeply indebted to my friends and fellow researchers, Monique Reiske, Konstantin Häusler, Stefan Pauly, Tilman Fries, Kilian Kamkar, Gerry Koch and Marius Heinemann. Thank you so much for always taking the time to let me explain a new idea and for always providing me with helpful and inspiring suggestions. It was you that motivated me to stay curious and inspired me to follow up the path of academia.

I would also like to thank the students of the faculty for their curiosity and hard work. Special thanks go to Raphael Reule for the tremendous assistance with all the administrative work.

Finishing my dissertation required more than academic support. My deepest thanks go out to my parents, Thomas and Ute, and my sister Anna who supported me endlessly and believed in my talents. I cannot begin to express my profound gratitude to my wonderful wife Nora, who not only motivated me but influenced this thesis through her clear ideas for structure and her valuable gift to help me express my thoughts.

Finally, financial support from the DFG via the IRTG 1792 "High Dimensional Nonstationary Time Series" is gratefully acknowledged.

Contents

List of Figures

List of Tables

Abbreviations

ATE	Average Treatment Effect
AMD	Absolute Mean Difference
BART	Bayesian Additive Regression Trees
CATE	Conditional Average Treatment Effect
CART	Classification and Regression Tree
CF	Causal Forest
CI	Confidence Interval
CLAN	Classification Analysis
DGP	Data Generating Process
DR	Doubly-Robust
GATE	Group Average Treatment Effect
GRF	Generalized Random Forest
GLM	Generalized Linear Model
IP(T)W	Inverse Probability (Treatment) Weighting
ITE	Individualized Treatment Effect
LASSO	Least Absolute Shrinkage and Selection Operator
MAE	Mean Absolute Error
MCM	Modified Covariate Method
MCMC	Markov Chain Monte Carlo
ML	Machine Learning
MSE	Mean-Squared Error
OAL	Outcome-Adaptive Lasso
OARF	Outcome-Adaptive Random Forest
OLS	Ordinary least squares
OOB	Out of Bag
RCT	Randomized Controlled Trial
RF	Random Forest
RHC	Right Heart Catheterization
RMSE	Root-Mean-Squared Error
RRF	Regularized Random Forest
SD	Standard Deviation
SUTVA	Stable Unit Treatment Value Assumption
USA	United States of America

Chapter 1

Introduction

Information Systems, computer science, and econometrics often deal with different tasks. The former two aim to build regression and classification models from higher dimensional datasets or even unstructured data like images and text. They focus on dimension reduction and variable selection and hence smoothing. Most importantly, they offer a broad range of superior data-driven prediction models which we call machine learning (ML). Econometrics, on the other hand, is often interested in parameter estimation, even if this means trading off prediction accuracy. In this thesis, I deal with estimating a parameter from a treatment - the treatment effect. The most intuitive one might be the average treatment effect (ATE) which answers the question of which effect a treatment has on average. Its usage is unproblematic if we believe that the effect is homogeneous among all observations. However, this might be a strong assumption. Heterogeneity in treatment effects is, perhaps, a more realistic view. Estimating heterogeneity allows us to see a broader picture, namely if some people benefit more than the average and, more importantly, if there are people that even suffer from being treated. An important ingredient to identifying such subgroups is to observe pre-treatment characteristics about each observation. Based on their differences, we can estimate multiple average treatment effects, where the assumption of homogeneity within a subgroup is more plausible. This parameter is called the conditional average treatment effect (CATE). Having collected many covariates, one important question is if all of those are important. Machine learning can help with this question. It can perform data-driven variable selection to identify neighborhoods where the treatment effect is homogeneous (Athey et al., 2019).

Another interesting aspect of causal inference is the self-selection problem. If the treatment is not randomly assigned to observations then the distribution of characteristics might differ between the treatment and the control group. This results in a selection bias when estimating the treatment effect for which we have to control. There are different methods to do so, for example one can use the propensity score to weight the different proportions (Hirano & Imbens, 2001), build two models to estimate the counterfactual outcome (Hansotia & Rukstales, 2002), or partial out the effect that the characteristics have on the outcome and the treatment (Chernozhukov et al., 2018). All these methods have one common component: They use predicted values for each observation to control for selection bias.

Machine learning can help again in providing such prediction models. Using ML in this context is especially useful if the underlying functions are high-dimensional or non-linear.

However, simply applying ML algorithms and expecting to get closer to the true parameter does not work in practice. As we will see in the following chapters, there are certain pitfalls that we have to avoid through carefully designing an estimator that uses econometric theory and the prediction and selection power of ML. To be more precise, questions that we have to address are the following: How and to what extend do methods that aim to estimate the CATE differ from the true parameter in finite samples? What ML method should we use for each of the prediction functions? Are there differences depending on how the sample is split to avoid overfitting? How certain are we that the estimated parameter is true? Is there significant heterogeneity? Can we get more robust estimates when only looking at the most and least affected observations? What variables should we take into account for estimation? What are important variables and how do they drive the heterogeneity? This thesis aims to answer those questions and shows that using ML for causal inference is natural but not straightforward. In this thesis, I give an extensive overview of how ML can help to get a better understanding of the effects of treatment. I point out issues and address them by developing new methods and by providing guidelines for practitioners. I also deeply hope to encourage people to join the journey to better understand how ML and causal inference can be used together.

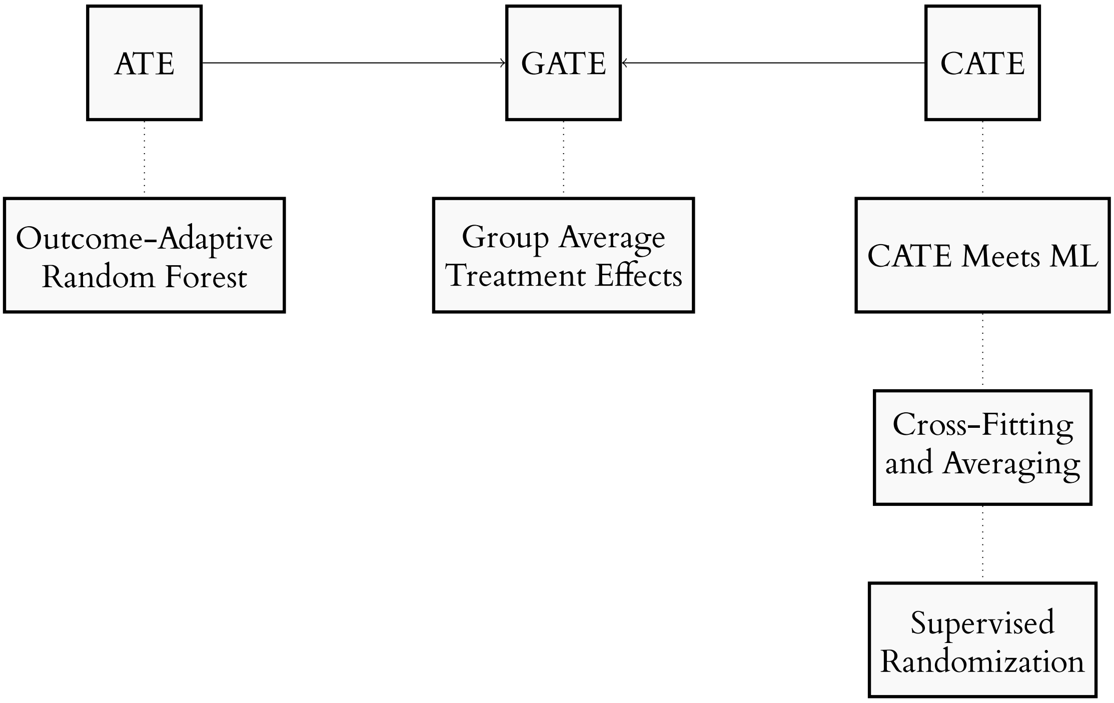

ATE
GATE
CATE
Outcome-Adaptive Random Forest
Group Average Treatment Effects
CATE Meets ML
Cross-Fitting and Averaging
Supervised Randomization

Figure 1.1: Three parameters in causal inference and the corresponding chapters.

Figure 1.1 provides an overview of the three estimators that this thesis focuses on. Starting with the ATE, as a single parameter of treatment, to the CATE that offers a parameter for each observation and the GATE that makes use of both concepts. Below, I list the corresponding chapters that deal with the three estimators.

Chapter 2 provides an overview of novel methods to estimate the CATE. I classify the considered estimators into two categories. The first one, so-called meta-learners, are methods that are flexible in the choice of the ML algorithm. The perhaps simplest meta-learner is the IPTW estimator where the propensity score can be estimated using, for example, a random forest, boosting methods, neural networks, or Bayesian methods. The resulting pseudo-outcome then has to be regressed on the covariates to get the CATE. An issue with this method arises if the propensity score takes on extreme values, making the IPTW estimator a non-robust method. Another attempt to estimate the CATE is to estimate the counterfactual outcomes by building two regression functions, one for the treated and one for the non-treated observations. Taking the difference of both outcomes for each observation again results in the CATE. This approach is known as the plug-in method since each observation is plugged in both functions to get the outcomes. It is also known as the T-learner since it uses two models. An issue with this approach is that the two regression functions try to minimize the mean squared error of the outcome, rather than the true CATE. This can result in two different functions where even different covariates are selected.

To solve both problems, the doubly-robust estimator uses inverse probability weighting on the residual of the outcome as an additional term on the T-learner. Combining the two approaches leads to more robust results as well as an unbiased estimator for the CATE (see Kennedy (2020) for recent results on the doubly-robust estimator). The X-learner, proposed by Künzel et al. (2019), follows the logic of the T-learner but uses the already observed outcome and only the estimated counterfactual to create treatment effects for each observation. Using a true loss function that can be minimized is the basis of the R-learner (Nie & Wager, 2020). The idea is to partial out any confounding effect and to use the residuals to minimize a loss function for the CATE. The R-learner is closest to the generalized random forest (GRF) by Athey et al. (2019), which belongs to the second category of methods that I considerer in this chapter, namely modified ML methods. These methods adapt existing ML algorithms to estimate the CATE and hence do only partially offer some flexibility in choosing a different algorithm. Other methods that fall into this category are causal BART methods by Hill (2011) and recently by Hahn et al. (2020) as well as causal boosting proposed by Powers et al. (2018).

In this chapter, I explain every method in detail and apply them to two empirical datasets as well as to simulated data where the true CATE is known. For all four examples, I show the influence of different sample splitting procedures and provide an attempt to construct confidence intervals. I further show the effect of different ML methods and how to use an ensemble with cross-validation to not only select tuning parameters but to find the best ML algorithm for each function that has to be estimated. To easily apply all methods in practice, I provide Quantlets for all methods and the analysis of cross-fitting and confidence intervals.

Chapter 3 proposes a natural extension of the outcome-adaptive lasso (OAL), developed by Shortreed and Ertefaie (2017). Their method aims to estimate the ATE using an adaptive variable selection approach. Using the inverse probability of treatment weighting (IPTW) estimator to control for selection bias in observational studies, one has to estimate the propensity score. The authors noticed that this estimator is more efficient if the propensity score model only uses variables that are indeed confounders and excludes all other variables. To do so, they propose a two-step estimation procedure. First, the outcome is regressed via a linear model on all covariates and the treatment variable. Second, the propensity score is estimated via an adaptive lasso using the coefficients from the first step regression as an additional penalization term. An important limitation of the OAL approach is that it is restricted to parametric models. Both the linear outcome model and the propensity score model have to be correctly specified. However, to properly control for selection bias, one would like to collect as many pre-treatment characteristics as possible. This could even mean that one has more covariates than observations. Another realistic assumption is that the dependence of the characteristics on the outcome and or treatment is non-linear. In both cases, the OAL method would fail to select the correct covariates and would even produce biased estimates.

To address both issues, I propose the outcome-adaptive random forest (OARF). This approach is designed to allow high-dimensional datasets and the outcome or propensity score function to be non-linear and potentially complicated. First, instead of a linear outcome model a random forest is used and instead of coefficients I estimate carefully designed standardized feature importance values. These values are used in a regularized random forest to only select variables that are predictive of the outcome. Variables that have no association with either treatment and outcome and variables that are only predictive of the treatment are not included. To evaluate the OARF, I perform a Monte Carlo study investigating the bias, the variance, and confidence interval coverage. I show a numerical convergence rate by varying the sample size to visually see the rate at which the convergence to the true parameter occurs. The accuracy of the variable selection is measured by the inclusion probability over all Monte Carlo iterations.

Chapter 4 investigates the finite sample performance of estimators that use different sample splitting and cross-fitting techniques. When estimating the CATE using ML, sample splitting is a necessary step to avoid overfitting and to make regularity conditions more plausible. Let us say we want to estimate the CATE using the doubly-robust estimator. This estimator contains three nuisance functions, the two conditional mean functions from the outcome, given treatment status, and the propensity score. Asymptotically, it would be sufficient if we split the sample into two equal parts and use one part to train the three functions and the other part for estimating the nuisance parameter. However, it turns out that the way the sample splitting procedure is build has a strong influence on the estimation accuracy. For example, when only using half of the sample to estimate the parameter, the estimator is less efficient than an estimator that can use the whole sample. One way to obtain the full efficiency would be to estimate two parameters, both on the counterpart of the sample used to train the models and average them.

In this chapter, I propose twelve different ways how the sample can be split, how efficiency can be restored, and how median averaging adds to decreasing an additional bias based on "by chance" sample splitting. I use four novel methods, the doubly-robust estimator, the R-, T- and the X-learner. All twelve sample splitting procedures are evaluated on 15 different data generating processes and for each meta-learner method. I also vary the sample size (500, 2000, 4000, and 8000) such that each estimator is evaluated on 240 different settings (4 methods, 4 sample sizes, and 15 DGP's). I find that the estimator with the lowest MSE uses 5-fold sample splitting, cross-fitting, and median averaging. The reason behind the 5-fold is the following: The more complicated the nuisance function is, the harder it is to approximate such a function. One important factor is the number of observations. In the 5-fold setting, only one fold is used for estimation allowing 80% of the data to be used for training. As in cross-validation, the next fold is used for estimation and so on, until each fold was used once. The resulting 5 parameters are then averaged.

In the previous chapters, I propose methods that deal with the ATE and the CATE. The ATE is a much simpler parameter than the CATE. The OARF is designed to decrease bias and primarily the variance. For the CATE, the uncertainty through sample splitting increases the variance in the CATE estimates for different methods. There is a tradeoff between a robust estimate and exploring heterogeneity when using ML algorithms. However, designing an estimator that only uses features of the CATE and somehow works as an ATE for such features could decrease this tradeoff.

Chapter 5 proposes such a method to provide valid estimation and inference for a causal interpretation of parameters. The parameters of interest, the features of the CATE, are the most and least affected observations. The idea is to find groups of observations depending on the estimated treatment effect heterogeneity. The method consists of three steps. First, I apply a doubly-robust estimator to estimate the conditional average treatment effect. I call this function "score function". Second, I orthogonalize the effect that the covariates have on both, the outcome variable unconditional of the treatment assignment as well as the treatment variable. Third, I use a linear regression model to estimate the group average treatment effects. I also use sample splitting as a form of cross-fitting by using the auxiliary sample to estimate the score function via the doubly robust estimator. I then use the main sample to predict the final score function which is used in the parametric step. In this way, I limit the danger of overfitting. To average out the uncertainty of a specific sample splitting, I repeat this procedure multiple times and take the median over the coefficients for the most and least affected. In this chapter, I consider splitting the data into five groups. The first group has the smallest effect and the last group contains the most affected observations. In simulations, I find that as soon as selection bias is introduced in the data generating process, the MSE and squared BIAS is smaller for the GATE method compared to the average of the groups from the CATE directly. Even in a randomized control trial that has an imbalanced treatment assignment distribution, the GATE method performs better. This result not only holds for the most and least affected but all of the five groups.

Chapter 6 follows a different logic to deal with selection bias and uncertainty while allowing individual targeting of observations. Randomized controlled trials are costly because there is no targeting policy. Such a policy could, for example, be that only customers with a higher expected gain from treatment are targeted. By doing so, the evaluation of such a treatment allocation is difficult since the targeting is no longer randomized. This chapter proposes a method to select cost-optimized data and selection bias correction for evaluation. We call this method supervised randomization. It is supervised since we can control the degree and form of selection bias that is used during targeting. Given that, we apply a doubly-robust method to fully correct for the introduced selection bias. This allows us to get an unbiased estimate of the treatment effect. In an empirical analysis, we show that our method can reduce the cost of an experimental campaign by 7.1 to 9.4% compared to full randomization and 2.9 to 8.2% compared to imbalanced randomization. We also show that, when applying the known oracle propensity score, the selection bias can be fully controlled for.

Bibliography

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *Annals of Statistics*, *47*(2), 1148–1178. https://doi.org/10.1214/18-AOS1709

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, *21*(1), C1–C68. https://doi.org/10.1111/ectj.12097

Hahn, P. R., Murray, J. S., & Carvalho, C. M. (2020). Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects. *Bayesian Analysis*, *15*. https://doi.org/10.1214/19-BA1195

Hansotia, B., & Rukstales, B. (2002). Incremental value modeling. *Journal of Interactive Marketing*, *16*(3), 35–46. https://doi.org/10.1002/dir.10035

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, *20*(1), 217–240. https://doi.org/10.1198/jcgs.2010.08162

Hirano, K., & Imbens, G. W. (2001). Estimation of causal effects using propensity score weighting: An application to data on right heart catheterization. *Health Services and Outcomes research methodology*, *2*(3), 259–278. https://doi.org/10.1023/A:1020371312283

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

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, *116*(10), 4156–4165. https://doi.org/10.1073/pnas.1804597116

Nie, X., & Wager, S. (2020). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*. https://doi.org/10.1093/biomet/asaa076

Powers, S., Qian, J., Jung, K., Schuler, A., Shah, N. H., Hastie, T., & Tibshirani, R. (2018). Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in Medicine*, *37*(11), 1767–1787. https://doi.org/10.1002/sim.7623

Shortreed, S. M., & Ertefaie, A. (2017). Outcome-adaptive lasso: Variable selection for causal inference. *Biometrics*, *73*(4), 1111–1122. https://doi.org/10.1111/biom.12679

Chapter 2

CATE meets ML - Conditional Average Treatment Effect and Machine Learning

Publication

Jacob, D. (2021). CATE meets ML - The Conditional Average Treatment Effect and Machine Learning, *Digital Finance*

Abstract

For treatment effects - one of the core issues in modern econometric analysis - prediction and estimation are two sides of the same coin. As it turns out, machine learning methods are the tool for generalized prediction models. Combined with econometric theory, they allow us to estimate a personalized treatment effect - the conditional average treatment effect (CATE). In this tutorial, we give an overview of novel methods, explain them in detail, and apply them via Quantlets in real data applications. We study the effect that microcredit availability has on the amount of money borrowed and if 401(k) pension plan eligibility has an impact on net financial assets, as two empirical examples. The presented toolbox of methods contains meta-learners, like the DR, R-, T- and X-learner, and methods that are specially designed to estimate the CATE like causal BART and generalized random forest. In an additional simulation study we further compare the methods to observe patterns and similarities.

JEL Classification: C15, C21, D14, G21

Replication code is available on Quantlet .

2.1 Introduction

Estimation and prediction of treatment effects are important tasks for every economist and financial econometrician since treatment effects are often the basis for policy and business decisions. As an illustration, let us look at an idea of microcredits, dating back to Muhammad Yunus, a Nobel Price winner, who discovered in 1976 that very small loans could make a disproportional difference to a poor person. Microcredits work as shown in Figure 2.1. They can increase investments since such credit is easy to get and pay back. Business activity is hence more flexible and could be improved. Increasing gains from a business could increase the household income and further allow for more savings which can be invested in, for example, education.

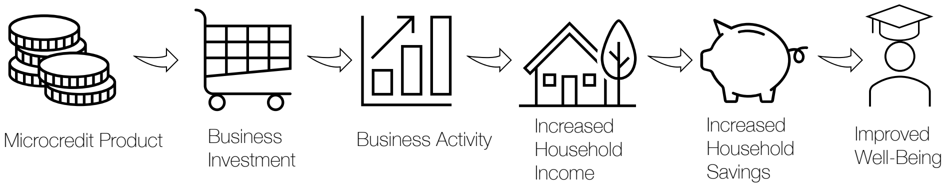


Figure 2.1: The theory of microcredits.

This specific example was recently applied by Crépon et al. (2015) who studied the setting where certain villages in Morocco get access to microcredit (the treatment group) while others don't (the control group). As economists, one is interested in the effect that microcredit availability has on the amount of loans which could be an indicator of how demanded such microcredits are. Since we observe certain characteristics for each household, we can condition on such observed variables to see if there is heterogeneity in the effect from microcredit. Figure 2.2 shows an example of what we want to do. The goal is to find subgroups based on characteristics where we believe that the treatment effect is different. As an example, we can partition the households by age and compare young vs. older household members in terms of their effect on microcredit. In both subgroups, we need to make sure that we observe people that are treated and others that did not receive treatment. We can estimate the average treatment effect (ATE) for the young household members, for example, by taking the difference of their mean outcome given treatment status. We repeat this for the subgroup of older households. Recent methods to estimate the ATE using nonparametric methods on the whole sample include target maximum likelihood estimation (TMLE) (van der Laan, 2010) and double machine learning (Chernozhukov, Chetverikov, et al., 2018). If the data has many covariates (let us say it has high-dimensionality) and if we don't know which specific subgroup we should focus on, as is the case here, we can use methods that are presented in this tutorial. These methods estimate a treatment effect for each observation based on their covariates, the conditional (on covariates) average treatment effect (CATE). In a further step, we can then look at the heterogeneity and try to link characteristics that are drivers for different treatment effects.

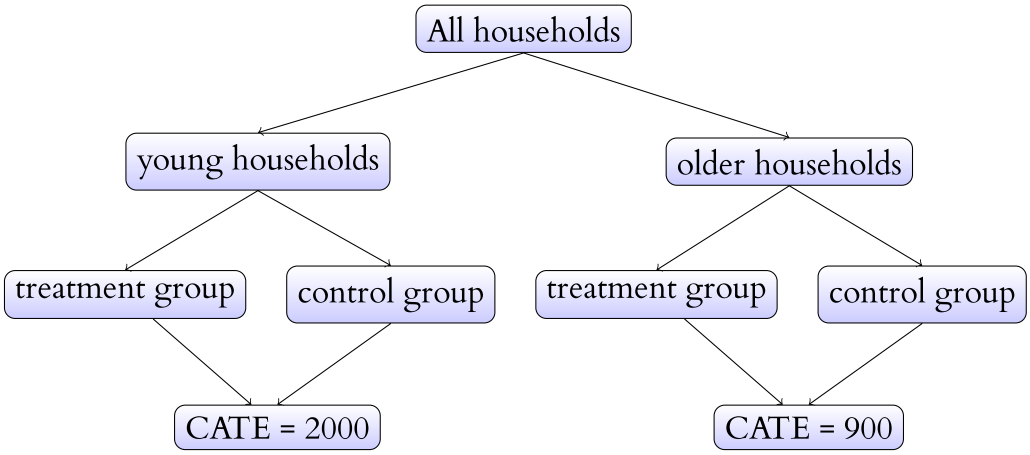


Figure 2.2: CATE example for microcredits

The high-dimensionality of a dataset does not necessarily mean that one has more covariates than observations by default. However, if we are unsure about the structural form, we could include interaction and quadratic terms, and soon the number of dimensions increases. For example, if we have 1000 observations and 30 covariates, then by only including quadratic interactions the amount of covariates increases to 495. Including up to cubic terms leads to a dimension of 5455. If we further assume that only a few covariates are dependent on the outcome and the treatment (often called the approximate sparsity assumption), the task transfers into a selection problem where standard parametric models are limited and we might want to use machine learning (ML) methods. The reason why this is the case is either that we have more covariates than observations or that the functional forms are complex and we don't know which interaction terms to include in a linear model.

Machine learning is not easily defined. It contains many algorithms with the main focus of prediction (regression), classification, and grouping tasks like clustering. While clustering, as a form of dimensionality reduction, only uses covariates but not outcomes with labels, we call this branch unsupervised ML. The counterpart is called supervised ML. Supervised ML, in general, uses a set of covariates to predict an observed outcome. When talking about prediction we mean the following: Construct an estimator $\hat{\mu}(x)$ of $\mathsf{E}[Y|X_i = x]$ using Y and a set of covariates from some training set and predict the values of Y from an independent test set. The goal is to minimize deviations between the true outcome and predicted outcomes from the test set. Note that this is in contrast to the term forecasting. The only assumption so far is that the observations are independent and that the joint distribution of X and Y in the training set is the same as that of the test set. To achieve the goodness of fit (for example minimize the average of the mean-squared error), in an independent test set many alternative models are estimated and the model that maximizes a criterion is selected. We will talk about cross-validation - a concept for model selection - later. The key is that the functional form is mostly determined as a function of the data. Regularization together with systematic model selection may be the main advantages of ML methods. When we talk about ML in this tutorial, we mean supervised machine learning models that are used to make predictions. For a detailed discussion about machine learning in economics see Mullainathan and Spiess (2017) and Athey (2019).

How do we get from prediction to causal inference? A simple, pure prediction approach to get the CATE is to estimate two conditional mean functions, one for the treated observations and one for the non-treated (the control group). For each observation, we can predict the outcome under treatment and control by plugging each observation into both functions. Taking the difference between the two outcomes results in the CATE. Mapping the support of X on Y is a classic regression task for which machine learning methods are well suited to find generalizable predictive patterns. Since we are only interested in getting a good prediction of the conditional mean, we do not need to know the underlying structural form of this function which enables vanilla ML methods to be sufficient. We call such functions, where the parameters are not of immediate interest, a nuisance function. While the above example of estimating the CATE is quite simple and intuitive, we will see that there are more efficient or automated methods to estimate heterogeneous treatment effects. We will also see that while prediction models are easy to evaluate, causal parameters are not. This is mainly since the objective is different. In prediction, we can optimize a goodness of fit criterion since we observe the true outcome. The causal parameter, however, is never observed in any dataset. As in econometrics, we need to carefully design methods that aim to estimate such parameters of interest, apply statistical theory and expand the set of assumptions to interpret a parameter as causal.

This tutorial is structured as follows. First, we provide an overview of the potential outcome framework and state the necessary assumptions to interpret our parameter of interest as a causal parameter. We then explain different methods that we consider, methods that are very flexible in the choice of the ML algorithm, and methods that are developed to estimate the CATE, mostly relying on tree-based algorithms. As in classical ML, we make use of sample splitting to limit overfitting and allowing for less restrictive assumptions on the nuisance functions. We cover explanations on why and how to do sample splitting and cross-validation. Next, we investigate two empirical datasets, the microcredit example, and the 401(k) pension plan survey. Last, we include a simulation study where we generate the true treatment effect. This allows us to directly compare all different methods in terms of accuracy. Whenever possible we provide and link to Quantlets that are ready-to-use code snippets to implement the discussed methods (the Quantlets are all written in R). The files are not only a replication code for the empirical analysis and the simulation study but contain functions to implement novel methods that aim to estimate the CATE directly. During this tutorial, we will use the terms model, method, and algorithm interchangeably.

Figure 4.4 gives an idea of how a causal structure may look. In the first graph, only the treatment has an impact on the outcome while the second graph also includes covariates that might make the treatment effect dependent on some characteristics. The same is true for the third graph but now the covariates also influence the treatment probability. We say that such a setting is from an observational study since the researcher has no control of the treatment assignment. The first two settings can be seen as a randomized controlled trial (RCT) but only in the second and third can we hope to observe treatment heterogeneity and hence estimate the CATE.

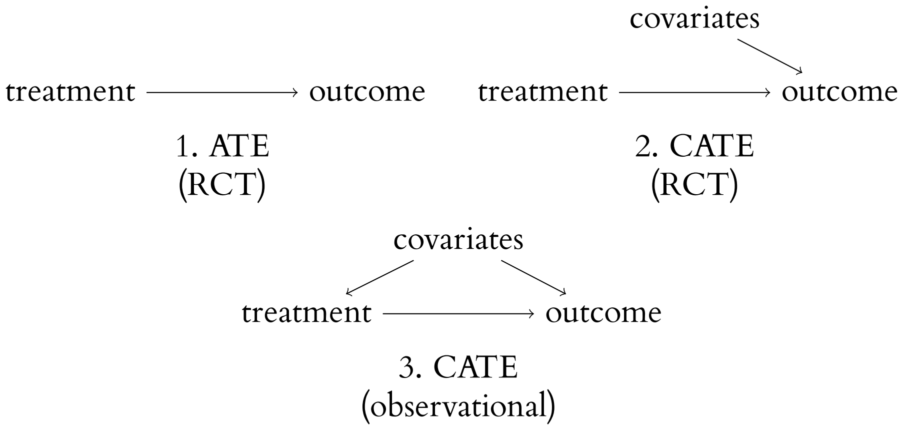


Figure 2.3: Simple causal diagrams - from ATE to CATE.

2.2 Methods

Let us start with an introduction of the potential outcome framework for which we use the following notations: Each observation has two potential outcomes, Y^1 and Y^0 of which we only observe one, namely the former if someone was treated or the latter if not. The binary treatment indicator is $D \in \{0; 1\}$ and we denote observed covariates $X \in \mathbb{R}$. To interpret the estimated parameter as a causal relationship, the following assumptions are needed; see, for example, Rubin (1980):

1. Conditional independence (or conditional ignorability/exogeneity or conditional unconfoundedness):

$$\left(Y_i^1, Y_i^0\right) \perp\!\!\!\perp D_i | X_i.$$

2. Stable Unit Treatment Value Assumption (SUTVA) (or counterfactual consistency):

$$Y_i = Y_i^0 + D_i(Y_i^1 - Y_i^0).$$

3. Overlap Assumption (or common support or positivity):

$$\begin{aligned} \forall x \in supp(X_i), \quad 0 < P(D_i = 1 | X_i = x) < 1, \\ P(D_i = 1 | X_i = x) \overset{\text{def}}{=} e(x). \end{aligned} \tag{2.1}$$

4. Exogeneity of covariates:

$$X_i^1 = X_i^0.$$

Assumption 1 together with Assumption 4 is very natural since they state that the treatment assignment is independent of the two potential outcomes and that the covariates are not affected by the treatment. Assumption 2 ensures that there is no interference, no spillover effects, and no hidden variation between treated and non-treated observations. Assumption 3 states that no subpopulation defined by $X_i = x$ is entirely located in the treatment or control group, hence the treatment probability needs to be bounded away from zero and one. Equation (4.2) is called the propensity score.

Now we define the conditional expectation of the outcome for the treatment or control group as

$$\mu_d(x) = \mathsf{E}[Y_i | X_i = x, D_i = d] \quad with \quad D \in \{0, 1\}.$$

If we don't use any subscript, we refer to this function as the general conditional expectation.

Our parameter of interest is the CATE ($\tau(x)$), which is formally defined as:

$$\tau(x) = \mathsf{E}\left[Y_i^1 - Y_i^0 \mid X_i = x\right] = \mu_1(x) - \mu_0(x). \tag{2.2}$$

Equation 2.3 shows how the two conditional mean functions can represent the two potential outcomes and hence, by taking the difference, lead to the CATE.

$$\begin{aligned}\tau(x) &= \mu_1(x) - \mu_0(x)\\ &= \mathsf{E}\left[Y_i \mid D_i = 1, X_i = x\right] - \mathsf{E}\left[Y_i \mid D_i = 0, X_i = x\right]\\ &= \mathsf{E}\left[Y_i^1 \mid D_i = 1, X_i = x\right] - \mathsf{E}\left[Y_i^0 \mid D_i = 0, X_i = x\right]\\ &= \mathsf{E}\left[Y_i^1 \mid X_i = x\right] - \mathsf{E}\left[Y_i^0 \mid X_i = x\right]\\ &= \mathsf{E}\left[Y_i^1 - Y_i^0 \mid X_i = x\right]\end{aligned} \tag{2.3}$$

This estimator is of special interest in many areas like medicine or policy actions since it tells us if there are differences in the treatment effect in the population and how big these differences are. It could be, for example, that the average treatment effect of a policy is +2, containing half of the people with a treatment effect of +6 and the other half of −2. Instead of treating everyone, we should only treat people that have a positive effect from the policy (if positive means better). If this is not possible, let us say due to legal or ethical reasons, the policy should not be implemented at all. The CATE will tell us the exact distribution of the effects and, at best, allows us to identify subgroups. To estimate the CATE, we are not primarily interested in the coefficient from regressing X on Y, nor are we interested in the coefficients from the propensity score model. What we want instead is to have a good approximation of the function and hence good estimates from e.g. $\mu_1(x)$ and $\mu_0(x)$. This is why ML methods are well suited for the job.

When reviewing recently proposed methods for the estimation of the CATE, we can categorize them into two groups. The first group contains methods that are built on off-the-shelf machine learning methods (such as the lasso, random forest (RF), Bayesian additive regression trees (BART), boosting methods, or neural networks). Since the base learners are not designed to estimate the CATE directly, the literature calls them meta-learners or generic ML algorithms. The second group of methods alters existing machine learning methods in a way that they can be used to estimate the CATE directly (examples are causal boosting by Powers et al. (2018), causal forest by Athey et al. (2019) or Bayesian regression tree models for causal inference by Hahn et al. (2020)). See Künzel et al. (2019) for a comparison between meta-learners like the S-, T-, and X-learner as well as the causal forest in a simulation study. Knaus et al. (2020) compare meta-learners like inverse probability weighting (IPW) estimator, doubly-robust (DR), modified covariate method (MCM), R-learner, and different versions of the causal forest in an empirical Monte Carlo study while Nie and Wager (2020) compare their R-learner with the S-, T-, X- and U-learner as well as causal boosting. Regarding the base learners (the ML methods), Künzel et al. (2019) use a random forest and BART. Knaus et al. (2020) use RF and lasso while Nie and Wager (2020) use boosting and the lasso for the estimation of the nuisance functions and the treatment effects. In high dimension, the use of machine learning methods, such as boosting or random forests to estimate the propensity score, works quite well as McCaffrey et al. (2004) and Wyss et al. (2014) show. The estimation of probabilities given a large set of covariates is nothing less than a prediction problem in where ML methods are superior. In Table 2.1 we list popular methods by category, including links to the Quantlets. The references refer to recent papers that use these methods and provide theoretical properties.

Table 2.1: Methods to estimate CATE

Category	Method	Reference	Quantlet
Meta-Learner	DR-learner	Kennedy (2020)	Q$_{DR}$
	IPW-learner	Horvitz and Thompson (1952)	Q$_{IPW}$
	R-learner	Nie and Wager (2020)	Q$_{R}$
	S-learner	Hill (2011)	Q$_{S}$
	T-learner	Hansotia and Rukstales (2002)	Q$_{T}$
	X-learner	Künzel et al. (2019)	Q$_{X}$
Modified ML Methods	Causal BART	Hahn et al. (2020)	Q$_{CBART}$
	Causal Boosting	Powers et al. (2018)	Q$_{CB}$
	Causal Forest	Athey et al. (2019)	Q$_{GRF}$

2.2.1 Meta-Learners

In the following, we briefly describe the considered meta-learners. We follow the definition of meta-learners by Künzel et al. (2019) and describe them as methods to estimate the CATE using ML methods that are built for regression or classification tasks only. The S- and T-learner, for example, can use any vanilla ML method to predict the conditional outcome. The prediction models can then be used to estimate the conditional average treatment effects. The second class of methods uses additional information from the propensity score. They contain the DR- and X-learner. Again, the conditional mean functions, as well as the propensity score, can be estimated using a broad range of ML methods. The aforementioned methods are also called transformed outcome methods. The idea is to generate a pseudo-outcome using the estimated nuisance functions in the first step. This can be seen as an approximation of the conditional average treatment effect. The pseudo-outcome, which we show in Table 4.1, is an unbiased estimate of the CATE given that the nuisance parameters are known (e.g. if we would know the true propensity score). In a second step, the pseudo-outcome is mapped on the covariates to get the final estimate and to make predictions on new observations. The reason for prediction is that the observed data after a treatment assignment includes the outcome, covariates, and the treatment assignment variable. If we want to classify new observations, we only observe the covariates. Hence we need a model that maps the covariates on the estimated treatment effect. The mapping also serves as a smoother since it could be the case that some pseudo-outcome values are quite extreme (e.g. if the propensity score estimate is very low or high). The last method that we examine in this category is the R-learner. However, it does not generate a pseudo-outcome in the classical sense as it needs algorithms that can modify the loss function. Still, many ML methods can be used which is why we include the R-learner into the category of meta-learners. Currently, R-packages are available for the R-, S-, T-, U-, and X-learner (`install_github("xnie/rlearner")`) and the M-, S-, T-, and X-learner (`install_github("soerenkuenzel/causalToolbox")`). Causal analysis via the potential outcome framework and causal graph theory for Python can be found in Sharma, Kiciman, et al. (2019). For heterogeneous treatment effect analysis via machine learning in Python see EconML (2019). The list of methods above is not a complete list of methods in this subject. For example, we do not talk about methods for instrumental variables, multiple-treatment, difference-and-difference methods, or regression discontinuity designs. We also note that there are other methods to estimate treatment effects in cross-sectional settings. For example, one of the first methods developed to control for confounding bias is the inverse probability weighting (IPW) estimator by Horvitz and Thompson (1952). In Algorithm 9 we show how to implement this method. Some methods use neural networks to estimate heterogeneous treatment effects. See, for example, the recent work by Farrell et al. (2021) who use a deep neural network for semiparametric inference and develop nonasymptotic high probability bounds.

Single- (S-learner) and two-model learner (T-learner):

Let us first start with a very simple and intuitive method, the T-learner. It is a two-step approach where the conditional mean functions $\mu_1(x) = \mathsf{E}[Y^1|X_i = x]$ and $\mu_0(x) = \mathsf{E}[Y^0|X_i = x]$ are estimated separately with any generic machine learning algorithm. The difference between the two functions results in the CATE as shown in Table 4.1 and as seen in equation 2.3. One problem with the T-learner is that it aims to minimize the mean squared error for each separate function rather than the mean squared error of the treatment effect. By splitting the sample into two groups there is only information on one group. This might be problematic if the two functions shrink different covariates which are important in both groups. This is especially the case in an RCT. See, for example, Kennedy

(2020) and Künzel et al. (2019) for settings when the T-learner is not the optimal choice. An alternative is to model only one function and include the treatment assignment into this function. This approach is called the S-learner. See for example, Hill (2011) and Foster et al. (2011) for early examples of proposing the S-learner. Algorithm 7 in the Appendix describes how to implement the S-learner while algorithm 1 shows the implementation for the T-learner.

Algorithm 1: T-learner

```
Input: $Z_i = \{Y_i, D_i, X_i\}_{i \in N}$
Split sample $Z$ into $K$ random subsets
for $k$ in {1, ...,K} do
    assign Sample $S_a = Z \uplus S_k$ and $S_k$
    regress $Y_i^0 = \hat{\mu}_0\left(X_i^0\right) + \hat{U}_i^0$, with $i \in S_a|D = 0$
    regress $Y_i^1 = \hat{\mu}_1\left(X_i^1\right) + \hat{U}_i^1$, with $i \in S_a|D = 1$
            estimate $\hat{Y}_i^0 = \hat{\mu}_0\left(X_i\right)$, with $i \in S_k$
            estimate $\hat{Y}_i^1 = \hat{\mu}_1\left(X_i\right)$, with $i \in S_k$
    create $\hat{\tau}_k(X_i) = \hat{\mu}_1(X_i) - \hat{\mu}_0(X_i)$
combine $\hat{\tau}(X_i) = \{\hat{\tau}_1, \hat{\tau}_k, \ldots, \hat{\tau}_K\}$
```

Doubly-Robust learner (DR-learner):

A more efficient method than the T-learner can be the DR-learner. It builds on the T-learner and adds a version of the inverse probability weighting (IPW) scheme on the residuals of both regression functions $\{Y^d - \hat{\mu}_d(x)\}$. We can think of it as combining two different models and hence avoid drawbacks like the minimization goal from the T-learner and a potentially high variance from an IPW model when some propensity scores are small. The doubly-robust learner takes its name from a double robustness property which states that the estimator remains consistent if either the propensity score model or the conditional outcome model is correctly specified Lunceford and Davidian (2004). The DR-learner creates a pseudo-outcome which is an unbiased estimator for the CATE:

$$\hat{\psi}_{DR} = \hat{\mu}_1(x) - \hat{\mu}_0(x) + \frac{D\left\{Y - \hat{\mu}_1\left(x\right)\right\}}{\hat{e}\left(x\right)} - \frac{\left(1 - D\right)\left\{Y - \hat{\mu}_0\left(x\right)\right\}}{\left(1 - \hat{e}\left(x\right)\right)}. \tag{2.4}$$

Equation 2.17 in the Appendix shows the double-robustness property by rewriting equation 2.4. Whenever the propensity score or the conditional mean function is correctly specified the doubly-robust estimator converges to $\mathsf{E}[Y^1] - \mathsf{E}[Y^0]$.

The first part in equation 2.4 can be seen as a regression adjustment parameter (this is the difference between the two conditional mean functions). The second part, which makes use of the propensity score, can be seen as an inverse probability weighting estimator applied to the residual from the conditional mean functions. The DR-learner is very flexible in the choice of the ML method. Estimating the nuisance parameters we can use any ML method. Since the pseudo-outcome is already an unbiased estimator for the CATE the loss-function in the final regression task is to minimize the mean squared error. This allows using the same range of ML methods as in the first step. Recently, this estimator has gained popularity to estimate the CATE, especially in high-dimensional settings. See, for example, the work by Fan et al. (2020). Most recently, Kennedy (2020) find that for estimating the CATE, the finite-sample error-bound from the DR-learner at most deviates from an oracle error rate by the product of the mean squared error of the propensity score and the conditional mean estimator. As can be seen from equation 2.4, extreme propensity score estimates can lead to large pseudo-outcome estimates. Hence it is necessary to look at the distribution of the propensity score and, if necessary, apply methods to make the overlap assumption more realistic.

Algorithm 2: DR-learner

```
Input  : $Z_i = \{Y_i, D_i, X_i\}_{i \in N}$
Split sample $Z$ into $K$ random subsets
for k in {1, ...,K} do
    assign Sample $S_a = Z \uplus S_k$ and $S_k$
    regress $D_i = \hat{e}(X_i) + \hat{V}_i$, with $i \in S_a$
    regress $Y_i^0 = \hat{\mu}_0\left(X_i^0\right) + \hat{U}_i^0$, with $i \in S_a | D = 0$
    regress $Y_i^1 = \hat{\mu}_1\left(X_i^1\right) + \hat{U}_i^1$, with $i \in S_a | D = 1$
        estimate $\hat{D}_i = \hat{e}(X_i)$, with $i \in S_k$
        estimate $\hat{Y}_i^0 = \hat{\mu}_0(X_i)$, with $i \in S_k$
        estimate $\hat{Y}_i^1 = \hat{\mu}_1(X_i)$, with $i \in S_k$
    create $\hat{\psi}_{DR,k} = \hat{\mu}_1(x) - \hat{\mu}_0(x) + \frac{D\{Y - \hat{\mu}_1(x)\}}{\hat{e}(x)} - \frac{(1-D)\{Y - \hat{\mu}_0(x)\}}{(1 - \hat{e}(x))}$
    store $\hat{\psi}_{DR,k}$ for $i \in S_k$
combine $\hat{\psi}_{DR} = \{\hat{\psi}_{DR,1}, \hat{\psi}_{DR,k}, \dots, \hat{\psi}_{DR,K}\}$
Cross-fitting:
for oob in (1:2) do
    if oob = 1: $S_{oob} = Z_i$ with $i \in \{1, ..., N/2\}$ and $S_{train} = Z_i \uplus S_{oob}$
    if oob = 2: $S_{train} = Z_i$ with $i \in \{1, ..., N/2\}$ and $S_{oob} = Z_i \uplus S_{in}$
    for l in 1:5 do
        split $S_{train}$ in $\{S_1, S_2, \dots, S_5\}$
        regress $\hat{\psi}_i = \hat{t}_{DR}(X_i) + W_i$, for $i \in S_l$
            estimate $\tilde{\tau}_l(X_i) = \hat{t}_{DR}(X_i)$, with $i \in S_{oob}$
    average $\hat{\tau}_{oob}(X_i) = \mathsf{E}[\tilde{\tau}(X_i)]$
row bind $\hat{\tau}(X_i) = \{\hat{\tau}_1, \hat{\tau}_2\}$
```

R-learner:

The R-learner makes use of the idea of orthogonalization to cancel out any selection bias that may arise in observational studies from observed covariates. Here, the residuals from the regression of Y on X are regressed on the residuals from the regression of D on X

and weighted by the squared residuals, $\{D - \hat{e}(x)\}^2$. This is similar to the double machine learning approach from Chernozhukov, Chetverikov, et al. (2018) where their estimator of interest is the ATE. Nie and Wager (2020) develop a general class of two-step algorithms for the estimation of the CATE. The R-learner, as from residualized and as an homage to Peter M. Robinson, makes explicit use of machine learning methods. Achieving Neyman orthogonality using a residuals-on-residuals (or debiasing) approach has a long history in econometrics (see the Frisch–Waugh–Lovell theorem from the 1930s for linear regression) and mainly builds on the work by Robinson (1988) who replaces the linear parts by non-parametric kernel regression. The CATE from the R-learner is obtained by the following minimization task:

$$\begin{aligned}\hat{\tau}(\cdot) = \operatorname{argmin}_{\tau} \Bigg\{ \frac{1}{n} \sum_{i=1}^{n} \Big[\big\{ Y_i - \hat{\mu}^{(-i)}\left(X_i\right) \big\} \\ - \big\{ D_i - \hat{e}^{(-i)}\left(X_i\right) \big\} \tau\left(X_i\right) \Big]^2 + \Lambda_n\{\tau(\cdot)\} \Bigg\}. \end{aligned} \tag{2.5}$$

The superscript $(-i)$ indicates the sample splitting. The conditional mean functions are trained without the i-th observations and evaluated only for i. We will explain certain sample splitting procedures later. The term $\Lambda_n\{\tau(\cdot)\}$ can be interpreted as a regularizer on the complexity of the $\tau(\cdot)$ function. In practice, this regularization term could be explicitly given as in penalized regression or implicitly introduced, e.g., as provided by a carefully designed deep neural network. The main difference to the pseudo-outcome estimators (the DR- and X-learner) is that the R-learner needs to alter the loss-function of the ML method. Even if ψ_R with weights equal to 1 as an estimator for $\tau(x)$ it can suffer from high variance if the nuisance functions are not known. The variance is mainly caused by the propensity score since $\{D - \hat{e}(x)\}$ is in the denominator. This is where the weighting comes into play. Observations that have a high variance are weighted by the squared of $\{D - \hat{e}(x)\}$ and hence are less important. The weights for each observation directly influence the loss function (e.g. in boosting methods they manipulate the gradient). Therefore, applying the R-learner needs ML methods that have the option of altering the loss-function through weighting. The following methods have this option: lasso and ridge regression (`glmnet`), boosting (included in the DMatrix format) (`XGBoost`), neural network (`nnet`) and the random forest (`ranger`). Note that the ranger package seems to be the only implementation of weights for a random forest. The weights are applied on the whole training sample and observations with larger weights will be selected with higher probability in the bootstrap (or subsampled) samples for the trees.

Algorithm 3: R-learner

```
Input: $Z_i = \{Y_i, D_i, X_i\}_{i \in N}$
Split sample $Z$ into $K$ random subsets
for $k$ in {1, ...,K} do
    assign Sample $S_a = Z \uplus S_k$ and $S_m = S_k$
    regress $D_i = \hat{e}\left(X_i\right) + \hat{V}_i$, with $i \in S_a$
    regress $Y_i = \hat{\mu}\left(X_i\right) + \hat{U}_i$, with $i \in S_a$
            estimate $\hat{D}_i = \hat{e}\left(X_i\right)$, with $i \in S_m$
            estimate $\hat{Y}_i = \hat{\mu}\left(X_i\right)$, with $i \in S_m$
    create $\hat{\psi}_R = \frac{(Y_i - \hat{\mu}(X_i)}{(D_i - \hat{e}(X_i))}$ and $w_i = (D_i - \hat{e}(X_i))^2$
    store $\hat{\psi}_{R,k}$ and $w_{i,k}$ for $i \in S_k$
combine $\hat{\psi}_R = \{\hat{\psi}_{R,1}, \hat{\psi}_{R,k}, \dots, \hat{\psi}_{R,K}\}$, $w_R = \{w_1, w_k, \dots, w_K\}$
Cross-fitting:
for oob in (1:2) do
    if oob = 1: $S_{oob} = Z_i$ with $i \in \{1, ..., N/2\}$ and $S_{train} = Z_i \uplus S_{oob}$
    if oob = 2: $S_{train} = Z_i$ with $i \in \{1, ..., N/2\}$ and $S_{oob} = Z_i \uplus S_{in}$
    for l in 1:5 do
        split $S_{train}$ in $\{S_1, S_2, \dots, S_5\}$
        regress $\hat{\psi}_i = \hat{t}_R(X_i) + W_i$ and weight by $w_R$, for $i \in S_l$
                estimate $\tilde{\tau}_l(X_i) = \hat{t}_R(X_i)$, with $i \in S_{oob}$
    average $\hat{\tau}_{oob}(X_i) = \mathsf{E}[\tilde{\tau}(X_i)]$
row bind $\hat{\tau}(X_i) = \{\hat{\tau}_1, \hat{\tau}_2\}$
```

X-learner:

Künzel et al. (2019) propose the X-learner which estimates a treatment effect separately for the control and the treatment group. This might be especially helpful in situations where the proportion of the two groups is highly imbalanced. The X-learner has several steps. The first step is identical to the T-learner, namely estimating the two conditional mean functions. In the second step, we predict the counterfactual outcome using the two functions. If a person is treated and hence the observed outcome is Y^1 we subtract the estimated counterfactual. If a person is in the control group we use the estimated counterfactual outcome and subtract the observed outcome (Y^0). This results in two imputed treatment effects:

$$\hat{\psi}_X^1 \overset{\text{def}}{=} Y^1 - \hat{\mu}_0\left(x^1\right) \quad \text{for} \quad D_i = 1, \tag{2.6}$$

$$\hat{\psi}_X^0 \overset{\text{def}}{=} \hat{\mu}_1\left(x^0\right) - Y^0 \quad \text{for} \quad D_i = 0. \tag{2.7}$$

These imputed effects are now used in a third step to regress them individually on the covariates to obtain $\hat{\tau}_0(x)$ (the CATE for the control group) and $\hat{\tau}_1(x)$ (the CATE for the treatment group). The final estimator combines the two estimators plus some weights, $g(x)$:

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

In a randomized controlled trial, the two estimates should not differ significantly. If there are confounding variables and if the support of the treatment variable given covariates differs among the treatment status we would expect the two estimates to be different. Hence, a natural weighting function for $g(x)$ could be the propensity score. We would use $1 - \hat{e}(x)$ for the treatment group and $\hat{e}(x)$ for the control group estimate, respectively. Algorithm 4 describes the procedure in detail and takes sample-splitting into account.

Algorithm 4: X-learner

```
Input: $Z_i = \{Y_i, D_i, X_i\}_{i \in N}$
Split sample $Z$ into $K$ random subsets
for $k$ in {1, ...,K} do
    assign Sample $S_a = Z \uplus S_k$ and $S_k$
    regress $D_i = \hat{e}(X_i) + \hat{V}_i$, with $i \in S_a$
    regress $Y_i^0 = \hat{\mu}_0\left(X_i^0\right) + \hat{U}_i^0$, with $i \in S_a|D = 0$
    regress $Y_i^1 = \hat{\mu}_1\left(X_i^1\right) + \hat{U}_i^1$, with $i \in S_a|D = 1$
        estimate $\hat{D}_i = \hat{e}(X_i)$, with $i \in S_k$
        estimate $\hat{Y}_i^0 = \hat{\mu}_0(X_i)$, with $i \in S_k$
        estimate $\hat{Y}_i^1 = \hat{\mu}_1(X_i)$, with $i \in S_k$
    create $\hat{\psi}_X^1 \overset{\text{def}}{=} Y^1 - \hat{\mu}_0\left(x^1\right)$ for $i \in S_k$
    create $\hat{\psi}_X^0 \overset{\text{def}}{=} \hat{\mu}_1\left(x^0\right) - Y^0)$ for $i \in S_k$
    store $\hat{\psi}_X^1$, $\hat{\psi}_X^0$ and $\hat{e}(X_i)$ for $i \in S_k$
combine $\hat{\psi}_X^1 = \{\hat{\psi}_{X,1}^1, \hat{\psi}_{X,k}^1, \dots, \hat{\psi}_{X,K}^1\}$, $\hat{\psi}_X^0 = \{\hat{\psi}_{X,1}^0, \hat{\psi}_{X,k}^0, \dots, \hat{\psi}_{X,K}^0\}$,
  $\hat{e}(X_i) = \{\hat{e}_1(X_i), \hat{e}_k(X_i), \dots, \hat{e}_K(X_i)\}$
regress $\hat{\psi}_X^1 = \hat{t}^1(X_i) + W_i^1$, with $i \in Z$
regress $\hat{\psi}_X^0 = \hat{t}^0(X_i) + W_i^0$, with $i \in Z$
    estimate $\hat{\tau}_k^1(X_i) = \hat{t}^1(X_i)$, with $i \in Z$
    estimate $\hat{\tau}_k^0(X_i) = \hat{t}^0(X_i)$, with $i \in Z$
average $\hat{\tau}_k(X_i) = \hat{e}(X_i)\hat{\tau}_k^0 + (1 - \hat{e}(X_i))\hat{\tau}_k^1$
```

Summary of meta-learners:

We summarise the considered meta-learners in Table 4.1 where $\hat{\psi}$ states the pseudo-outcome or estimator for each of the learners. The last column counts the number of nuisance functions needed to estimate the pseudo-outcome or estimator. In brackets, we state the total number of models needed to get the final CATE estimate. Note that the X-learner is regressed only for the treated observations and again only for the observations in the control group. This is why we need two more additional models for the final estimate.

The estimators from Table 4.1 can be represented as a weighted minimization problem which solves the following:

Table 2.2: Summary of meta-learners

Method	Estimator/Pseudo-outcome	Weights (w_i)	# of Models
S-learner	$\hat{\psi}_S = \hat{\mu}(x, d = 1) - \hat{\mu}(x, d = 0)$	1	1 (2)
T-learner	$\hat{\psi}_T = \hat{\mu}_1(x) - \hat{\mu}_0(x)$	1	2 (3)
DR-learner	$\hat{\psi}_{DR} = \hat{\psi}_T + \frac{D\left\{Y - \hat{\mu}_1(x)\right\}}{\hat{e}(x)} - \frac{(1-D)\left\{Y - \hat{\mu}_0(x)\right\}}{(1-\hat{e}(x))}$	1	3 (4)
R-learner	$\hat{\psi}_R = \frac{\left\{Y - \hat{\mu}(x)\right\}}{\left\{D - \hat{e}(x)\right\}}$	$\left\{D - \hat{e}(x)\right\}^2$	2 (3)
IPW-learner	$\hat{\psi}_{IPW} = \frac{DY}{\hat{e}(x)} - \frac{(1-D)Y}{(1-\hat{e}(x))}$	1	1 (2)
X-learner	$\hat{\psi}_X^1 \overset{\text{def}}{=} Y^1 - \hat{\mu}_0(x^1)$ $\hat{\psi}_X^0 \overset{\text{def}}{=} \hat{\mu}_1(x^0) - Y^0$	1	3 (5)

Notes: Considered meta-learners that estimate the CATE. # of Models counts the number of nuisance functions to estimate the pseudo-outcome. Numbers in brackets count the total number of models to train to get the final CATE estimate or to make predictions.

$$\min_{\tau}\left\{N^{-1}\sum_{i=1}^{N} w_i \left\{\hat{\psi}_i - \tau(x)\right\}^2\right\}.$$

The choice of ML algorithms for meta-learners:

The accuracy of the CATE estimation depends on the accuracy of the nuisance functions and hence on the choice of the ML method. To minimize the dependence of the ML methods on our estimates, we do not assign specific machine learning methods for the estimation but consider a range of different popular methods. To choose which ML method to use for each nuisance function as well as for any additional functions, we use a stacking method. In such a setting, not only one ML method may be chosen but an ensemble of methods that are stacked together with different weights. We use the SuperLearner package as proposed by Polley et al. (2011). It also enables us to choose different models for each nuisance function and setting. The package offers a general class of prediction methods to be considered by the ensemble. From the 42 different algorithms, we select gradient boosted trees (`xgboost`), neural network (`nnet`) and random forest (`ranger`) for our analysis. Note that the R-learner needs to include weights to minimize the R-loss in the algorithm, so we need to make sure that the ML methods we use have this possibility included. While many researchers include the lasso in their simulations or empirical analysis, we do not use this approach. The reason is that the lasso algorithm would ideally need to assume a parametric form. This means that if we believe that there are interaction and or polynomial effects from X on Y, we would need to include such transformations. There are extensions like the adaptive lasso that expand the feature space by including such additional factors. The computation time does however increase the more features we include. These are the main reasons why we do not include the lasso in our analysis.

We use 10-fold cross-validation to estimate the performance of all machine learning models. Cross-validation is a resampling procedure used to evaluate ML models on a finite data sample. Depending on the ML model, the data can be fit perfectly and hence produce a high variance (overfitting). This is, however, on the training sample and the model can behave poorly on unseen data. Hence, we have to validate our models. We could use a part of the data for validation. Since there is never enough data, removing a part of it poses a potential for underfitting (we might lose trends in the data or important patterns). What we require instead is a method that provides enough data for training the model and also leaves enough data for validation. K-fold cross-validation does exactly that. This approach involves randomly splitting the set of observations into K groups, or folds, of approximately equal size. The model is fit on folds 2 to K while the first fold is used as a validation set. It is also important that any preparation of the data before fitting the model occur on the training sample that is used for cross-validation within the loop rather than on the broader data sample. This also applies to any hyperparameter tuning, for example, the number of trees, the minimum observations within a node, learning rates, or shrinkage parameters. There is no formal rule for the choice of K but usually, it is set to 5 or 10. These values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance. The reason is the following: The larger K, the smaller the difference in size between the (original) training set and the resampling subset ($K - 1$ folds). As this difference decreases, the bias of the technique becomes smaller. This means that the bias is smaller for $K = 10$ than for $K = 5$. A special case of cross-validation is the leave-one-out cross-validation (LOOCV). In this case, K is set to the sample size and only one observation is the validation sample. In all procedures, the I resampled estimates of performance are summarized (e.g. by the mean and the standard error).

Since we apply multiple models to estimate the nuisance functions, we create a weighted average among all models. Using stacking, we can find the optimal combination of a collection of prediction algorithms or even different settings within one model. In other words, we build a linear model that uses the outcome variable of the validation set as the dependent variable and all different base learners as the input variables. For the random forest, we set the following tuning parameters: `n.trees=1000, min.node.size=10`.

2.2.2 Modified ML Methods

We now describe some methods that modify existing ML methods to estimate the CATE directly. In contrast to meta-learners that are flexible in the choice of the ML algorithm, these methods use a specific ML method (mostly tree-based algorithms). Packages or code in R are available for the causal forest (`grf`), the causal Boosting (`https://github.com/saberpowers/causalLearning`) and the causal BART (`install _github("vdorie/bartCause")`. Since causal boosting is computationally expensive, we do not consider this method in our analysis.

Causal Forest:

The causal forest method, part of the generalized random forest (GRF) by Athey et al. (2019) builds on a random forest algorithm to find neighborhoods in the covariate space. These neighborhoods are built by recursive splitting the covariates into subgroups while the criterion to do so is based on heterogeneity in treatment effects. The idea is to find leaves where the treatment effect is constant but different from other leaves. If we know that $\tau(x)$ were constant over some neighbourhood $N(x)$, we could solve a partially linear model over $N(x)$ using the residual-on-residual approach (see e.g. Robinson (1988)): First, we estimate $e(x) = \mathsf{E}[D_i|X_i = x]$ and second, $\mu(x) = \mathsf{E}[Y_i|X_i = x]$. We can use any non-parametric method like the lasso, random forests, boosting methods, neural networks and others. The final step is to estimate $\tau(x)$ over the neighbourhood $N(x)$:

$$\hat{\tau}(x) = \frac{\sum_{\{i:X_i \in \mathcal{N}(x)\}} \{Y_i - \hat{\mu}(X_i)\} \{D_i - \hat{e}(X_i)\}}{\sum_{\{i:X_i \in \mathcal{N}(x)\}} \{D_i - \hat{e}(X_i)\}^2}. \tag{2.8}$$

Note that this approach looks similar to the R-learner. Chernozhukov, Chetverikov, et al. (2018) showed that when using any of the aforementioned ML methods for the estimation of the nuisance functions and then use the residual-on-residual approach to estimate the average treatment effect the following regularity condition holds:

Given that,

$$\mathsf{E}\left[\{\mu(X_i) - \hat{\mu}(X_i)\}^2\right]^{\frac{1}{2}} \ll \frac{1}{n^{1/4}}, \quad \mathsf{E}\left[\{e(X_i) - \hat{e}(X_i)\}^2\right]^{\frac{1}{2}} \ll \frac{1}{n^{1/4}}, \tag{2.9}$$

we get a central limit theorem such that $\sqrt{n}(\hat{\tau} - \tau) \Rightarrow \mathcal{N}(0, V)$. The treatment effect in the above setting, however, has to be constant. We can assume that with heterogeneous treatment effects, there are subgroups such that the constant effect assumption holds. The question of how to find such accurate subgroups is exactly where the (causal) random forest comes into play. To create leaves that consist of observations with the same (average) treatment effect, the splitting criterion has to rely on maximizing the heterogeneity in treatment effects between leaves (similar to maximizing the variance between the leaves). Here we use again the method from equation (2.8). In observational studies where self-selection into treatment is present, the first splits might not be a good representation of the treatment effect rather than differences due to confounding variables. To overcome this problem, Athey et al. (2019) suggest applying local-centering. This means that we use the residuals of the outcome and treatment variable as data instead of the original values. Therefore one has to train two nuisance functions beforehand to predict the conditional mean which is used to create the residuals. While machine learning methods rely on sample splitting to avoid overfitting, the causal random forest integrates this via an honesty condition. A tree is honest if, for each training sample i, it only uses the response Y_i to estimate the within-leaf treatment effect or to decide where to place the split, but not both.

So far we have looked at how a single tree is build and how the final treatment effect can be estimated. To extend this procedure to multiple trees, let us view a forest as a weighting function:

$$\hat{\mu}(x) = B^{-1}\sum_{b=1}^{B}\sum_{i=1}^{n} Y_i \frac{1\{X_i \in L_b(x)\}}{|L_b(x)|} = \sum_{i=1}^{n} Y_i \underbrace{B^{-1}\sum_{b=1}^{B}\frac{1\{X_i \in L_b(x)\}}{|L_b(x)|}}_{\alpha_i(x)}. \tag{2.10}$$

Instead of seeing a forest as a double average over observations within a leaf and B single trees, we can integrate the first sum to be a weighted average over all X_i that fall into the leaf $L_b(x)$ and divide by the total number of observations within the leaf ($|L_b(x)|$). This weighted average tells us how often Y_i falls into a certain leaf and hence the weight that we have to apply to control for the different proportions. The weights can be represented as $\alpha_i(x)$. We can now use these weights to weigh each observation in a generalized method of moments estimator where we apply a linear model, regressing the residuals of D_i on the residuals of Y_i and weigh by α_i. This is how we get the CATE using a random forest. The algorithm is implemented in the `grf` package. See Friedberg et al. (2018) for an extension of this approach to local linear forests. Algorithm 5 describes the approach to estimating the CATE for each observation using the causal forest. The results from steps 4-7 are used for local-centering, as described above. If not provided in the causal forest (step 8), the nuisance functions are estimated internally. We use the `regression_forest` function to estimate the nuisance parameters. This function uses the honest estimation which means the prediction is based on out-of-bag observations. We state the estimation explicitly since it might be the case that a different ML method is better suited in predicting the conditional mean or the propensity score (e.g a boosting method). When using different methods, we just need to make sure that the predictions (steps 6 and 7) are again based on either out-of-bag observations or a different subsample. Theoretically, if relying completely on the causal forest we do not need to split the sample at all since the honest condition applies to each step (the nuisance parameters and the estimation of the CATE). Since we use K fold sample splitting for all other methods we apply the same subsamples when using the causal forest.

Algorithm 5: Causal Forest

```
   Input: $Z_i = \{Y_i, D_i, X_i\}_{i \in N}$
1  Split sample $Z$ into $K$ random subsets
2  for $k$ in $\{1, \ldots, K\}$ do
3      assign Sample $S_a = Z \dot{\cup} S_k$ and $S_k$
4      regress $D_i = \hat{e}\,(X_i) + \hat{V}_i$, with $i \in S_a$
5      regress $Y_i = \hat{\mu}\,(X_i) + \hat{U}_i$, with $i \in S_a$
6              estimate $\hat{D}_i = \hat{e}\,(X_i)$, with $i \notin L_b$
7              estimate $\hat{Y}_i = \hat{\mu}\,(X_i)$, with $i \notin L_b$
8      apply Causal Forest using $Y_i, D_i, X_i, \hat{\mu}(X_i), \hat{e}(X_i)$ with $i \in S_a$
9              estimate $\hat{\tau}_k(X_i)$ with $i \in S_k$
10 combine $\hat{\tau}(X_i) = \{\hat{\tau}_1, \hat{\tau}_k, \ldots, \hat{\tau}_K\}$
```

Causal Boosting:

An alternative to random forest based causal inference is given by Powers et al. (2018) who introduces boosted trees and causal multivariate adaptive regression splines (MARS). By iteratively fitting weak learners to the residuals of a model, an approximation of the function is build. The idea is to fit a causal tree in the style of Wager and Athey (2018) while setting the basis function $\hat{G}(x, D)$ to zero. Now we estimate the residuals by $Y_i - \varepsilon \times \hat{g}_k(X_i, D_i)$ and update $\hat{G}_k = \hat{G}_{k-1} + \varepsilon \times \hat{g}_k$. k defines the terminal nodes from the tree and ε is the learning rate parameter. After K iterations we return $\hat{G}_K(x, D)$. Estimating the CATE is done by setting D to 1 for the treated observations and 0 for the control-group observations, such that:

$$\hat{\tau}(x) = \hat{G}_K(x, 1) - \hat{G}_K(x, 0). \tag{2.11}$$

Like in the causal forest the problem remains how to control for overfitting. Especially boosting methods are prone to overfit the data since the trees are not built independently. While a random forest would benefit from using more trees over which to average, in gradient boosting the number of trees is an important tuning parameter that needs to be controlled. In supervised ML we would ideally apply cross-validation. In our case, the parameter of interest is the CATE and we do not observe the true value for each observation. Hence, cross-validation does not apply here. Instead, we can do something like the honest approach from the causal forest.

Powers et al. (2018) propose to split the data into two distinct sets. The training set is used to build the causal boosting. Using the split-points and split-variables from the training set we use the covariates from the validation set, lets call it X_v, for validation and get new estimates based on D_v and Y_v for each terminal node. This procedure is done for any of the K trees, using again the residuals (this time from the validation set tree) to reestimate the terminal nodes of the next causal tree. This allows estimating a validation error for each of the original K models. The overall validation error for a causal boosting model is given by the differences of the CATE from the original vs. the validation trees.

Causal BART:

While (causal) boosting relies on multiplying each sequential tree by the learning rate (ε), the idea developed by Chipman et al. (2010) is to estimate a posterior distribution of the prediction by explicitly setting priors for the trees and ensemble structure (e.g. the depth of the tree, the probability of a new split). Using a Bayesian approach allows for a broader set of information than the point estimate from regression and classification methods. The Bayesian Additive Regression Trees (BART) approach is a combination of three methods: Using gradient boosting trees, a Bayesian framing for each individual tree, and Markov chain Monte Carlo (MCMC) sampling to do backfitting (using additive and generalized additive models for posterior sampling). Hill (2011) proposes to use such nonparametric Bayesian models to estimate treatment effects. Given strong ignorability, one way to estimate treatment effect is to estimate the response function $\mu(X_i, D_i)$. This function is estimated in one step instead of estimating two functions. Hence, the prior is set directly for the response surface. This approach is also called the S-learner - train one function and set D_i to 1 and 0 for each observation to get estimates for both potential outcomes. Hahn et al. (2020) extends the idea of using a Bayesian approach to estimate treatment effects but expresses the response surface as:

$$\mathsf{E}\left[Y_i \mid X_i, D_i = d\right] = \mu\left\{X_i, \hat{e}\left(X_i\right)\right\} + \tau\left(X_i\right) D_i, \tag{2.12}$$

where $\hat{e}(x)$ is the estimated propensity score and the functions $\mu(\cdot)$ and $\tau(\cdot)$ are independent BART priors. The inclusion of the estimated propensity score can be seen as a covariate-dependent prior to control for confounding bias. The method is specially designed to estimate the CATE from observational studies with small effect sizes and heterogeneous effects. The package we use is built on the model by Hill (2011) (`install_github("vdorie/bartCause")`). A package that implements the method proposed by Hahn et al. (2020) is in development (`install_github("socket778/XBCF")`. This package is also available for Python. Note that the causal BART produces credible intervals as a contrast to confidence intervals. They are estimated from the posterior probability function and hence rely on the prior distribution while confidence intervals are based on data only. We will only use the term confidence interval on all methods, however, we do mean credible intervals for the causal BART and (frequentists) confidence intervals for all other methods.

Algorithm 6: Causal BART

```
Input: $Z_i = \{Y_i, D_i, X_i\}_{i \in N}$
Split sample $Z$ into $K$ random subsets
for $k$ in {1, ...,K} do
    apply Causal BART using $Y_i, D_i, X_i$ with $i \in S_a$
        estimate $\hat{\tau}_k(X_i)$ with $i \in S_k$
combine $\hat{\tau}(X_i) = \{\hat{\tau}_1, \hat{\tau}_k, \dots, \hat{\tau}_K\}$
```

2.2.3 Sample splitting and cross-fitting

To aim for a consistent estimator, we need to assume certain complexity conditions on the nuisance functions. Specifically, we want them to be smooth (i.e. differentiable) and the entropy of the candidate nuisance functions to be small enough to fulfill Donsker conditions (e.g. if we assume Lipschitz parametric functions or VC classes). In high-dimensional settings (p>n) or when using ML methods that are complex or adaptive, the Donsker conditions might not hold; see, for example, Robins et al. (2013), Chernozhukov et al. (2016) and Rotnitzky et al. (2017). As Chernozhukov, Chetverikov, et al. (2018) noticed, verification of the entropy condition is so far only available for certain classes of machine learning methods, such as lasso and post-lasso. For classes that employ cross-validation or for hybrid methods (like the SuperLearner), it is likely difficult to verify such conditions. Luckily, there is an easy solution available: sample splitting. When splitting the sample, we can use independent sets for estimating the nuisance functions and constructing the treatment estimation equation. By using different sets, we can treat the nuisance functions as fixed functions which allow avoiding conditions on the complexity. It also allows us to use any ML method such as random forest or boosting or even an ensemble of different methods. The split-sample approach to avoid smoothness conditions dates back at least to Bickel (1982) and was extended to also use cross-fitting by Schick (1986).

To overcome a potential loss in efficiency, since only a subset of the data is used when estimating the CATE, cross-fitting is an increasingly popular approach to combine ML methods with semi-parametric estimation problems; see, for example, Chernozhukov, Chetverikov, et al. (2018), Newey and Robins (2018) and Athey and Wager (2017). We note that there are two definitions of cross-fitting. First, it is defined in the context of estimating the CATE for all observations. For example, we split the data into two folds, subset A and M. We use fold A to train the nuisance functions and then estimate the parameter of interest using subset M. Now we switch the roles of the sets, using subset M for training and subset A for estimation. As a result, we get estimates of the CATE for all observations.

The second definition is more in the spirit of averaging CATE estimates obtained from different partitions that are used for the nuisance parameter estimation. For example, let us say we have again the data as above but also an independent test set. Now we can use the procedure as before. First, we train the nuisance function on subset A and predict on subset B to get the pseudo-outcomes. We again train a regression function based on B but predict the CATE using the test set. Now we reverse the roles of A and B and get a second prediction of the CATE for the test set. The two results are now averaged to get the final estimate. In this tutorial, we will combine the two definitions of cross-fitting. First, we estimate the CATE on all observations through reversing roles of samples. Second, we use cross-fitting as an averaging tool over K folds. When referring to cross-fitting we mainly mean the latter definition.

We give an example of the benefit from cross-fitting in Figure 4.1. We show the MSE from the true treatment effect for a single estimator and the cross-fit estimator based on a 50:50 sample split. We used the R-learner as the meta-learner and create 50 Monte Carlo replications of the data using the same data generating process (DGP) which simulates a RCT and has the following properties: $N = 2000$, $X = \mathbb{R}^{10}$, $e(X) = 0.5$, and $\tau(x) = X_1 + \mathbb{1}(X_2 > 0) + W \quad \text{with} \quad W \sim \mathcal{N}(0, 0.5)$. Using cross-fitting decreases the MSE compared to the single estimator in about 90% of the cases. We also find that the variance is smaller compared to the single estimator.

In empirical studies we do not have an independent test set and setting aside a partition might not be efficient since we lose observations for the estimation part. In the following, we present an approach to use cross-fitting without an additional test set.

We apply 5-fold sample splitting and use 80% of the full data (denoted by Z) for training the nuisance functions (denoted by S_a) and 20% to for estimation (denoted by S_k). We propose to estimate the nuisance parameters for each of the 5 folds, using all folds but k for training and fold k to predict the conditional mean and the propensity score. We

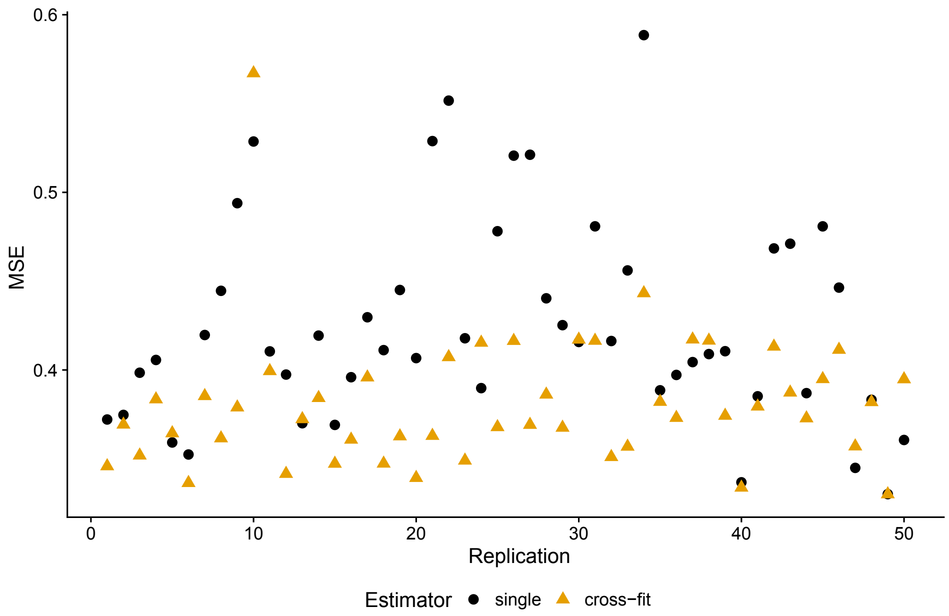


Figure 2.4: Single vs. cross-fit estimation of CATE. Q$_{CF}$

then store the estimates. As a result, we have estimates of all nuisance parameters to create the pseudo-outcomes for each observation obtained from independent samples. Hence, the above-mentioned regularity conditions should be fulfilled. Now we want to train a regression model on the pseudo-outcomes (or minimize the R-loss). Instead of using the full sample for training and prediction, we divide the sample into different parts. We assign half of the sample as the test set (denoted by S_{oob}, which is short for out-of-bag) and the other half that is used to train the regression model. Let us say we want to rely on 5-fold cross-fitting (taking the average of S_{oob} over 5 folds). We therefore split the other half of the sample into 5 folds (denoted by $S_{train} = \{S_1, S_2, \ldots, S_5\}$). Using each fold to train the regression model and predict on S_{oob} leads to 5 estimates that we average by taking the mean. Now we reverse the role of S_{train} and S_{oob} and proceed as above. We apply this procedure to the DR- and R-learner. The S- and T-learner only needs one estimation step and hence it suffices to only use two different samples (the S_a and S_k). In all other methods, we need an additional model (for example, the IPW-learner would also benefit from cross-fitting). The X-learner is quite robust even without cross-fitting. This might be because it only uses the propensity score in the last step. The advantage of the two-step sample splitting approach is that we have more observations to train the nuisance functions (in this example we have $S_a = 0.8Z$ observations instead of $0.8(Z - S_{oob})$). Figure 2.5 shows the procedure in detail. As above, we denote S_k as the fold which is used to estimate the nuisance parameters (e.g. the propensity score, the pseudo-outcomes). The estimators $\tilde{\tau}(x)$ refer to the CATE estimates obtained using S_{oob} given different folds for training. For example,$\tilde{\tau}_1(x)$ first uses S_1 for training and S_{oob} for estimation. To get estimates on the other half of the data (also denoted as S_{oob}) we use S_6 for training. Hence, $\tilde{\tau}_1(x)$ are estimates of the CATE for the whole dataset Z. A more detailed version of the cross-fitting part is shown in Figure 2.15 in the Appendix.

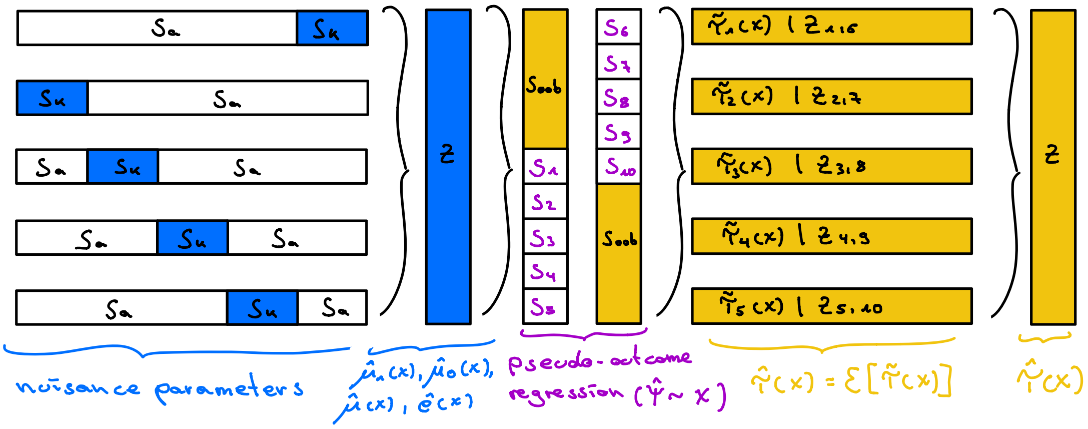

S_a
S_k
S_k
S_a
S_a
S_k
S_a
S_a
S_k
S_a
S_a
S_k
S_a
Z
S_{oob}
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8
S_9
S_{10}
S_{oob}
$\tilde{\tau}_1(x) \mid Z_{1,6}$
$\tilde{\tau}_2(x) \mid Z_{2,7}$
$\tilde{\tau}_3(x) \mid Z_{3,8}$
$\tilde{\tau}_4(x) \mid Z_{4,9}$
$\tilde{\tau}_5(x) \mid Z_{5,10}$
Z
nuisance parameters
$\hat{\mu}_1(x), \hat{\mu}_0(x)$, $\hat{\mu}(x)$, $\hat{e}(x)$
pseudo-outcome regression $(\hat{\psi} \sim X)$
$\hat{\tau}(x) = \mathcal{E}[\tilde{\tau}(x)]$
$\hat{\tau}(x)$

Figure 2.5: Two-step sample splitting procedure.

There might be alternative ways and averaging procedures to ensure robust estimates and prediction results. For example, we could repeat the whole procedure M times and generate different folds in the first place (the S_k folds). The result would be M estimates for each observation of $\hat{\tau}(x)$ over which we could take the median. This might lead to more robust estimates since it takes the sample splitting uncertainty into account. See Jacob (2020) for a Monte Carlo study about the implications of different sample-spitting, cross-fitting, and averaging approaches for meta-learner methods. The simulation study finds that the 5 fold cross-fitting with median averaging procedure works best. Our approach mimics this procedure but changes the way to define a test set on which the cross-fitting is applied. For the meta-learners, we have to do sample splitting and cross-fitting manually while the causal forest as well as the causal boosting relies on honest estimation and does sample splitting by default. Cross-fitting, as we define it, is not implemented in any of the modified ML methods.

2.3 Empirical Examples

To illustrate the methods presented in the previous sections, we consider two empirical examples. In the first example, we examine the effect that microcredits have on the total amount of loans, resulting from a randomized experiment in Morocco. In the second example, we study the effect of 401(k) eligibility on accumulated assets. This example deviates from random treatment assignment and contains self-selection into a treatment. While all presented methods condition on observed pre-treatment variables to estimate heterogeneity in treatment effects they should also be able to control for confounding variables. However, methods that use the propensity score should be more suited to eliminate the selection bias. For each method, we estimate the CATE and provide confidence intervals. We also show how to link the CATE to observed covariates for further analysis. In both examples, we apply the two-step sample splitting with a cross-fitting approach for the DR- and R-learner.

2.3.1 Effect of microcredits on borrowing

We start with an empirical dataset to analyze the effect of microcredit availability on borrowing activities such as the amount of loans (see Crépon et al. (2015) for a recent study using this dataset). Looking beyond the ATE and finding heterogeneous treatment effects is important to target specific groups and to make better policies. The allocation of treatment was randomized between 162 villages in Morocco. The villages were divided into pairs with similar observable characteristics. Then the treatment was randomly assigned to one of the pair while the counterpart was assigned to the control group. Treatment as microcredit availability in this context means that between 2006 and 2007 a microfinance institution started operating only in the treated villages. In 2009, 5,551 households were surveyed in a follow-up study. We use the results from this survey to estimate conditional average treatment effects using different methods and also show some strategies to get some insight into which characteristics are responsible for heterogeneity in treatment effects. We select the following pre-treatment covariates that are observed characteristics for each household such as the age of household's head, number of adults, number of children, total number of members in a household, indicators for households doing animal husbandry, other non-agricultural activity, household spouse responding to the survey, the education of the head and having an outstanding loan over the last year. Table 2.3 shows the mean value for some covariates. They are categorized by all observations, the treated and the control group. Given these unconditional means, we see that the amount of loans for the treatment group is much higher $(2,930)$ than for the control group $(1,802)$. We also see that the mean of the characteristics is quite balanced across the two groups. This reassures us that the treatment assignment was randomly selected and that there are no confounding variables that lead to self-selection into treatment. While there are small differences in some covariates, this is not concerning since all methods that we apply make use of the propensity score or condition on the covariates to estimate the treatment effect only on similar subgroups. For example, more people in the treatment group already have a loan in the last twelve months. We can estimate the probability of being in the treatment group given this variable and reweigh the treatment and control group to adjust for these differences. The dataset and R-code for the microcredit analysis can be found here Q$_{emp}$.

We use three different ML algorithms to estimate the nuisance functions and to map the covariates on the pseudo-outcome (for the DR- and X-learner) or to minimize the R-loss function. Table 2.4 shows the coefficients for each ML algorithm obtained through

Table 2.3: Descriptive statistics of households (mean)

	All	Treated	Control
Outcome Variable			
Total Amount of Loans	2,359	2,930	1,802
Baseline Covariates			
Number of Household Members	3.879	3.872	3.886
Number of Children	1.266	1.261	1.272
Head Age	35.976	35.937	36.014
Declared Animal Husbandry Self-employment Activity	0.415	0.426	0.404
Declared Non-agricultural Self-employment Activity	0.146	0.129	0.164
Borrowed from Any Source	0.210	0.224	0.196
Spouse of Head Responded to Self-employment Section	0.067	0.074	0.061
Member Responded to Self-employment Section	0.044	0.048	0.041

cross-validation in the SuperLearner. The loss-function is the non-negative least squares based on the Lawson-Hanson algorithm which works for both Gaussian and binomial outcomes. We find that the neural network gets the highest weight for all functions except for the propensity score where the random forest has a slightly higher weight. Based on these weights, we only use the neural network and the random forest for the construction of the bootstrapped confidence intervals. This is mainly since we believe that even with a bootstrapped sample, the weights of the algorithms for each function will not change dramatically. Excluding the boosting algorithms decreases the computation time by about 50%.

Table 2.4: Weights of ML methods.

	$\hat{e}(X)$	$\hat{\mu}_0(X)$	$\hat{\mu}_1(X)$	$\hat{\mu}(X)$	DR	R
Boosting	0.00	0.00	0.00	0.00	0.00	0.11
Neural Network	0.46	0.73	0.70	0.80	0.90	0.89
Random Forest	0.54	0.27	0.30	0.20	0.09	0.00

Table 2.5: CATE results for different methods.

Category	Method	20% Least	ATE	20% Most
Meta-Learner	DR-learner	-15.4	1,119.8	3,057.6
	R-learner	84.9	1,081.1	2,237.5
	T-learner	198.4	1,152.5	2,470.3
	X-learner	869.9	1,137.2	1,379.7
Modified ML Methods	Causal BART	593.8	1,132.3	2,304.7
	Causal Forest	296.6	1,129.6	2,329.6

Table 2.5 shows a summary for the heterogeneous treatment, namely the effect for the 20% least affected, the ATE, and the effect for the 20% most affected observations. Especially for the quantiles, we find differences in the estimates given the methods that we consider. This holds for the lower 20% where the effect ranges from -15 to 869 as well as for the 20% most affected with the lowest treatment effect from the X-learner with $1,379$ and the highest estimate from the DR-learner with $3,057$. The high value in the upper quantile from the DR-learner is because it predicts more extreme values at the tail of the distribution. The DR-learner also has the highest variance in terms of treatment effect with a range in number of loans from -15 to $3,057$ on average for the specific quantiles. The ATE is around $1,100$ for all methods and there is no large difference between them. Figure 2.6 shows the treatment effect for each observation, sorted by the size of the effect. We also show 95% confidence intervals (CI). They are estimated via bootstrapping with $B = 500$ replications. Here we adopt the procedure for the construction of CI's from Künzel et al. (2019). We first split our entire dataset into a training and validation set. We use the training set for bootstrapping by creating a sample from the training data of the same size with replacement. For each meta-learner and each bootstrap sample from the training data, we use the test set to estimate the CATE. We repeat this procedure $K = 5$ times looping through all k subsets and define them as the test set. In total, we end up having B estimates for each observation on the whole data. Now we calculate the standard deviation ($\hat{\sigma}$) for each observation which we use to generate a lower and upper bound around the CATE estimates $[\hat{\tau}(x) - q_{\alpha/2}\hat{\sigma}; \hat{\tau}(x) + q_{1-\alpha/2}\hat{\sigma}]$. Especially for the meta-learners, we have a high variance between the bootstrapped samples indicating that even if the CATE is different, there might not be a significant heterogeneity. This is also in line with the estimates from the causal BART and the causal forest that show tighter bounds but also an almost flat CATE curve. To calculate the CATE, denoted by $\hat{\tau}(x)$, we use the whole training data, not a bootstrapped version, and proceed as described in the pseudo-code for the specific meta-learner.

Figure 2.6 shows quite similar values for at least four of the methods. Only the DR- and R-learner have heavier tails for higher CATE estimates. The T-learner has the widest confidence intervals while all other methods show a similar range. Treatment effects based on the DR- and R-learner are heterogeneous, at least for the 20% least and most affected. The most homogeneous prediction comes from the causal BART and the X-learner. Their estimates of the CATE are quite similar. The causal forest also shows an increasing slope of the point estimates but wide confidence intervals for the most affected observations. Based on these results, there is no clear evidence of treatment effect heterogeneity.

In Figure 2.6, we sorted the treatment effect by its size for each method. This does not necessarily mean that all methods have the same order. To look into the order of the CATE based on each method, we show the correlation of the CATE among them in Figure 2.16.

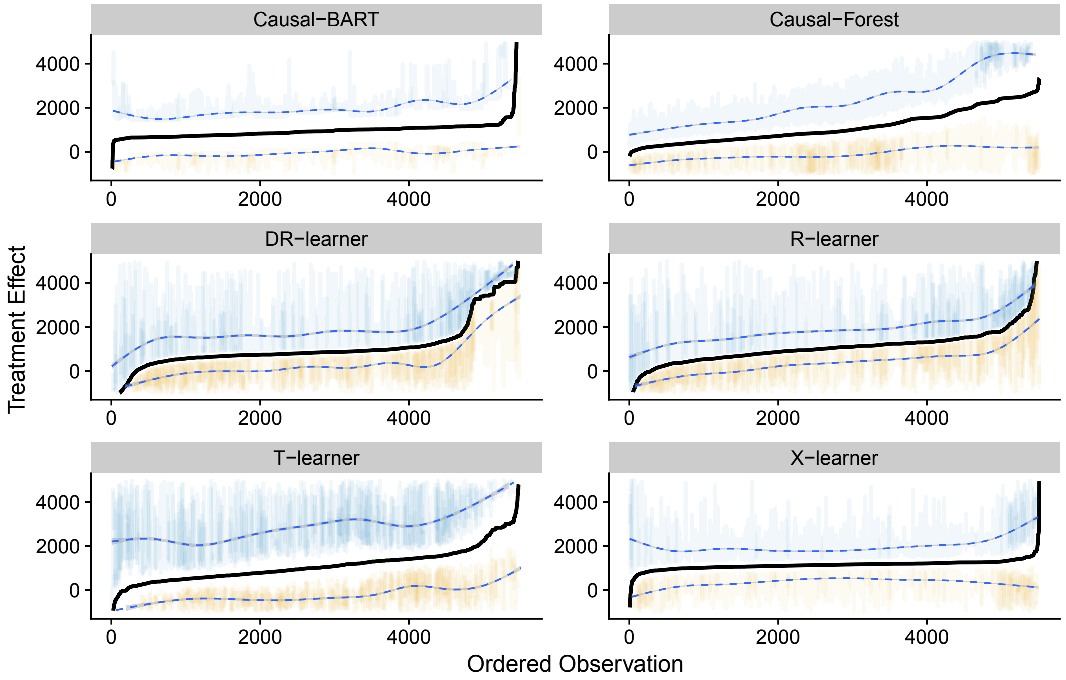


Figure 2.6: Microcredit: Observations sorted by level of treatment effect.Q$_{sort}$

We show the Bravais-Pearson correlation coefficient (ρ), a histogram of the CATE, and correlation ellipses. It is reassuring that all methods are positively correlated. The highest correlation is between the doubly-robust and the R-learner ($\rho = 0.57$) as well as between the T- and X-learner ($\rho = 0.5$). The smallest correlation appears between the causal BART and the R-learner with a correlation coefficient of 0.15.

If we believe that there is at least some difference in the effect between the least and most affected observations, then we can look at the average characteristics of these groups to understand what are potential drivers for the heterogeneity. Here we adopt a simple approach introduced by Chernozhukov, Demirer, et al. (2018), namely the Classification Analysis. The idea is to regress the least and most affected groups on some pre-chosen characteristics ($g(X)$) with G_q being the observations given a specific group of the treatment effect:

$$\delta_{least} = \mathsf{E}[g(X)|G_{least}] \quad \text{and} \quad \delta_{most} = \mathsf{E}[g(X)|G_{most}].$$

Here we focus on the head age, the probability of being self-employed in a non-agricultural sector, and whether someone had an outstanding loan over the past 12 months (borrowed from any source). In Table 2.6 we estimate the average of the characteristics for the two groups as well as if there is a significant difference between the groups. We show results for two methods, the doubly-robust meta-learner (DR-learner) and the causal forest. Detailed

Table 2.6: Classification results for DR-learner and causal forest.

	DR-learner			Causal Forest		
	Most Affected	Least Affected	Difference	Most Affected	Least Affected	Difference
Head age	19.19	46.92	-27.73	10.25	46.80	-36.55
	(17.81,20.58)	(45.53,48.30)	(9.271,11.22)	(45.82,47.77)	(-37.93,-35.17)	(-55.00,-51.49)
	-	-	[0.000]	-	-	[0.000]
Non-agricultural	0.118	0.186	-0.068	0.073	0.136	-0.064
self-employed	(0.097,0.139)	(0.165,0.207)	(0.055,0.091)	(0.118,0.154)	(-0.089,-0.038)	(0.121,0.232)
	-	-	[0.000]	-	-	[0.000]
Borrowed from	0.138	0.338	-0.201	0.050	0.388	-0.338
any source	(0.113,0.162)	(0.314,0.363)	(0.028,0.072)	(0.366,0.411)	(-0.370,-0.307)	(-0.351,-0.219)
	-	-	[0.000]	-	-	[0.000]

Notes: 90% confidence interval in parenthesis and p-values in brackets.

CLAN results that include all methods can be seen in the Appendix (2.12). For both methods, we find a significant difference in head age, probability of being self-employed in a non-agricultural sector, and probability of having a loan. The most affected people seem to be younger. Low values in the head age can arise since many people did not respond to that question and got a value of zero. However, we believe that the non-respondents are missing at random. This allows us to interpret the difference between the two. Looking at employment, we find diverse effects. Interpreting results from the DR-, X-learner, and causal forest we find that people who benefit most from microcredits are those who do not work in the non-agricultural sector. Results from the R-, T-learner and the causal BART suggest the other way around. We note that the result from the T-learner is not statistically significant. The absolute magnitude in the probability difference is rather small which is why we do not interpret this variable as a driver for treatment effect heterogeneity. We also find that people who already have a loan (with a higher probability) are less affected by microcredit availability. There are other possibilities to investigate which covariates might be drivers for effect heterogeneity. For example, if a tree-based method should be the best method for mapping the covariates on the treatment effects then we could use variable importance plots to see which variables (at a certain split in a tree) increase the variance between two leaves. If a variable is (randomly) chosen for a split and the mean values in the two resulting nodes are quite the same as before the split then this variable might not be very useful to explain the heterogeneity. We can also apply partial dependence plots to see how the treatment effect changes if we change one variable.

2.3.2 Effect of 401(k) eligibility on accumulated assets

While the microcredit data is based on a randomized controlled trial, the eligibility of a 401(k) pension plan is not. Only some firms offer access to a 401(k) and hence there is self-selection into treatment. It might be the case that more educated people chose firms that provide a 401(k) pension plan and that they have higher financial assets in the first place. Poterba et al. (1992) argue that conditioning on observed characteristics, like the income, can restore the random assignment mechanism. The dataset we use is the same as in Chernozhukov and Hansen (2004) which is based on the 1991 Survey of Income and Program Participation. We are interested in the question if 401(k) eligibility, our treatment variable, has an impact on accumulated assets (here we use the net financial assets as the outcome variable). We control for income and other variables related to the job choice that may have an impact on treatment assignment and assets. In total, we have 9,915 observations and 13 covariates consisting of age, family size, income, years of education, and indicator variables for married, two-earner status, defined benefit pension status, homeownership, and IRA participation. We split the dataset into 5 parts and proceed as described in the sample splitting section (2.2.3). The dataset and R-code for the 401(k) analysis can be found here Q$_{emp}$.

Table 2.7: Descriptive statistics of observations (mean)

	All	Treated	Control
Outcome Variable			
Net financial assets	18,052	30,347	10,788
Baseline Covariates			
Age	41.06	41.48	40.81
Income	37,201	46,862	31,494
Years of education	13.21	13.76	12.88
Proportion of being married	0.60	0.67	0.56
Proportion of two-earners	0.38	0.48	0.31
Proportion of home-ownership	0.63	0.74	0.57

Table 2.7 shows the mean values for the net financial assets and for some pre-treatment covariates. The amount of assets is higher in the treatment group than in the control group. Concerning the self-selection into treatment, we see that some characteristics are different between the treatment and control group. For example, the proportion of home-ownership, years of education, and income is higher for treated people. There are further reasons to believe that such characteristics are positively correlated with financial assets. In this case, we have to control for such variables to account for the self-selection into treatment. Table 2.8 shows the weights of the considered ML methods from the ensemble method. The preferred method for all nuisance functions is the random forest algorithm.

Table 2.8: Weights of ML methods.

	$\hat{e}(X)$	$\hat{\mu}_0(X)$	$\hat{\mu}_1(X)$	$\hat{\mu}(X)$	DR	R
Boosting	0.36	0.08	0.05	0.08	0.00	0.00
Neural Network	0.14	0.09	0.06	0.06	0.49	0.33
Random Forest	0.50	0.82	0.89	0.85	0.51	0.67

Table 2.9: CATE results for different methods.

Category	Method	20% Least	ATE	20% Most
Meta-Learner	DR-learner	4,998	7,120	9,806
	R-learner	4,250	7,410	11,320
	T-learner	-4,171	7,579	25,326
	X-learner	-285	7,631	18,648
Modified ML Methods	Causal BART	2,466	9,055	21,525
	Causal Forest	5,210	8,228	12,360

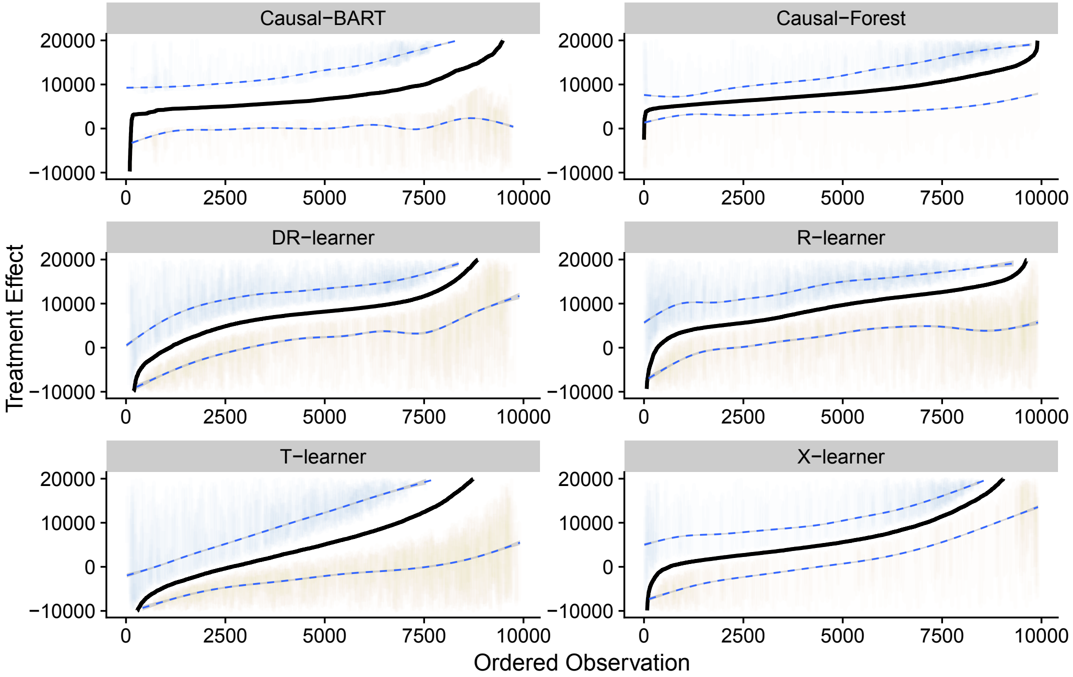


Figure 2.7: 401k: Observations sorted by level of treatment effect.Q$_{sort}$

Table 2.9 shows the estimated CATE for the 20% least affected and 80% most affected as well as the ATE. The ATE is positive and ranges from 7,120 to 9,055, depending on the method. Its variance between the methods is quite low, compared to the estimates for the least and most affected groups. While the T- and X-learner predict a negative effect from 401(k) eligibility on financial assets for the lowest group, all other methods predict a positive effect. The highest affected group has values from 9,806 (from the DR-learner) to 25,326 (from the T-learner). The causal forest predicts values with the lowest heterogeneity. Except for the causal forest, all other learners predict extreme values in the tails of the CATE. If we would use a majority vote from all the methods to interpret the estimated effects, then it is reassuring that everyone has a positive effect from the 401(k) eligibility as can be seen in Figure 2.7. Given the wide confidence intervals, the evidence of treatment effect heterogeneity is not so clear.

Figure 2.17 shows the correlation of the CATE between the different methods. We find that the methods are highly correlated with each other. The lowest correlation is between the DR- and T-learner with a correlation coefficient of $\rho = 0.64$ while the highest correlation is between the causal BART and causal forest ($\rho = 0.85$). The reason why the estimated CATE is more similar might be the large sample size of $N = 9915$.

Since the data does not come from a randomized controlled trial, we expect the distribution of the covariates to be different given treatment status. To see this, we plot the distribution of age, years of education, marital status, income, homeowner status, and two-earner status in Figure 2.8. As we already saw from Table 2.7, treated people have a higher income, slightly more years of education and, among others, are more often homeowners. To see if the estimated propensity score can catch the differences, we can look at a weighted histogram. What we do is weigh the counts in each variable by the inverse of the propensity score. If someone is in the treatment group, we weigh by $1/\hat{e}(x)$ and the control group observations by $1/(1 - \hat{e}(x))$. The result is shown in Figure 2.9. Indeed, we see that the distributions are quite similar after reweighing with the propensity score.

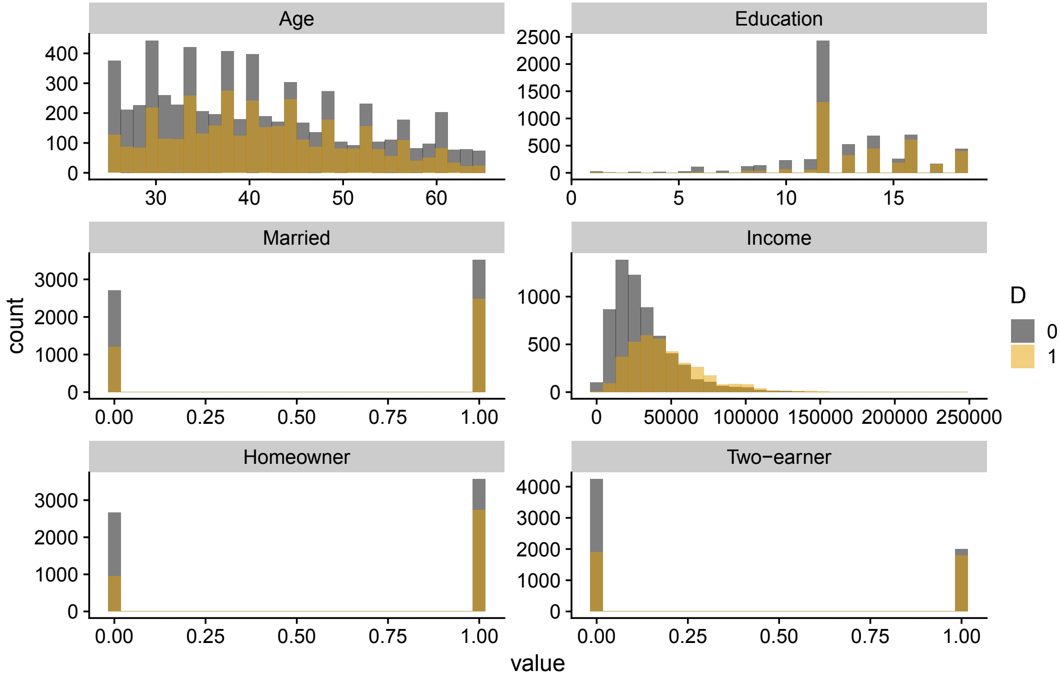


Figure 2.8: Unweighted distribution of variables given treatment status.

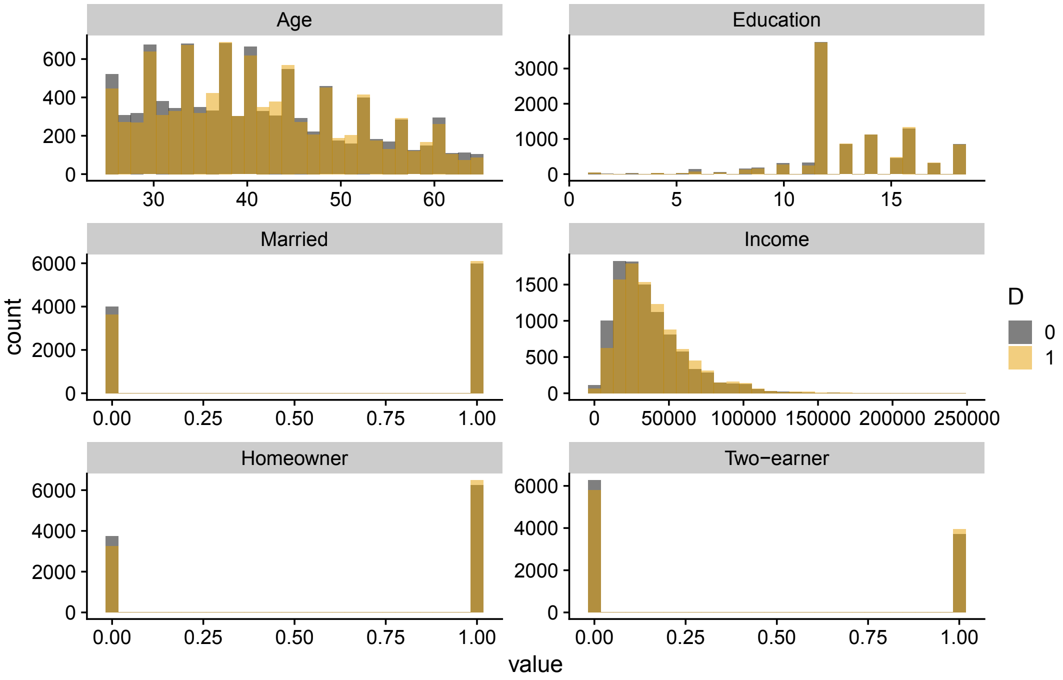


Figure 2.9: IP weighted distribution of variables given treatment status.Q$_{IPW}$

2.4 Simulated Data

Since the true treatment effect is never known beforehand, we provide a simulation to evaluate different approaches in terms of performance for parameter estimation. The data-generating process allows controlling the number of observations, dimensionality, and the distributions of the variables. The possibility to specify datasets for different simulations and scenarios helps to investigate the methods used in this tutorial. Note that simulated data often lack realistic data structures. An alternative is to rely on synthetic data where only the treatment effect is artificially added. Since a simulation study in the type of a Monte Carlo study is not the main focus of this tutorial, we will only use two simulated data generating processes. The purpose is to give an idea of how to simulate data and test different methods. Instead of relying on purely artificial data, Wendling et al. (2018) creates synthetic data based on real covariates and a treatment assignment mechanism. Only the outcome is simulated based on non-parametric models of the real outcome.

2.4.1 Data Generating Process

The basic model used in this tutorial is a partially linear regression model based on Robinson (1988) with extensions:

$$Y = \tau(X_i)D + \mu_0(X_i) + U, \qquad \mathsf{E}[U|X, D] = 0, \tag{2.13}$$

$$D = e(X_i) + V, \qquad \mathsf{E}[V|X] = 0, \tag{2.14}$$

$$\tau(X_i) = t(X_i) + W, \qquad \mathsf{E}[W|X] = 0. \tag{2.15}$$

Let Y be the outcome variable. $\tau(X_i)$ is the true treatment effect or population uplift, while D is the treatment status. The vector $X = (X_1, ..., X_p)$ consists of p different features or covariates and U, V and W are unobserved covariates which follow a random normal distribution = $N(0, 1)$.

Equation 5.25 is the propensity score. In the case of completely random treatment assignment, the propensity score is constant for all units, and, if equally distributed, then $e(X_i) = 0.5$. The covariates X are generated from a random multivariate normal distribution ($N(0, 1)$). Note that all values are continuous. In business applications, discrete values (categorical variables) are very common. For the data generation process as well as for the evaluation of most models, it would make no difference if such variables are present. This is because vanilla machine learning methods can handle categorical variables quite well. An exception is the causal forest where one has to use one-hot encoding, to transform the variable into dummies. Next, we describe the generation of the functions in detail.

Covariates (X)

1. Generate a random positive definite covariance matrix Σ based on a uniform distribution over the space $p \times p$ of the correlation matrix. Let $p = 20$.
2. Scale the covariance matrix. This equals the correlation matrix and can be seen as the covariance matrix of the standardized random variables $\Sigma = \frac{X}{\sigma(X_i)}$.
3. Generate random normal distributed variables $X_{N \times p}$ with mean = 0 and variance = Σ.

The function $\mu_0(X_i)$ is calculated using a linear function with interaction terms and contains the following covariates:

$$\mu_0(X) = X_1 \otimes X_2 + X_3 \otimes X_4 + X_5. \tag{2.16}$$

All covariates are normal distributed except X_5 which only takes four values, namely $\{-0.2, 0, 0.2, 0.6\}$. Next, we describe how to build the function $e(X_i)$ as well as how to create heterogeneous treatment effects. A varying treatment effect implies that its strength differs among the observations and is therefore conditioned on some covariates X. Regarding the treatment assignment, two settings are considered. Setting 1 assumes D to be completely randomly assigned among the observations. In this case, D is just a vector of random numbers with values 0 or 1. In setting 2, the treatment assignment is dependent on the covariates. The binary treatment assignment is generated through a Bernoulli function. This implies per default a sort of uncertainty or random error. Even if the probability from the propensity score is 90% for $D = 1$, there is still a 10% chance that it is generated to be zero. The functions are generated as follows:

Treatment Assignment (e_0)

Setting 1: e_0

$$D \overset{ind.}{\sim} \text{Bernoulli}(e_0), \qquad\qquad \text{with} \quad e_0 = 0.5$$

Setting 2: $e(X_i)$

1. Dependence is non-linear (through interaction terms): $a(X_i) = X_1 \otimes X_2 + X_3 \otimes X_4$.
2. Calculate the probability distribution for the vector a from the normal distribution function:

$$e(X_i) = \Phi\left(\frac{a - \mu(a)}{\sigma(a)}\right) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{a - \mu(a)}{\sigma(a)\sqrt{2}}\right)\right]$$

   Here $\mu(a)$ denotes the mean of a and $\sigma(a)$ the standard deviation.

3. Apply a random number generator from a Binomial function $B\{N, e(X_i)\}$ with probability for success = $e(X_i)$. This creates a vector $D \in \{0; 1\}$ such that $D \overset{ind.}{\sim}$ Bernoulli$\{e(X_i)\}$.

Regarding the treatment effect, we also consider two different settings. First, $\tau(X_i)$ depends linear on covariates X, and second, $\tau(X_i)$ has a non-linear, more complex form concerning the covariates. In both settings, we can examine heterogeneous treatment effects. The vector $b = \frac{1}{l}$ with $l \in \{1, 2, ..., p\}$ represents weights for every covariate.

Treatment Effect ($\tau(X_i)$)

Setting 1: linear dependence

$$\tau(X_i) = 0.6X_1 + 0.6X_2 + 0.6X_3 + 0.6X_4 + X_5 + W \quad \text{with} \quad W \sim \mathcal{N}(0, 0.5).$$

Setting 2: non-linear dependence

$$\tau(X_i) = \sin(X_{1:3} \times b_{1:3}) + 1.5\cos(X_4) + X_5.$$

The simulated data that include the true treatment effect can be found here: Q$_{sim}$.

2.4.2 Results

To evaluate the different methods, we consider two data generating processes (DGP). Setting 1 is a randomized controlled trial with a constant propensity score of 0.5, while the treatment effects depend linear on covariates. In setting 2, we consider confounding variables, namely that the treatment probability now depends on covariates (through interactions of covariates) while the treatment effect depends non-linear on the covariates. In both settings, we set $N = 2000$ and $p = 20$. We use up to 5 variables to generate the different variables and the treatment effects while all other variables have no dependence on any function. They are spurious and the hope is that the ML methods find the important variables while excluding the others. Table 2.11 shows the mean squared error (MSE) for all considered methods and both settings. We list to different versions of the DR- and R-learner. The first is the in-sample estimator where the regression and estimation of the CATE based on the pseudo-outcome or R-loss is done on the same sample. Only the nuisance functions are regressed and estimated on different samples. This is in line with the sample splitting theory. In the last step, we just want to have a good approximation of the CATE which is why we can use the same sample for training and prediction. Note that the DR-learner already estimates the CATE in the pseudo-outcome. Using the whole sample should increase the prediction power.

In practice, however, we find that it might be better to split the sample again and not use the same sample for training and prediction. The reason is the following: If the predictions of the nuisance functions are not perfect, the pseudo-outcome deviates from the true CATE. The deviation becomes clearer with a higher estimation error and also if there are extreme values in the propensity score. Using different samples in the last step aims to smooth the function and discard outliers. This approach adds sample splitting (the two-step sample splitting without cross-fitting). Here we apply this approach with cross-fitting. This means we not only want to have different samples for training and prediction but also want to average the prediction fold over different training samples. Therefore we split the sample into 6 folds (the proportions are 10% for the first 5 folds and 50% for the last one). We use fold 1 to 5 individually to train regression functions and predict on all fold 6. Then we average the 5 estimates for fold 6. Now we reverse the role, combining fold 1 to 5 and split fold 6 in 5 parts. We proceed as above. We call the sample split estimator simply cross-fit estimator. Table 2.11 shows the results for both versions. Using the cross-fit version we can decrease the MSE by at least 50%. In simulations, we find that even a 50:50 split where we use 50% for training and predict on the other half can decrease the MSE. The cross-fit version turns out to further decrease the MSE in simulations with different DGP's. For completeness, we show the procedure of the 50:50 split approach in Figure 2.14 in the Appendix.

Table 2.10 shows that in setting 1 the tree-based methods (boosting and random forest) perform best in predicting the propensity score while the neural network does better in the regression tasks. The lasso only gets significant weight in the treatment effect regression. The lasso is excluded when applying the R-learner since in a linear setting the loss-function slightly differs from the more general non-parametric one. If the data generating process becomes more complex, the lasso method becomes less important shifting weights towards the neural network. In setting 2, the tree-based methods are most important in all tasks but for the treatment-effect regression based on the R-learner. We also experimented with excluding the neural network and found that in the linear setting, more weight is based on the lasso, while in the non-linear setting, the tree-based methods are superior. Since all methods are important in at least one task, we include all methods but the lasso when creating bootstrapped confidence intervals.

Table 2.10: Weights of ML methods.

	$\hat{e}(X)$	$\hat{\mu}_0(X)$	$\hat{\mu}_1(X)$	$\hat{\mu}(X)$	DR	R
Setting 1						
Boosting	0.91	0.11	0.27	0.27	0.26	0.12
Lasso	0.00	0.00	0.01	0.00	0.34	—
Neural Network	0.00	0.83	0.70	0.68	0.00	0.75
Random Forest	0.09	0.06	0.02	0.04	0.39	0.12
Setting 2						
Boosting	0.40	0.62	0.72	0.72	0.14	0.13
Lasso	0.00	0.00	0.00	0.00	0.09	—
Neural Network	0.00	0.07	0.06	0.07	0.13	0.39
Random Forest	0.60	0.31	0.22	0.21	0.65	0.47

Figure 2.10 and 2.12 show the sorted treatment effect heterogeneity with 95% confidence intervals for setting 1 and 2, respectively. While the causal BART method produces the lowest MSE, it has higher credible intervals than the causal forest. Figure 2.20 in the

Table 2.11: MSE for different methods. Q$_{res}$

Category	Method	MSE Setting 1	MSE Setting 2
Meta-Learner	DR-learner (in-sample)	2.68	3.42
	DR-learner (cross-fit)	1.11	1.17
	R-learner (in-sample)	2.36	2.58
	R-learner (cross-fit)	0.80	1.34
	T-learner	1.17	2.09
	X-learner	0.58	1.10
Modified ML Methods	Causal BART	0.55	0.48
	Causal Forest	0.86	1.34

Appendix shows boxplots of all methods and their variation. The blue line indicates the true ATE, hence we can see how accurate all methods are to predict the ATE. We find that all methods are unbiased if the DGP is linear. The bias increases if the functions are more complex as shown in Figure 2.21. Figure 2.20 and 2.21 also shows the decrease in outliers for the DR- and R-learner when we apply additional sample splitting and cross-fitting. We do not observe these outliers for the X-learner. In Figure 2.11 and 2.13, we plot scatterplots for the estimated vs. the true CATE. The blue line indicates a linear regression estimate for each method. As we have seen from the MSE, the causal BART method performs best over the whole interval and in both settings. One observation is that the meta-learners estimate the CATE with higher variance (and potentially producing more outliers that need to be controlled for) than the two modified ML methods. The T-learner has the highest variance in both settings while the DR- and R-learner show the second-highest variance in setting 1 and 2. Looking at the correlation of the methods, we find that the highest correlation is between the causal BART and causal forest ($\rho = 0.85, 0.81$ for the two settings). In general, the correlation is quite high in both settings (ranging from $\rho = 0.64$ to $\rho = 0.85$). However, we do not see any improvement in the correlation in functions that are easier to estimate.

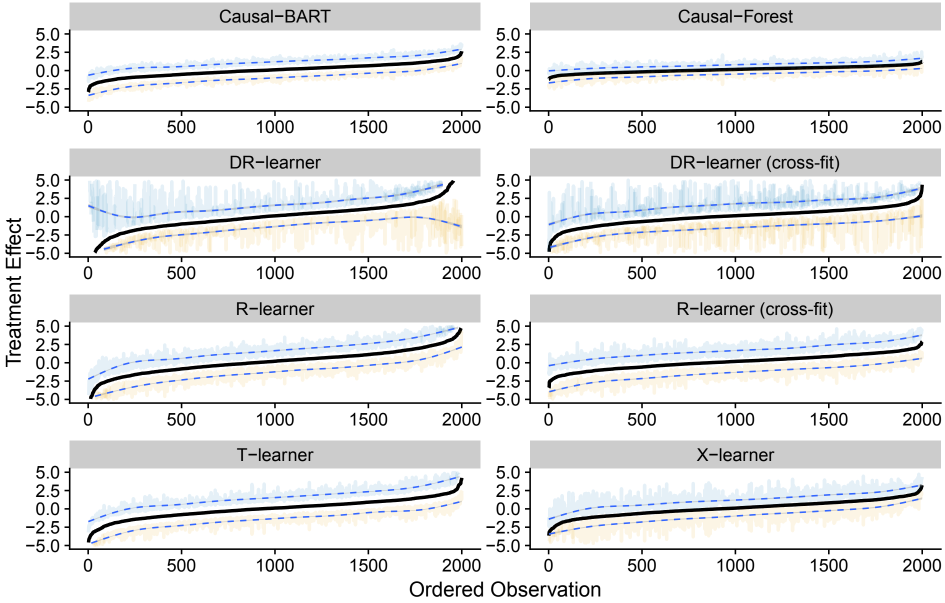


Figure 2.10: Setting 1: Observations sorted by treatment effect. Q$_{S1}$

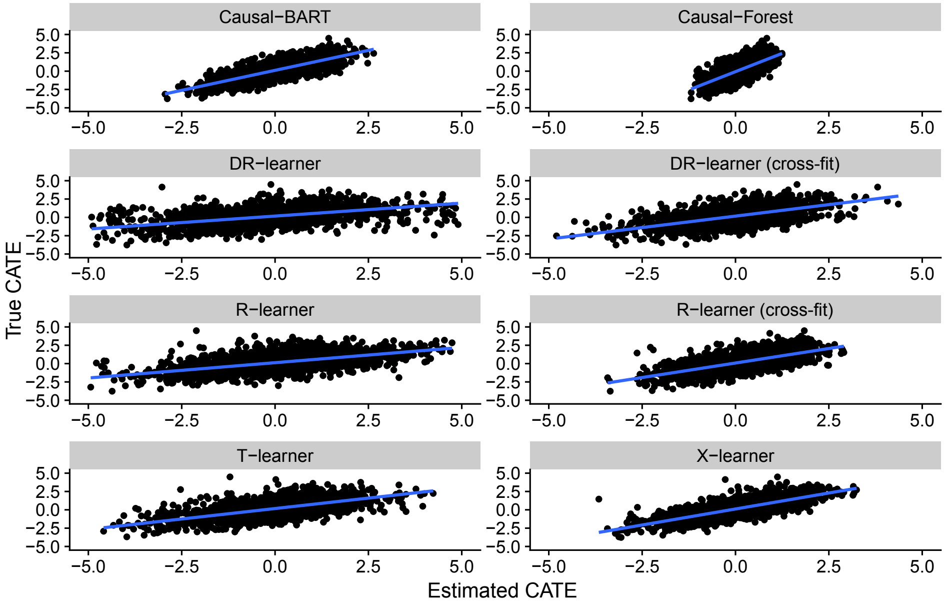


Figure 2.11: Setting 1: Scatterplot of estimated and true CATE. Q$_{T1}$

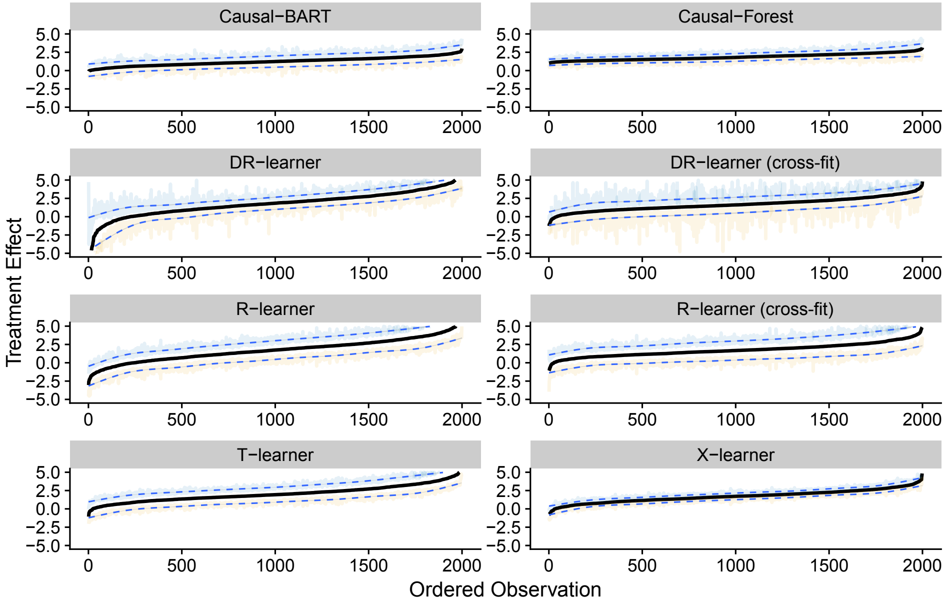


Figure 2.12: Setting 2: Observations sorted by treatment effect. $_{S2}$

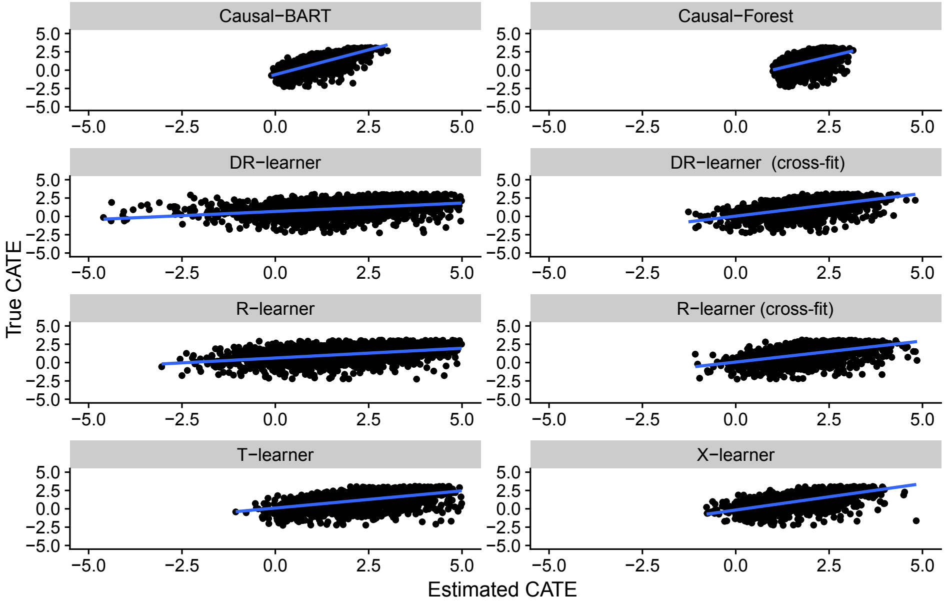


Figure 2.13: Setting 2: Scatterplot of estimated and true CATE. $_{T2}$

2.5 Conclusion

In this tutorial, we present novel methods to estimate the conditional average treatment effect using machine learning methods. We categorize the methods into two branches. First, so-called meta-learners, that make use of off-the-shelf machine learning methods by creating a transformed outcome to estimate the CATE. They are flexible in the choice of the machine learning method as long as they converge with a specific rate. For example, we can use classification and regression trees, random forest, boosting methods, and even neural networks. The second branch contains machine learning methods that are specific designed to estimate the CATE. These methods rely on trees or an ensemble of trees like the generalized random forest, causal boosting, and a Bayesian approach using additive regression trees. The use of meta-learners needs special care because they are quite flexible in the choice of the ML method and also concerning sample splitting. We, therefore, provide pseudo-code along with R-code for many of such meta-learners and show how they can be used to estimate the CATE on the whole dataset. We also demonstrate how to use the second branch of methods by integrating the packages in R-code that uses the same data structure as the meta-learners. When possible we apply cross-fitting as an averaging procedure of a subset of the data conditional on different training folds.

To demonstrate the strength and differences of all the methods that we consider, we present four examples. Two empirical examples, the first from a randomized control trial and the second from an observational study. The third and fourth examples contain simulated data where the true treatment effect can be observed and hence compared with the estimates from all the methods. In the empirical examples, we find strong evidence of positive treatment effects for each observation while significant heterogeneity in the effects is not that clear. This is mainly if we base the conclusion on calculated confidence intervals via the bootstrap or credible intervals. We do, however, find differences in the width of the confidence intervals and also in the CATE prediction among the methods. These differences also occur in the simulated data. We, therefore, recommend that practitioners not rely on only one method but rather use multiple methods and compare the results. One should also carefully think about the different tuning parameters that can be set when using machine learning methods. Depending on the method there can be a variety of options to consider. We try to avoid the problem of manually selecting such parameters through cross-validation and the selection of different ML methods for each nuisance function. Sample splitting and cross-fitting is a further necessary step to get robust and accurate estimates among the methods. One observation from this simulation is clearly that the meta-learners can improve in terms of MSE with simpler functions. We note that the results heavily depend on the chosen ML methods. Through applying different ML methods in the Super Learner we find that the selection of the best ML method depends on the data generating process and varies across the functions. For example, if the data structure is quite complicated and non-linear, a model based on the lasso might not be the best choice. Including more ML methods could improve the prediction accuracy depending on the data generating process. Using two-step sample splitting with cross-fitting further improves the prediction.

Bibliography

Athey, S. (2019). The impact of machine learning on economics. *The economics of artificial intelligence* (pp. 507–552). University of Chicago Press.

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *Annals of Statistics*, *47*(2), 1148–1178. https://doi.org/10.1214/18-AOS1709

Athey, S., & Wager, S. (2017). Efficient policy learning. *arXiv preprint arXiv:1702.02896*.

Bickel, P. J. (1982). On adaptive estimation. *Annals of Statistics*, *10*(3), 647–671. https://doi.org/10.1214/aos/1176345863

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, *21*(1), C1–C68. https://doi.org/10.1111/ectj.12097

Chernozhukov, V., Demirer, M., Duflo, E., & Fernández-Val, I. (2018). Generic machine learning inference on heterogeneous treatment effects in randomized experiments, with an application to immunization in india. https://doi.org/10.3386/w24678

Chernozhukov, V., Escanciano, J. C., Ichimura, H., Newey, W. K., & Robins, J. M. (2016). Locally robust semiparametric estimation. *arXiv preprint arXiv:1608.00033*. https://doi.org/10.1920/wp.cem.2016.3116

Chernozhukov, V., & Hansen, C. (2004). The effects of 401 (k) participation on the wealth distribution: An instrumental quantile regression analysis. *Review of Economics and statistics*, *86*(3), 735–751. https://doi.org/10.1162/0034653041811734

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, *4*(1), 266–298. https://doi.org/10.1214/09-AOAS285

Crépon, B., Devoto, F., Duflo, E., & Parienté, W. (2015). Estimating the impact of microcredit on those who take it up: Evidence from a randomized experiment in morocco. *American Economic Journal: Applied Economics*, 7(1), 123–50. https://doi.org/10.1257/app.20130535

EconML, M. R. (2019). EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation [Version 0.x]. https://github.com/microsoft/EconML, Last accassed: 20.06.2021.

Fan, Q., Hsu, Y.-C., Lieli, R. P., & Zhang, Y. (2020). Estimation of conditional average treatment effects with high-dimensional data. *Journal of Business & Economic Statistics*, 1–15. https://doi.org/10.1080/07350015.2020.1811102

Farrell, M. H., Liang, T., & Misra, S. (2021). Deep neural networks for estimation and inference. *Econometrica*, *89*(1), 181–213. https://doi.org/10.3982/ecta16901

Foster, J. C., Taylor, J. M., & Ruberg, S. J. (2011). Subgroup identification from randomized clinical trial data. *Statistics in medicine*, *30*(24), 2867–2880. https://doi.org/10.1002/sim.4322

Friedberg, R., Tibshirani, J., Athey, S., & Wager, S. (2018). Local linear forests. arxiv. *arXiv preprint arXiv:1807.11408*.

Hahn, P. R., Murray, J. S., & Carvalho, C. M. (2020). Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects. *Bayesian Analysis*, *15*. https://doi.org/10.1214/19-BA1195

Hansotia, B., & Rukstales, B. (2002). Incremental value modeling. *Journal of Interactive Marketing*, *16*(3), 35–46. https://doi.org/10.1002/dir.10035

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, *20*(1), 217–240. https://doi.org/10.1198/jcgs.2010.08162

Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, *47*(260), 663–685. https://doi.org/10.1080/01621459.1952.10483446

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

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

Knaus, M. C., Lechner, M., & Strittmatter, A. (2020). Machine Learning Estimation of Heterogeneous Causal Effects: Empirical Monte Carlo Evidence. *The Econometrics Journal*. https://doi.org/10.1093/ectj/utaa014

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, *116*(10), 4156–4165. https://doi.org/10.1073/pnas.1804597116

Lunceford, J. K., & Davidian, M. (2004). Stratification and weighting via the propensity score in estimation of causal treatment effects: A comparative study. *Statistics in Medicine*, *23*(19), 2937–2960. https://doi.org/10.1002/sim.1903

McCaffrey, D. F., Ridgeway, G., & Morral, A. R. (2004). Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological methods*, *9*(4), 403–425. https://doi.org/10.1037/1082-989X.9.4.403

Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, *31*(2), 87–106.

Newey, W. K., & Robins, J. R. (2018). Cross-fitting and fast remainder rates for semiparametric estimation. *arXiv preprint arXiv:1801.09138*. https://doi.org/10.1920/wp.cem.2017.4117

Nie, X., & Wager, S. (2020). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*. https://doi.org/10.1093/biomet/asaa076

Polley, E. C., Rose, S., & Van der Laan, M. J. (2011). *Super learning*. Springer. https://doi.org/10.1007/978-1-4419-9782-1_3

Poterba, J. M., Venti, S. F., & Wise, D. A. (1992). *401(k) plans and tax-deferred saving* (Working Paper No. 4181). National Bureau of Economic Research. https://doi.org/10.3386/w4181

Powers, S., Qian, J., Jung, K., Schuler, A., Shah, N. H., Hastie, T., & Tibshirani, R. (2018). Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in Medicine*, *37*(11), 1767–1787. https://doi.org/10.1002/sim.7623

Robins, J. M., Zhang, P., Ayyagari, R., Logan, R., Tchetgen, E. T., Li, L., Lumley, T., & van der Vaart, A. (2013). New statistical approaches to semiparametric regression with application to air pollution research. *Research report (Health Effects Institute)*, (175), 3–129.

Robinson, P. M. (1988). Root-n-consistent semiparametric regression. *Econometrica*, 931–954. https://doi.org/10.2307/1912705

Rotnitzky, A., Robins, J., & Babino, L. (2017). On the multiply robust estimation of the mean of the g-functional. *arXiv preprint arXiv:1705.08582*.

Rubin, D. B. (1980). Randomization analysis of experimental data: The fisher randomization test comment.*Journal of the American Statistical Association*, *75*(371), 591–593. https://doi.org/10.2307/2287653

Schick, A. (1986). On asymptotically efficient estimation in semiparametric models. *Annals of Statistics*, *14*(3), 1139–1151. https://doi.org/doi:10.1214/aos/1176350055

Sharma, A., Kiciman, E. et al. (2019). DoWhy: A Python package for causal inference. *https://github.com/microsoft/dowhy, Last accessed: 17.06.2021.*

van der Laan, M. J. (2010). Targeted maximum likelihood based causal inference: Part i. *The international journal of biostatistics*, *6*(2). https://doi.org/10.2202/1557-4679.1211

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests.*Journal of the American Statistical Association*, *113*(523), 1228–1242. https://doi.org/10.1080/01621459.2017.1319839

Wendling, T., Jung, K., Callahan, A., Schuler, A., Shah, N., & Gallego, B. (2018). Comparing methods for estimation of heterogeneous treatment effects using observational data from health care databases. *Statistics in medicine*, *37*(23), 3309–3324. https://doi.org/10.1002/sim.7820

Wyss, R., Ellis, A. R., Brookhart, M. A., Girman, C. J., Jonsson Funk, M., LoCasale, R., & Stürmer, T. (2014). The role of prediction modeling in propensity score estimation:

An evaluation of logistic regression, bcart, and the covariate-balancing propensity score. *American journal of epidemiology*, *180*(6), 645–655. https://doi.org/10.1093/aje/kwu181

2.A Additional Proofs

Proof of the doubly-robustness property for the DR-learner. If either the propensity score or the conditional mean is correctly specified, the doubly-robust estimator is an unbiased estimator. Let us look at $\hat{mu}_1(x)$, the same procedure holds analog or $\hat{\mu}_0$. That is, $\hat{\mu}_1(x)$ estimates the following:

$$\begin{aligned}
&=\mathsf{E}\left[\mu_1(x)+\frac{D\left\{Y-\mu_1\left(x\right)\right\}}{e\left(x\right)}\right]\\
&=\mathsf{E}\left[\frac{e(x)}{e(x)}\mu_1(x)+\frac{D\left\{Y-\mu_1\left(x\right)\right\}}{e\left(x\right)}\right]\\
&=\mathsf{E}\left[\frac{DY}{e(x)}-\frac{(D-e(x))\mu_1(x)}{e(x)}\right]\text{, add }\frac{e(x)}{e(x)}Y,-\frac{e(x)}{e(x)}Y\\
&=\mathsf{E}\left[\frac{(D-e(x))Y}{e(x)}-\frac{(D-e(x))\mu_1(x)}{e(x)}+\frac{e(x)}{e(x)}Y\right]\\
\hat{\mu}_1(x)&=\mathsf{E}\left[\frac{(D-e(x))(Y^1-\mu_1(x))}{e(x)}+Y^1\right]\\
\hat{\mu}_1(x)&=\mathsf{E}[Y^1]+\mathsf{E}\left[\frac{(D-e(x))(Y^1-\mu_1(x))}{e(x)}\right] \qquad (2.17)
\end{aligned}$$

2.B Additional Plots

Figure 2.14 describes the two-step sample splitting. The first splitting is necessary for nuisance parameter estimation. The second split is done to get out-of-bag estimates of the CATE. In this approach no cross-fitting is applied. Figure 2.15 now uses the samples that include the pseudo-outcomes and split it into different folds. Each fold is used to train a regression model. Prediction is done in the out-of-bag fold. Through the different folds we get as many predictions as folds used for training. These estimates are then averaged which applies the cross-fitting procedure.

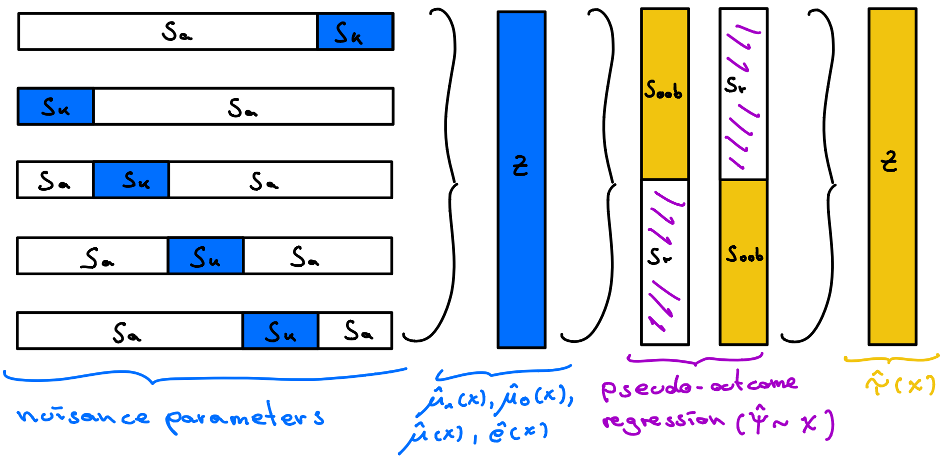

Sa
Sk
Su
Sa
Sa
Su
Sa
Sa
Su
Sa
Sa
Su
Sa
z
Soob
Sr
Sr
Soob
z
nuisance parameters
$\hat{\mu}_1(x), \hat{\mu}_0(x),$
$\hat{\mu}(x), \hat{e}(x)$
pseudo-outcome
regression ($\hat{\Psi} \sim X$)
$\hat{\tau}(x)$

Figure 2.14: Two-step sample splitting: 50:50, no cross-fitting.

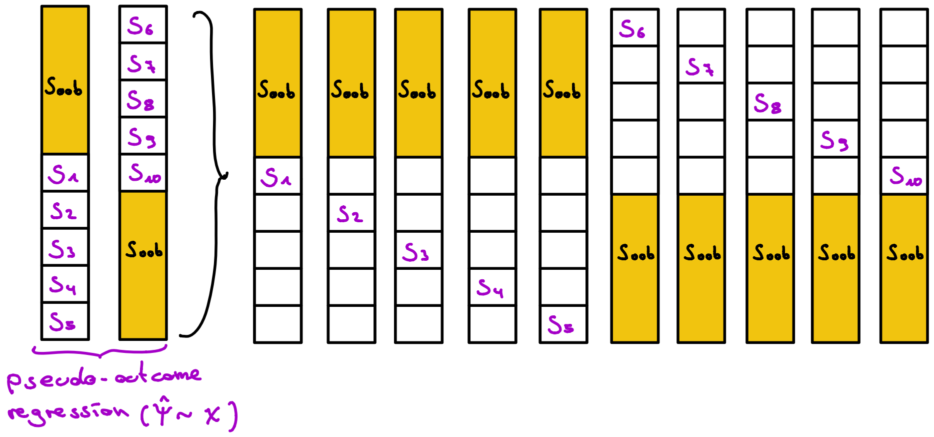


Figure 2.15: Detailed cross-fitting procedure for 5 folds.

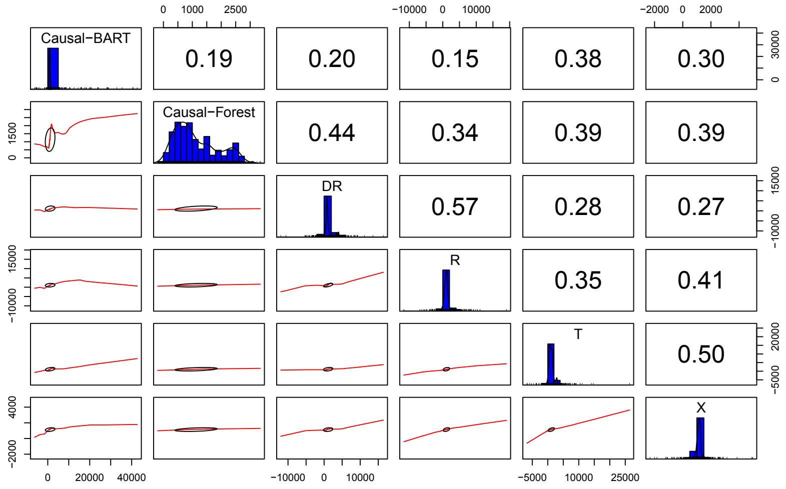


Figure 2.16: Correlation of CATE between different methods from the microcredit data. Q_{C1}

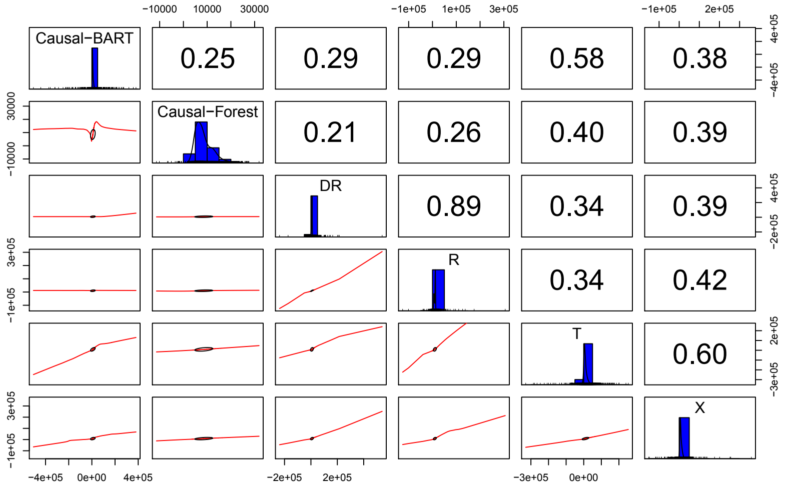


Figure 2.17: Correlation of CATE between different methods from the 401k data. $_{C2}$

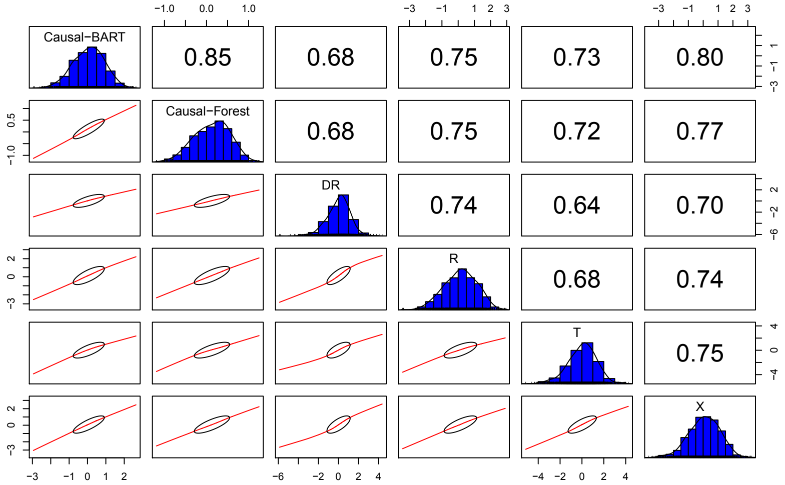


Figure 2.18: Correlation of CATE between different methods for simulation setting 1. $_{C3}$

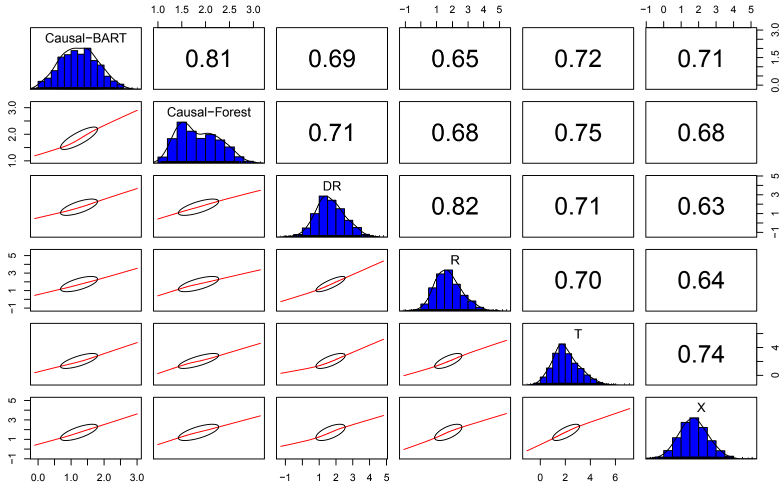


Figure 2.19: Correlation of CATE between different methods for simulation setting 2. $_{C4}$

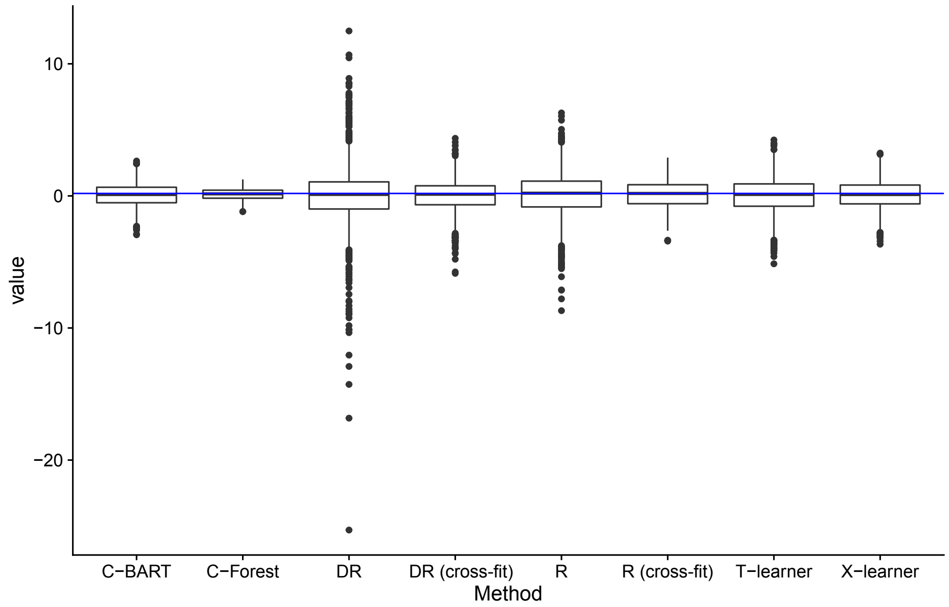


Figure 2.20: Setting 1: Boxplots of different methods. $_{B3}$

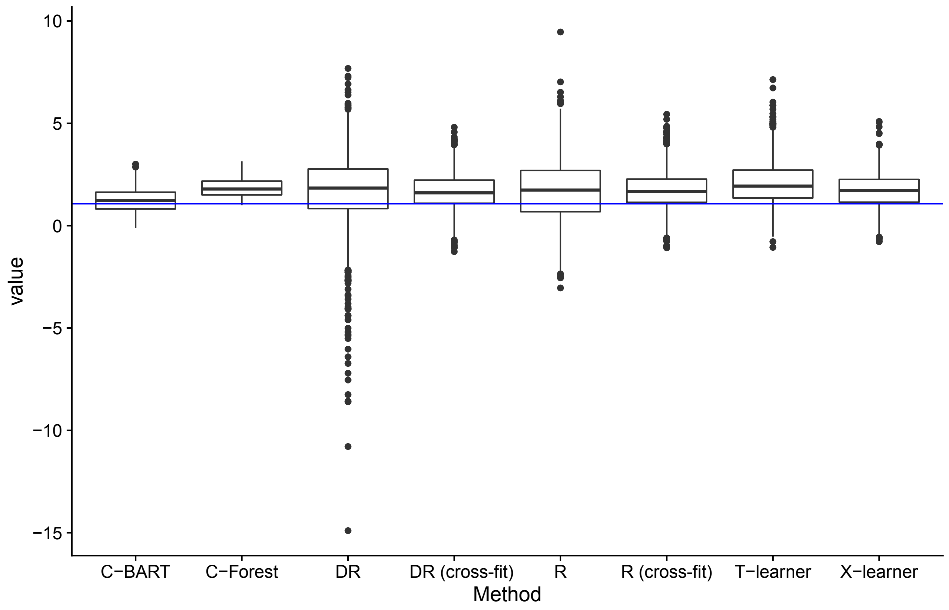


Figure 2.21: Setting 2: Boxplots of different methods. $_{B4}$

2.C Tables

Classification results for the microcredit example:

Table 2.12: CLAN results for different methods.

	DR-learner			R-learner			T-learner		
	Most Affected	Least Affected	Difference	Most Affected	Least Affected	Difference	Most Affected	Least Affected	Difference
Head age	19.19	46.92	-27.73	29.99	42.52	-12.53	23.43	38.17	-14.74
	(17.81,20.58)	(45.53,48.30)	(-29.68,-25.77)	(28.50,31.49)	(40.98,44.06)	(-14.67,-10.38)	(21.93,24.94)	(36.67,39.68)	(-16.87,-12.61)
	-	-	[0.000]	-	-	[0.000]	-	-	[0.000]
Non-agricultural	0.118	0.186	-0.068	0.178	0.097	0.081	0.149	0.126	0.023
self-employed	(0.097,0.139)	(0.165,0.207)	(-0.098,-0.038)	(0.158,0.198)	(0.076,0.118)	(0.052,0.110)	(0.128,0.169)	(0.106,0.146)	(-0.006,0.051)
	-	-	[0.000]	-	-	[0.000]	-	-	[0.244]
Borrowed from	0.138	0.338	-0.201	0.181	0.320	-0.139	0.116	0.273	-0.157
any source	(0.113,0.162)	(0.314,0.363)	(-0.235,-0.166)	(0.155,0.206)	(0.294,0.346)	(-0.175,-0.103)	(0.093,0.139)	(0.250,0.296)	(-0.189,-0.125)
	-	-	[0.000]	-	-	[0.350]	-	-	[0.000]
	X-learner			Causal BART			Causal Forest		
	Most Affected	Least Affected	Difference	Most Affected	Least Affected	Difference	Most Affected	Least Affected	Difference
Head age	19.66	38.94	-19.28	14.85	50.05	-35.21	10.25	46.80	-36.55
	(18.19,21.14)	(37.47,40.42)	(-21.37,-17.19)	(13.68,16.02)	(48.88,51.22)	(-36.86,-33.55)	(9.271,11.22)	(45.82,47.77)	(-37.93,-35.17)
	-	-	[0.000]	-	-	[0.000]	-	-	[0.000]
Non-agricultural	0.138	0.181	-0.043	0.150	0.099	0.051	0.073	0.136	-0.064
self-employed	(0.116,0.160)	(0.159,0.202)	(-0.073,-0.012)	(-0.073,-0.012)	(0.080,0.119)	(0.130,0.169)	(0.118,0.154)	(0.055,0.091)	(0.038,0.089)
	-	-	[0.013]	-	-	[0.000]	-	-	[0.000]
Borrowed from	0.091	0.417	-0.327	0.091	0.300	-0.210	0.050	0.388	-0.338
any source	(0.067,0.115)	(0.394,0.441)	(-0.360,-0.293)	(0.068,0.113)	(0.278,0.323)	(-0.242,-0.178)	(0.028,0.072)	(0.366,0.411)	(-0.370,-0.307)
	-	-	[0.000]	-	-	[0.000]	-	-	[0.000]

Notes: Averages are taken from the mean of the CATE over 500 bootstrap iterations.

Additional Pseudocode

Algorithm 7: S-learner

```
Input: $Z_i = \{Y_i, D_i, X_i\}_{i \in N}$
Split sample $Z$ into $K$ random subsets
for $k$ in {1, ...,K} do
    assign Sample $S_a = Z \cup S_k$ and $S_k$
    regress $Y_i = \hat{\mu}(X_i, D_i) + \hat{U}_i$, with $i \in S_a$
        estimate $\hat{Y}_i^0 = \hat{\mu}(X_i, D = 0)$, with $i \in S_k$
        estimate $\hat{Y}_i^1 = \hat{\mu}(X_i, D = 1)$, with $i \in S_k$
    create $\hat{\tau}_k(X_i) = \hat{\mu}(X_i, D = 1) - \hat{\mu}(X_i, D = 0)$
combine $\hat{\tau}(X_i) = \{\hat{\tau}_1, \hat{\tau}_k, \ldots, \hat{\tau}_K\}$
```

Algorithm 8: Bootstrap Confidence Interval

```
Input: $Z_i = \{Y_i, D_i, X_i\}_{i \in N}$
$p$ : evaluation point (out-of-sample)
$S_0 = \{i : D_i = 0\}$
$S_1 = \{i : D_i = 1\}$
$n_0 = \#S_0$
$n_1 = \#S_1$
for $b$ in $\{1, \ldots, B\}$ do
    $S_b^* = c\,(\text{sample}(S_0, S_1))$
    $y_b^* = y\left[s_b^*\right]$
    $d_b^* = d\left[s_b^*\right]$
    $x_b^* = x\left[s_b^*\right]$
    $\hat{\tau}_b^*(p) = \text{learner}\left(y_b^*, d_b^*, x_b^*\right)(p)$
$\hat{\tau}(p) = \text{learner}(y, d, x)(p)$
$\hat{\sigma} = \text{sd}\left(\left\{\hat{\tau}_b^*(p)\right\}_{b=1}^{B}\right)$
return $\left(\hat{\tau}(p) - q_{\alpha/2}\hat{\sigma}, \hat{\tau}(p) + q_{1-\alpha/2}\hat{\sigma}\right)$
```

Algorithm 9 refers to the inverse probability weighting (IPW) estimator based on Horvitz and Thompson, 1952. While Künzel et al., 2019 refer to this estimator as the F-learner, it is also known as the (simplest) transformed outcome estimator. This is because $\hat{\psi}_{IPW}$ is an unbiased estimate of the ATE. The only nuisance function that is needed to create this outcome is the propensity score. We then map the covariates onto the transformed outcome (or pseudo-outcome). The reason for this mapping is again to smooth the function since the IPW estimator can suffer from high variance if the propensity score estimates are near zero or one. Below we show that the IPW estimator is an unbiased estimator for the ATE.

$$\begin{aligned}
\mathsf{E}[\psi_{IPW}|X_i = x] &= \mathsf{E}[Y\{\frac{D}{e(x)} - \frac{1-D}{1-e\,(X_i)}\}|X_i = x] \\
&= \mathsf{E}[\{D\frac{Y}{e(x)} - (1-D)\frac{Y}{1-e\,(x)}\}|X_i = x] \\
&= \mathsf{P}(D = 1|X_i = x)\frac{1}{e(x)}\mathsf{E}[Y|D = 1, X_i = x] \\
&\qquad - \mathsf{P}(D = 0|X_i = x)\frac{1}{1-e(x)}\mathsf{E}[Y|D = 0, X_i = x] \\
&= \mathsf{E}[Y|D = 1, X_i = x] - \mathsf{E}[Y|D = 0, X_i = x] \\
&= \mu_1(x) - \mu_0(x) = \tau(x)
\end{aligned}$$

Algorithm 9: IPW-learner

```
Input: $Z_i = \{Y_i, D_i, X_i\}_{i \in N}$
Split sample $Z$ into $K$ random subsets
for k in {1, ...,K} do
    assign Sample $S_a = Z \uplus S_k$ and $S_k$
    regress $D_i = \hat{e}\,(X_i) + \hat{V}_i$, with $i \in S_a$
        estimate $\hat{D}_i = \hat{e}\,(X_i)$, with $i \in S_k$
    create $\hat{\psi}_{IPW} = Y\{\frac{D}{\hat{e}(x)} - \frac{1-D}{1-\hat{e}(X_i)}\}$
    store $\hat{\psi}_{IPW,k}$ for $i \in S_k$
Cross-fitting:
for oob in (1:2) do
    if oob = 1: $S_{oob} = Z_i$ with $i \in \{1, ..., N/2\}$ and $S_{train} = Z_i \uplus S_{oob}$
    if oob = 2: $S_{train} = Z_i$ with $i \in \{1, ..., N/2\}$ and $S_{oob} = Z_i \uplus S_{in}$
    for l in 1:5 do
        split $S_{train}$ in $\{S_1, S_2, \ldots, S_5\}$
        regress $\hat{\psi}_i = \hat{t}_{IPW}(X_i) + W_i$, for $i \in S_l$
            estimate $\tilde{\tau}_l(X_i) = \hat{t}_{IPW}(X_i)$, with $i \in S_{oob}$
    average $\hat{\tau}_{oob}(X_i) = \mathsf{E}[\tilde{\tau}(X_i)]$
row bind $\hat{\tau}(X_i) = \{\hat{\tau}_1, \hat{\tau}_2\}$
```

Chapter 3

Variable Selection for Causal Inference via Outcome-Adaptive Random Forest

ABSTRACT

Estimating a causal effect from observational data can be biased if we do not control for self-selection. This selection is based on confounding variables that affect the treatment assignment and the outcome. Propensity score methods aim to correct for confounding. However, not all covariates are confounders. We propose the outcome-adaptive random forest (OARF) that only includes desirable variables for estimating the propensity score to decrease bias and variance. Our approach works in high-dimensional datasets and if the outcome and propensity score model are non-linear and potentially complicated. The OARF excludes covariates that are not associated with the outcome, even in the presence of a large number of spurious variables. Simulation results suggest that the OARF produces unbiased estimates, has a smaller variance, and is superior in variable selection compared to other approaches. The results from two empirical examples, the effect of right heart catheterization on mortality and the effect of maternal smoking during pregnancy on birth weight, show comparable treatment effects to previous findings but tighter confidence intervals and more plausible selected variables.

JEL Classification: C31, C52, C54, C63

Replication code is available on Quantlet Q.

3.1 Introduction

In the causal inference literature, we can classify data into two categories. The one is data from a randomized controlled trial where the researcher or practitioner has full control of the selection process. The counterexample is data from a so-called observational study. In such a setting there are confounding variables that influence both, the outcome and the probability of treatment. To construct unbiased treatment effect estimates from observational studies, propensity score (PS) methods are an increasingly popular tool to control for confounding (Rosenbaum & Rubin, 1983). One model-based approach, the inverse probability of treatment weighting (IPTW), to directly adjust for the confounding bias using propensity scores was proposed by Hirano and Imbens (2001). Estimating the propensity score can be seen as a classification problem where one seeks to have a good prediction of the assignment probability given covariates, regardless of the functional form of the distribution of the probabilities. Besides logistic regression, non-parametric methods such as random forests (Lee et al., 2009; Westreich et al., 2010; Zhao et al., 2016), neural-networks, and support vector machines (Westreich et al., 2010) have been proposed to estimate the propensity score. An interesting question is which variables should be included to estimate the propensity score. Common suggestions are, to include all pre-treatment variables that influence the treatment while excluding variables that do not affect the treatment (this can be variables that do not influence any dependent variable - they are spurious, but also variables that are only predictive on the outcome). Following this rule, we would include confounding variables since they influence the treatment and the outcome as well as variables that only predict the treatment and not the outcome but exclude variables that are only predictive of the outcome.

We give an example of the different relationships between the covariates, the outcome, and the treatment in Figure 3.1. We denote X_t for covariates that predict only treatment but not the outcome, and X_o that predict the outcome but not the treatment probability. Of special interest are the confounding covariates X_c that we need to take into account to get an unbiased estimate of the average treatment effect (ATE) while X_s are spurious covariates that are uncorrelated to both, treatment and outcome. Let us illustrate the role of the variables using the vaccination against COVID 19 as an example. The treatment variable (D) is whether a person is vaccinated or not while the outcome variable (Y) is the individual probability of severe symptoms. Variable X_t could be the industry sector a person is working in since it influences the probability of being vaccinated but has no influence on symptoms. X_o could be whether a person smokes or not, which has an influence on the degree of symptoms but does not determine the vaccination probability. X_c might be the age of a person, which is associated with vaccination and symptoms while the variable body height might be unrelated and is classified as X_s.

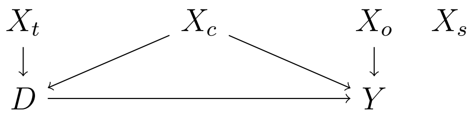


Figure 3.1: Dependencies of the covariates

The rational behind the classification of covariates is to perform variable selection when estimating the propensity score. First, and most important to ensure unbiased treatment effects in observational studies is to achieve unconfoundedness. In theory, it is sufficient to only use covariates X_c for the propensity score since the main analytic goal is to eliminate bias. This implies that we do not aim to explain the treatment assignment mechanism with high accuracy. If this would be the goal, variables X_t would need to be included. However, doing so could limit the overlap assumption. These are the two main reasons why we want to exclude covariates X_t in the propensity score model. Last, even if the dependency of X_o and D is zero in the true data generating process, the finite sample bias can be reduced if variables X_o are included in the propensity score model. The finite sample bias arises due to random confounding when the sample size is small.

In a recent paper by Shortreed and Ertefaie (2017), the authors suggest a different approach to get unbiased treatment effects from observational studies with a focus on decreasing the variance. Their proposed outcome-adaptive lasso (OAL) approach only selects features in the propensity score estimation that have a relationship with the outcome (X_c, X_o) but exclude variables that are predictive on the treatment (X_t) as well as spurious variables (X_s). To do so, they first find covariates that predict the outcome by regressing the outcome on the treatment and all covariates using a linear model. In the second step, the estimation of the propensity score, they use the lasso with an additional penalty term. The penalty contains individual weights for each covariate based on the coefficients of the covariates from the first step. The result is that the lasso excludes covariates that predict the treatment but are not related to the outcome as well as spurious covariates. Consider p covariates, denoted X_j for $j = 1, ..., p$, then the OAL estimator is defined as:

$$\begin{aligned}\widehat{\alpha}(OAL) = \underset{\beta}{\operatorname{argmin}} \Bigg[& \sum_{i=1}^{N} \left\{ -a_i \left(\mathbf{x}_\mathbf{i}^\top \alpha \right) + \log(1 + e^{\mathbf{x}_i^\top \alpha}) \right\} \\ & + \lambda_n \sum_{j=1}^{p} \widehat{\omega}_j |\alpha_j| \Bigg]\end{aligned} \tag{3.1}$$

where $\omega_j = |\tilde{\beta}_j|^{-\gamma}$ s.t. $\gamma > 1$. The vector $\tilde{\beta}_j$ refers to the coefficient estimates from regressing the covariates on the outcome, conditioning on the treatment.

An important limitation of the outcome-adaptive lasso is that this approach is restricted to parametric models. Both, the outcome model and the propensity score model have to be correctly specified. However, to control for selection bias we would like to have as many pre-treatment characteristics as possible to condition on them. This requirement lets new datasets easily become high-dimensional. In such settings, we face two potential challenges. First, the outcome and the treatment variable might not dependent linearly on the covariates. There can be interactions between variables and complex structures. Second, even if we could include such interactions, facing many covariates leads to the problem of which covariates to include in the outcome and propensity score model? It might be the case that only a few characteristics are important - the question is which are they? Such settings call for a non-parametric method that uses regularization. Keeping the idea of outcome-adaptive regularization but accounting for non-linearity and high-dimensionality, we propose the outcome-adaptive random forest (OARF). First, we estimate a modified and standardized variable importance score using a random forest used as the penalty weight. Second, we use the modified variable importance to regularize a random forest that learns the propensity score. To do so we penalize the gain at each split and propose the use of an initial feature space using the modified variable importance. This approach allows replacing both linear models, the regression (which we will refer to as OLS), and the lasso, with a random forest to estimate the ATE when the functions are non-linear or high-dimensional. We also make use of sample splitting to avoid overfitting and apply cross-fitting to restore efficiency. Our approach of a regularized random forest further selects only those covariates in the propensity score that are predictive of the outcome and excludes spurious variables. The OARF is designed to allow for categorical variables and is robust to different amounts of levels between categorical or continuous variables. The result is a flexible non-parametric method that allows unbiased estimation of the ATE while decreasing the variance. Our approach is also fast in computation (about 30 seconds for 2000 observations and 20 covariates).

3.2 Illustrating the outcome-adaptive estimation

Let us demonstrate the outcome-adaptive approach in a simple example. Assume the true outcome model as in equation 3.2. Y depends on the treatment D and linearly on three covariates (X_1, X_2, X_3). We also generate a propensity score model that includes variables X_1, X_2, X_5, and X_6 (see the Appendix for how the function is generated). The covariates are generated from a multivariate normal distribution and are independent. We set the sample size to $N = 1000$ and generate $p = 20$ covariates where only the first three are dependent on the outcome Y. Using the notation from Figure 3.1 we have the following structure: $X_1, X_2 \in X_c$, $X_3 \in X_o$ and $X_5, X_6 \in X_t$. Using a linear model (OLS), as in Shortreed and Ertefaie (2017), we want variables X_1 to X_3 to have the highest coefficients and penalize all others. Hence they should have a very small estimated coefficient. We run 500 Monte Carlo simulations and predict the treatment effect using the inverse probability of treatment weighted (IPTW) estimator (Hernán & Robins, 2006; Lunceford & Davidian, 2004). We use a logit model to estimate the propensity score, using all covariates (the full model) and only using covariates that are confounding or predict the outcome (target model). Figure 3.2 shows the estimated treatment effect using both models as well as the standardized and absolute coefficients from the OLS model. The target model (which can be seen as an oracle model) shows a smaller variance around the true treatment effect of 0.5 (indicated with the horizontal line). We also see that the OLS correctly assigns the highest coefficients to the variables that are predictors of the outcome while all other variables get smaller coefficients.

$$Y = 0.5D + 0.6X_1 + 0.6X_2 + 0.6X_3 + \varepsilon, \quad \varepsilon \sim N(0,1). \tag{3.2}$$

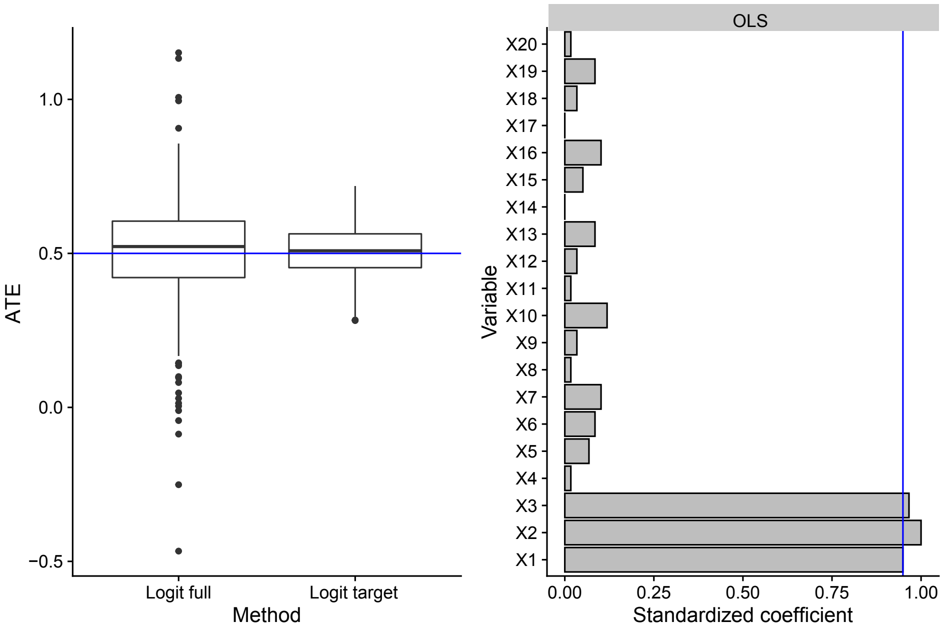


Figure 3.2: Left: IPTW estimates using full and oracle propensity score. Right: Selected variables from OLS.

The suggested approach by Shortreed and Ertefaie, 2017 to select only variables for the propensity score model that are predictive for the outcome works as expected if the underlying outcome and propensity score functions are linear. As soon as we introduce a more complicated function, the selection process in step 1 is biased. To demonstrate this, let us now assume the following true outcome model:

$$Y = 0.5D + 0.5X_1 + 0.8X_2 \otimes X_3 + \varepsilon, \quad \varepsilon \sim N(0,1). \tag{3.3}$$

Figure 3.3 shows the coefficients estimated by the OLS and the random forest, respectively. Values from the linear model are absolute coefficients while values from the random forest are based on the impurity measure which we explain below. The values for both methods are standardized between zero and one to make them comparable. Say we know that the model should select variables one to three. We draw a line indicating the lowest value from the variables that should be selected. If this would be a threshold (for selection or penalization), clearly the OLS estimates higher coefficients for at least 12 additional variables (besides the two with the highest coefficients). In contrast, the random forest does find the correct variables and assigns much lower importance to all other variables. Using the same threshold method on the random forest, all spurious variables would not be selected or at least heavier penalized compared to the OLS. The problem gets more severe if we not only allow for interaction terms but non-linear structures (say, by including quadratic and trigonometric functions into the data generating process).

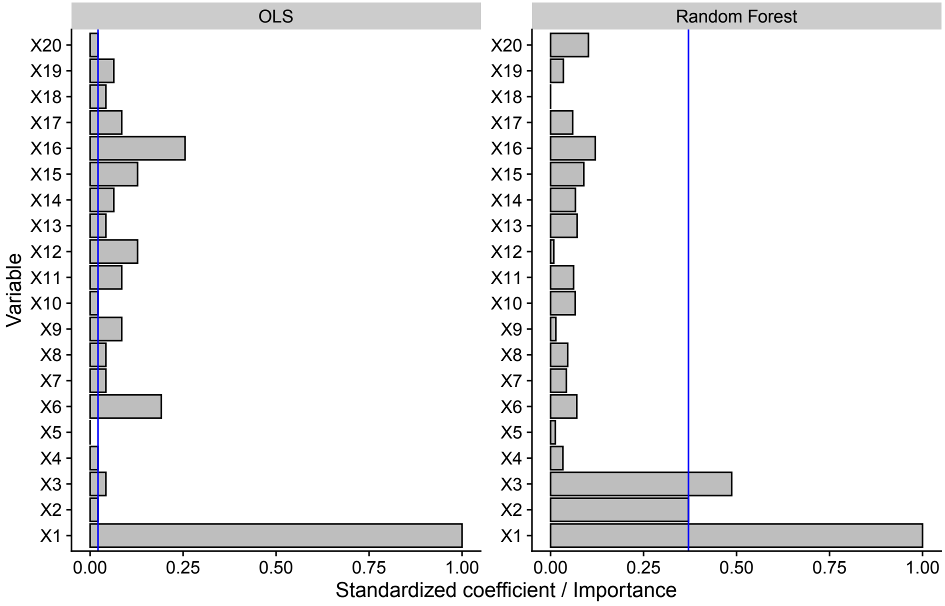


Figure 3.3: Standardized and absolute coefficient values for all covariates.

3.3 Method

Selecting variables using non-parametric models is not straightforward since we do not directly estimate coefficients for each feature like in the OLS. Using a random forest, we can instead create a variable importance measure to find the most predictive variables from the outcome model. Since we want to use a random forest for both steps, the outcome model and the propensity score, we have to replace the adaptive lasso by penalizing the tree-building mechanism. In this section, we show how to best estimate the variable importance measure from the outcome model and state the importance of the initial feature space. We describe in detail how the information about the covariates is then used to penalize the random forest that estimates the propensity score. This leads to a regularized version of the random forest that shrinks penalized variables to zero.

To ensure unbiased effects from a causal parameter the following assumptions from the potential outcome framework are required: Each observation has two potential outcomes, Y^1 if treated and Y^0 if not. We denote treatment by the binary indicator $D \in \{0;1\}$ and observed covariates $X \in \mathbb{R}^p$. See, for example Rubin (1980). First, ignorability: $(Y_i^1, Y_i^0) \perp\!\!\!\perp D_i|X_i$. It states that the treatment assignment is independent of the two potential outcomes. Second, the stable unit treatment value assumption (SUTVA), $Y_i = Y_i^0 + D_i(Y_i^1 - Y_i^0)$, ensures that there is no interference, no spillover effects, and no hidden variation. This means that the treatment status for individual i does not affect the potential outcomes of individual j. The third assumptions describes the propensity score: $P(D_i = 1|X_i = x) \overset{\text{def}}{=} e(x)$, which needs to be bounded away from zero and one: $\forall x \in supp(X_i), \quad 0 < P(D_i = 1|X_i = x) < 1,$. The last assumption states that the covariates are not affected by the treatment: $X_i^1 = X_i^0$.

We are interested in estimating the ATE (θ), assuming a partially linear model that takes the following form:

$$Y = \theta D + g(X) + \varepsilon, \quad \mathsf{E}[\varepsilon|D, X] = 0, \tag{3.4}$$

$$D = m(X) + \nu, \quad \mathsf{E}[\nu|X] = 0. \tag{3.5}$$

3.3.1 Variable Importance in Random Forest

Building a tree based on the classification and regression tree (CART) algorithm works as follows: A subset of the feature space $\mathcal{X}$ is selected at each node t in the tree. Internal nodes t are labelled with a split $s_t = (X_j < c)$, which creates at least two further subsets or children t_L and t_R (L and R refer to left and right, as in a tree). This procedure is repeated until we reach terminal nodes (or leaves) that are labeled with the best prediction of the outcome variable. In the regression case, this would be the mean value $\bar{y}_t$. The predicted output for a new instance is the mean value of the leaf reached by the instance when it is propagated through the tree. Using a recursive procedure at every note t, a tree identifies the split $s_t = s^*$ for which the partition of the sample into t_L and t_R maximizes the reduction of

$$\Delta(s,t) = \mathrm{cost}(\mathbb{D}_t) - \left(\frac{|\mathbb{D}_{t_L}|}{|\mathbb{D}_t|} \mathrm{cost}\left(\mathbb{D}_{t_L}\right) \right.$$
$$\left. + \frac{|\mathbb{D}_{t_R}|}{|\mathbb{D}_t|} \mathrm{cost}\left(\mathbb{D}_{t_R}\right) \right).$$

Let $\mathbb{D}$ be the set of observations for a specific decision or split. For the first step regression setting, we define the cost function as $\mathrm{cost}(\mathbb{D}) = \sum_{i \in \mathbb{D}} (y_i - \bar{y})^2$, where $\bar{y} = |\mathbb{D}|^{-1} \sum_{i \in \mathbb{D}} y_i$ is the mean of the outcome variable in the specified set or region. Maximising the decrease of the variance within each leaf can be seen as making the leafs pure in terms of the outcome values. Hence, the cost function is called the impurity measure.

The impurity change (or gain) through a split can be used to estimate the importance of a variable by evaluating each cost function given a specific feature j on which the split s^* is based. Hence, we can define the gain in terms of a feature j instead of split s: $\Delta(j,t) \overset{\mathrm{def}}{=} \Delta(s,t)$. The global importance value is given by accumulating the gain over a feature, $\Delta(j) = \sum_{t \in \mathbb{S}_j} \Delta(j,t)$ where $\mathbb{S}_j$ represents all the splitting points used in a single tree for the j-th feature. This is because a feature X_j can be used multiple times during the recursive growth of a tree. Since a random forest consists of B such trees, the importance value (Imp_j) is just an average over all trees:

$$Imp_j = \frac{1}{B} \sum_{b=1}^{B} \Delta(j)_b.$$

Nembrini et al. (2018) find that the impurity importance can be explained by two parts: The impurity reduction directly related to the true importance and a part of impurity reduction that is highly based on the structure of the feature (i.e. a dummy variable with only two levels vs. a continuous variable). The latter component introduces a bias in the impurity measure. To correct for the structure of different features, the authors propose to extend the feature space dimension p to $2p$. If the selected variable is within $\{1, ..., p\}$ then the variable is used as usual. If the variable is instead in the $\{p+1, ..., 2p\}$ set, the variable values are permuted. This means that the levels for each observation are reordered such that each observation has a different level as before. If the feature is reordered, the importance value adds negatively to the final measure and positively if the feature is untouched. This procedure, called actual impurity reduction (AIR), allows to de-bias the total importance value by controlling for the structure of each feature.

There are other measures for variable importance like permutation importance. To calculate the permutation importance of a feature, the prediction performance is calculated for observations that are not included in the bootstrap data (the so-called out-of-bag (OOB) observations). The values of the variable are then randomly permuted for each observation. Calculating the OOB error again with the permuted feature indicates the importance of the variable. The more the prediction error changes, the more important is the feature. The permutation performance is calculated for each feature and averaged over all trees. See Nembrini et al. (2018) for an overview and comparison of different importance measures. We note that the permutation method is computationally expensive when compared with the AIR method. This is because the former method relies on OOB predictions for each feature. In terms of robustness concerning the different amounts of levels between variables, they perform similarly.

In our setting, we have at least one binary variable, the treatment assignment, which has a limited range of splitting values compared to continuous variables. Therefore, it is important to take the different structures into account by applying the AIR measure (available in the `ranger` package as `'impurity corrected'`).

3.3.2 Penalization parameter

The penalization parameter λ_j should depend on the predictive power from the covariates on the outcome. Our measure of predictive power is the importance score Imp_j^* which can be included as proposed by Deng and Runger (2013):

$$\lambda_j = (1 - \gamma)\lambda_0 + \gamma Imp_j^*, \tag{3.6}$$

where λ_0 is a general penalization parameter and γ is a weight parameter that determines the proportion of general and specific feature penalization. Next, we define the normalized importance score (Imp_j^*) as

$$Imp_j' = \begin{cases} Imp_j \text{ if } Imp_j \geq 0 \\ 0, \text{otherwise} \end{cases} \tag{3.7}$$

$$Imp_j^* = \frac{Imp_j'}{max_{l=1}^{P} Imp_l'} \tag{3.8}$$

The new normalized importance score is scaled within the interval $[0,1]$ (see the Appendix for the proof). Equation 3.7 sets all negative importance scores from the AIR measure to zero. Since we only want to rely on the importance values obtained from the outcome model, we set $\lambda_0 = 0$ and $\gamma = 1$. This allows for the heaviest penalization based on outcome-related covariates, namely

$$\lambda_j^y = Imp_j^*. \tag{3.9}$$

3.3.3 Creating the feature space

Next, we want to use the information obtained in the first step and only use variables that have a high importance on the outcome to estimate the propensity score. Again, we want to use a random forest since the propensity score function can have a similar non-linear structure as the outcome model. To guide the feature selection, Deng and Runger (2012) introduce a (guided) regularized random forest (RRF) by proposing to weight the gains of the splits during the recursive procedure. As a result of the random feature selection at each split, after m splits, only a subset $\mathbb{F}$ of features are included in the tree. To limit the feature space the idea is to exclude variables not belonging to $\mathbb{F}$, unless their importance is substantially larger than the maximum of the gain for features already included. Deng and Runger (2013) define the regularized gain as

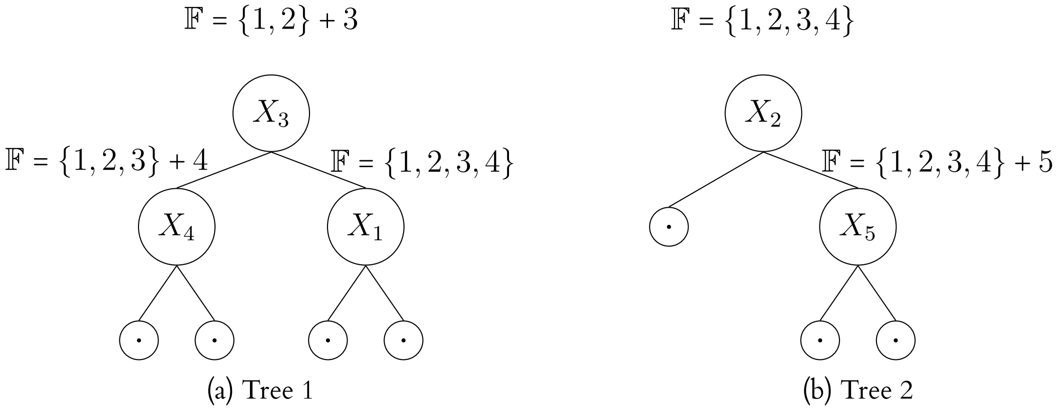
$$\mathrm{Gain}_R\left(\mathbf{X}_j, t\right) = \begin{cases} \lambda_j^y \Delta_\nu(j,t), X_j \notin \mathbb{F} \text{ and} \\ \Delta_\nu(j,t), X_j \in \mathbb{F} \end{cases} \tag{3.10}$$

where $\lambda_j \in (0,1]$ is the penalty coefficient that controls the gain for each feature j if this feature was not previously used for a split. Originally, the feature space $\mathbb{F}$ is an empty set at the root note in the first tree. Only if a feature adds enough predictive information it is included. Based on equation 3.10, the smaller the value for λ_j, the higher the penalty and hence the less likely it is for feature j to be included in the subset. In our setting, we want to include features that may not be that predictive of the treatment but of the outcome. We also want to make sure that the important features are used in the splitting process (at least with a higher probability). Therefore, we already include features in $\mathbb{F}$ that fulfill the following criterion:

$$\kappa_j = \mathbb{1}(Imp_j \geq \frac{1}{P}\sum_{j=1}^{P} Imp_j) \tag{3.11}$$

$$\mathbb{F} = \{\kappa_1 * X_1, \kappa_2 * X_2, ..., \kappa_P * X_P\} \tag{3.12}$$

If the importance score X_j is at least the mean over all importance scores, we include the variable in the initial feature space and drop values that are zero in $\mathbb{F}$. Figure 3.4 illustrates the guided feature selection process. We start with a non-empty set (here variables 1 and 2 are included). The first split is based on variable 3. If this variable is not in the feature space, the gain is multiplied by λ_3. If the penalized gain is higher than the gain from the parent the feature is included. As an illustration, we always move from top to bottom and from left to right. The next split is then on X_4, again if the penalized gain difference is positive, the feature is included in $\mathbb{F}$. Next, a split on X_1 is made, since the variable is already in the feature space the gain is not penalized. Still, it has to be higher than the gain from the parent node to keep the split. Building the first tree, we end up with a feature space containing four variables. The information of the feature space is now used to build further trees. In tree 2, we start with the initial features space as extracted from tree 1. Note that this procedure does not allow to grow trees in parallel since each tree needs the information from the former tree about the feature space to determine if the gain from variables should be penalized or not.



(a) Tree 1

(b) Tree 2

Figure 3.4: Initial feature space based on guided regularized random forest (GRRF)

The covariate selection process does not depend on how often a variable can be split and hence if the variable is continuous or, for example, binary. It is sufficient if the gain from one split is above the threshold. Different from calculating the variable importance using the impurity measure we do not average over all split within a tree. This allows the covariates to be of any form, such as binary, categorical or continuous.

3.3.4 Sample splitting

To avoid overfitting, which can easily happen when using flexible methods such as random forest, we make use of sample splitting and cross-fitting. First, we split our sample into two equal parts, I_A denotes the auxiliary sample and I_E is the estimation sample. We first use the subset I_A to train the propensity score function and I_E to estimate the treatment effect using the predicted propensity score in the IPTW estimator. Let us denote the resulting estimator as $\hat{\theta}(I_A, I_E)$. Now we switch the roles of the auxiliary and estimation sample to obtain a second estimator, called $\hat{\theta}(I_E, I_A)$. Since both estimators are estimated on only a subset of the observations there might be an efficiency loss. Cross-fitting, which was recently introduced by Chernozhukov et al. (2018), aims to restore efficiency by simply averaging the two estimates:

$$\tilde{\theta} = \frac{1}{2}\left\{\hat{\theta}(I_A, I_E) + \hat{\theta}(I_E, I_A)\right\} \tag{3.13}$$

This approach generalizes to K folds where I_A contains $K - 1$ folds and I_E the remaining fold. Similar to cross-validation, each fold is used to estimate the ATE by iteratively looping through the folds. The final estimator is the average over the K estimators.

We use the full sample to get the variable importance from the first step. This is especially helpful if the sample size is small. When using a logit model or the lasso as a benchmark, we also use the full sample and estimate the final treatment effect in one step.

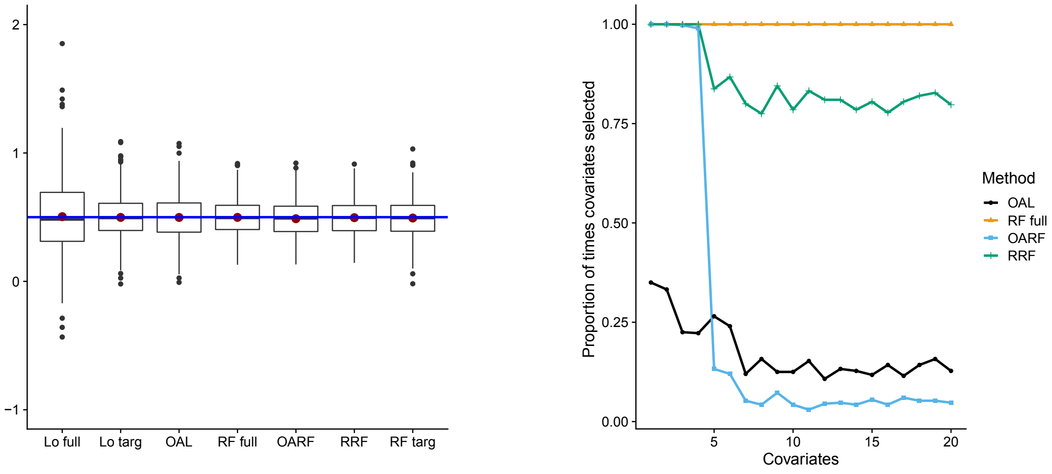
3.4 Simulation study

To evaluate the performance of our **OARF** method in more detail, we consider different data generating processes (DGP's) and consider the following methods for comparison and benchmarking: The **OAL** method by Shortreed and Ertefaie (2017) and two generalized linear models. The first one uses all covariates in a logit model (**Lo full**) while the second one only uses target variables (X_c, X_o) to estimate the propensity score (**Lo targ**). We use the same benchmarks for the random forest, denoted by **RF full** and **RF targ**. We also use the regularized random forest (**RRF**) which only sets a penalty based on the first step variable importance but does not make use of an initial feature set, as proposed here by the OARF. We use the following order of variables when we look at the variable selection plots: X_c, X_o, X_t. If the amount of the variables are set to two, then the first two are confounders, variable three and four are regressors on the outcome and five and six are regressors on the treatment. We use the `ranger` package in `R` for all estimations based on the RF and the `RRF` package by Deng (2013) in part for the OARF. The OAL approach is based on the replication file from Shortreed and Ertefaie (2017). The tuning parameter λ and γ for the OAL method are selected using a weighted absolute mean difference (wAMD) which we describe in the Appendix.

First, we consider two linear settings and generate a binary treatment, D, from a Bernoulli distribution with $logit\{P(D = 1)\} = \sum_{j=1}^{p} \nu_j X_j$ and the continuous outcome variable Y as $Y = \theta D + \sum_{j=1}^{p} \beta_j X_j + \varepsilon$, where $\varepsilon \sim N(0,1)$ and $\theta = 0.5$. The two settings differ in the strength of the confounding effect. Setting 1 sets $\beta = (0.6, 0.6, 0.6, 0.6, 0, 0, 0, ..., 0)$ and $\nu = (1, 1, 0, 0, 1, 1, 0, ..., 0)$, and setting 2 sets $\beta = (0.6, 0.6, 0.6, 0.6, 0, 0, 0, ..., 0)$ and $\nu = (0.4, 0.4, 0, 0, 1, 1, 0, ..., 0)$. Setting 2 hence has a weaker confounding relationship than setting 1. These two settings are identical to the one used in Shortreed and Ertefaie (2017). Setting 3 aims to have a non-linear relationship between the covariates and the outcome but the same linear structure for the propensity score as in setting 1. Setting 3 is generated as follows: $Y = \theta D + 0.8(X_1 \otimes X_2) + 0.8(X_3 \otimes X_4) + \varepsilon$; $logit\{P(D = 1)\} = \sum_{j=1}^{p} \nu_j X_j$, with $\nu = (1, 1, 0, 0, 1, 1, 0, ..., 0)$. In all three setting we set $N = 500$ and $p = 20$. We find that our OARF performs similar to the benchmark OAL method. The random forest using all covariates and the one that uses the penalization weights from the first step perform worse and are clearly biased. The reason for the bias might be that the full random forest has a higher chance to select variables X_t and X_s than OARF. We do see an improvement in terms of bias when using the RRF, which selects fewer of the above mentioned variables due to the penalization weights. We also find that if we weaken the confounding relationship, all methods have a smaller variance and the RF methods a smaller bias. Boxplots illustrating the IPTW estimator using different methods are shown in Figure 3.8. Selected covariates from the propensity score model are shown in Figure 3.12. In both settings, the OARF approach only selects the desired features. This is similar to the OAL method. For comparison, we show that the full RF selects all features and the RRF a higher proportion of all variables while always selecting the desired variables.

In setting 3, illustrated in Figure 3.5, all approaches are unbiased since the propensity score function depends linearly on the covariates. Only the outcome model is non-linear which is why the variance is higher in the linear models compared to the random forest. Next to the treatment effect estimates, the selected variables are illustrated. The OAL model fails to select the correct features as was expected based on the coefficient values from Figure 3.3. The RF without regularization uses all variables while the RRF always selects the correct four features but often (in about 80%) all other variables. Only the OARF selects the correct features and drops the covariates that are not of interest.

Next, we generate data processes where both functions are non-linear (settings 4 to 10). These are the settings where we would expect the OARF to perform superior. A complete list of all the DGP's is shown in Table 4.2. In those settings, we set the sample size to

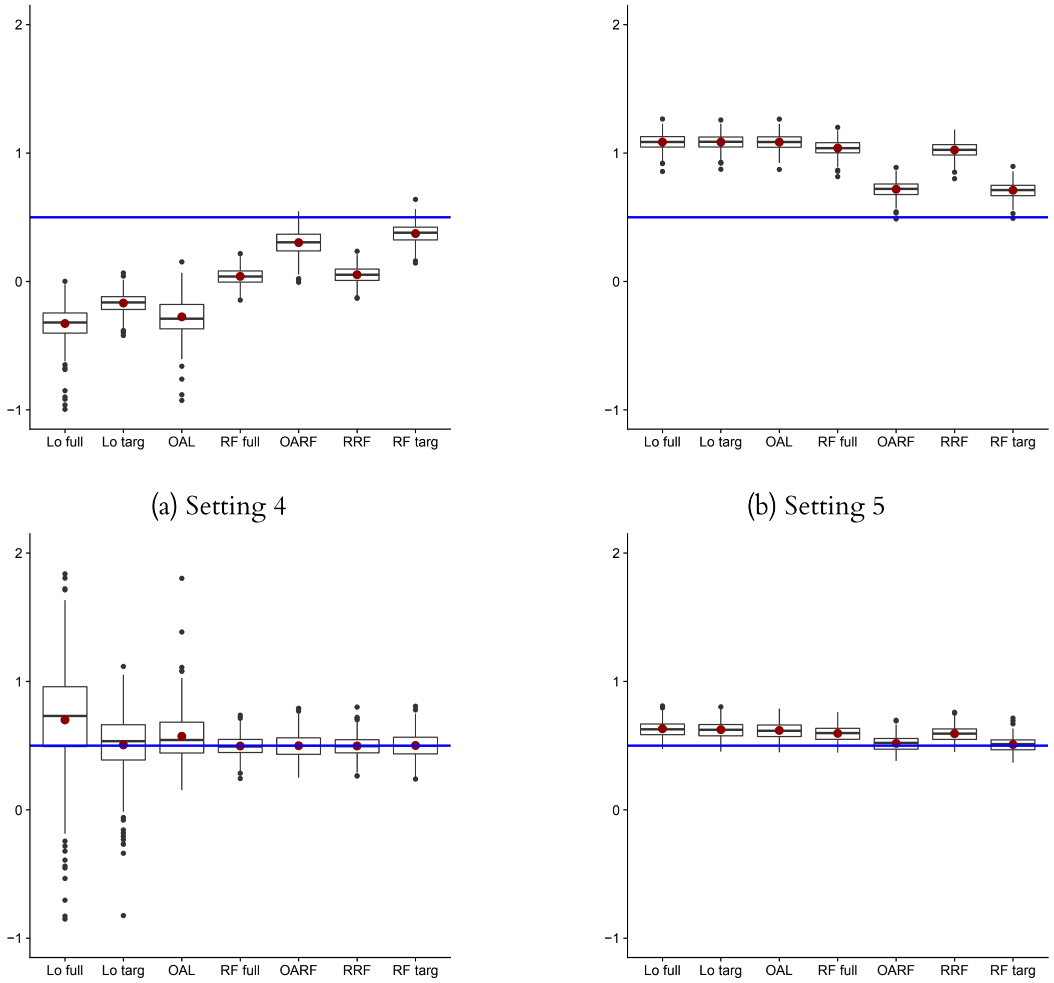


(a) Treatment effects

(b) Selected variables

Figure 3.5: Illustration with non-linear outcome model and linear propensity score model (Setting 3).

$N = 2000$. Our results are shown for $p = 20$ to allow a noticeable visualization. Varying the number of covariates to 50, 100, and 200 does not change the results of the ATE estimates nor the selection of the correct covariates. In setting 4 to 7, we keep the function on the outcome model and only change the propensity score function. Setting 1 to 7 sets the amount of variables for X_c, X_o, and X_t to 2 while the amount of spurious variables $X_s = p - (X_c + X_o + X_t)$. Setting 8 to 10 uses $X_c = 6, X_o = X_t = 2$ covariates and again the remaining set for X_s. The favourable covariates to select are X_1 to X_4 for settings 1 to 7 and X_1 to X_8 for setting 8 to 10. Figure 3.6 shows ATE estimates for settings where all functions are non-linear and potentially complicated while Figure 3.9 shows results for similar functions but with more depending covariates. The proportion of selected covariates over all simulations are illustrated in Figure 3.12 and Figure 3.13. Overall, we find that the OARF performs best and is close to the RF that only uses X_c and X_o (RF targ). Setting 4 and 5 show that all methods are biased while the OARF is closed to the true ATE. In setting 4, all methods are downward biased, which leads to a negative estimate using the OAL. Only the RF approaches estimate the correct sign of the treatment effect. In setting 6, the linear methods are slightly upward biased while the RF approaches show no bias. Setting 7 shows a similar effect of bias where only the OARF estimates unbiased treatment effects. Using more covariates allows making the functions even more complicated since the flexibility can be increased. Setting 8 and 10 again show some bias for the linear methods as well as a higher variance compared to the RF approaches. Setting 9 is comparable to setting 4 in the sense that all methods are downward biased. In this setting, even the OARF is biased and shows a higher variance. We notice that in some settings there is not so much difference between the OARF and the full RF. The advantage of the OARF remains since it achieves the same accuracy using fewer variables as the full RF. In all settings that have at least one non-linear function, only the OARF selects the correct features with high precision. The full RF selects all covariates and the RRF uses unnecessary covariates in about 80% of the cases.



(a) Setting 4

(b) Setting 5

(c) Setting 6

(d) Setting 7

Figure 3.6: Illustration with non-linear functions for both models.

Introducing a correlation between the covariates increases the bias and variance. We investigate the effect in a linear setting with moderate and heavy correlation. The results are shown in Figure 3.10. If the correlation is moderate ($\rho = 0.2$) the OARF is still unbiased while all other methods show a slight bias and an increase in variance. It is reassuring that the random forest can find the correct importance score even with moderate correlation among the variables. If we introduce a heavy correlation ($\rho = 0.5$) also the random forest approaches are biased while the variance in the linear models increases heavily. The OARF is still closest to the target method. The variable selection is still correct as illustrated in Figure 3.14. Last, we show results when the outcome function is more complicated and non-linear. Boxplots that illustrate the variance over 500 repetitions are shown in Figure 3.11 while the corresponding variable coverage rates are shown in Figure 3.15. Using a more complex outcome function increases the bias for the linear methods. Even the RRF shows a high bias in setting 13 and 14. The OARF is closest to the oracle RF. Setting 15 does show an equally small bias among all methods, again with the OARF and the oracle RF closest to the true treatment effect. The coverage rates regarding the selected variables show a similar picture as for all other settings. The vanilla RF does select all variables, the RRF selects the four correct variables in almost 100% of the cases but also all other variables in more than 75% of the cases. The OAL is not able to select the correct variables (e.g. variable 4 is selected in only 12.5% of the cases. For setting 13 and 14, the OARF selects the desired variables in 100% of the cases and drops all other variables with the same accuracy. Only in setting 15, one variable (X_2) is only selected around 37% of the time while all other variables are selected as desired.

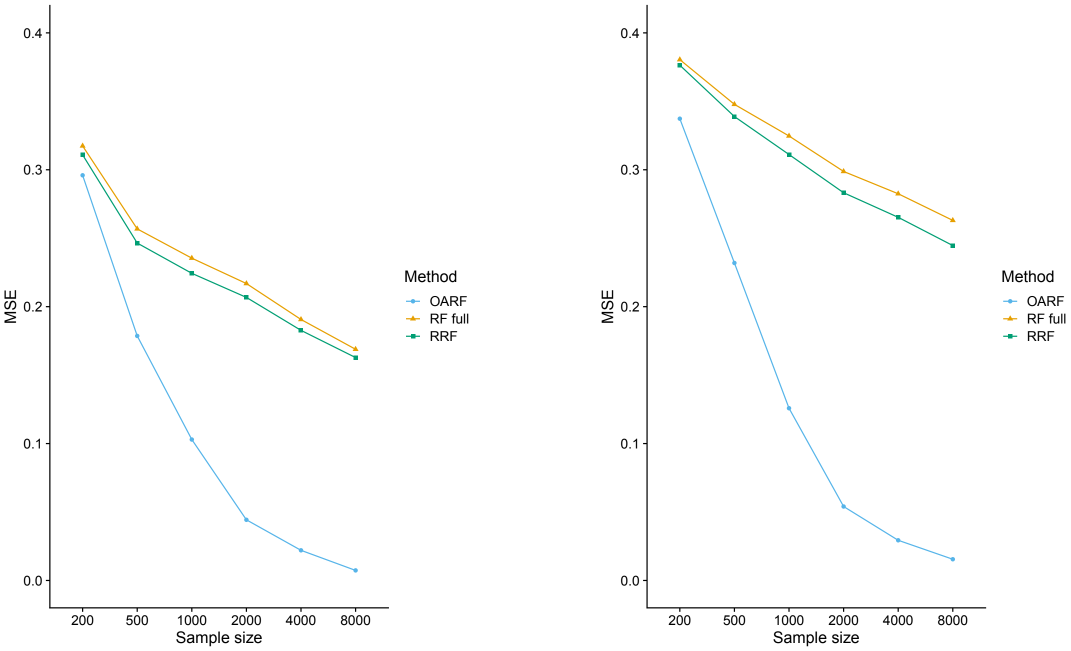
In Table 3.1 we show coverage rates of 95% confidence intervals and the width of the interval (in parentheses). Confidence intervals for IPW were constructed using a percentile-based nonparametric bootstrap. For the OAL method, we use a smoothed non-parametric bootstrap approach that takes the model selection procedure into account. This procedure is described in Efron (2014). The confidence intervals for all RF approaches are based on non-parametric bootstrapping. We use 500 bootstrap samples for each method. We then take the 0.025 ($\alpha/2$) and 0.975 ($1 - \alpha/2$) quantile from the empirical distribution as the lower and upper bound for the confidence interval. We apply three non-parametric versions, the **RF full** (without regularization), the **RRF** (which uses regularization but no initial feature set), and our proposed **OARF**. We also apply the **IPW** method using a linear model to estimate the propensity score and the **OAL** method using the outcome-adaptive lasso to estimate the propensity score. We notice that the width of the confidence interval for the OARF is smaller than for the OAL method (in some settings only half as wide). The vanilla RF and the RRF do not achieve a coverage rate of 95% for any settings, while the OARF achieve the rate in 8 out of the 15 settings. The results show that some data generating processes might be too complex and hence are biased for the IPTW estimator. This is why increasing the sample size does not lead to a higher coverage rate. In all other settings, we see an increase in the coverage rate and a decrease in the width of the confidence intervals when increasing the sample size from N=500 to N=2000.

The OARF approach does not only decrease the variance when increasing the sample size but also the bias. Figure 3.7 shows the mean squared error over 400 Monte Carlo replications for six different samples sizes (from 200 to 8000 observations). Here we use setting 4 and 5 as the data generating process since those settings have a low coverage rate of the confidence intervals. We compare the three random forest approaches, the full version, the regularized, and the OARF. The results show that the OARF achieves a significant decrease in MSE when increasing the sample size (e.g. for setting 5: from MSE = 0.35 for 200 observations, to MSE = 0.03 for 8000). The other two algorithms always have a higher MSE and the decrease is only slightly. The main reason for the high decrease in MSE when

Table 3.1: Coverage rates and width for 95% confidence intervals

Setting / Estimator	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
N = 500															
IPW	0.81 (0.98)	0.96 (0.58)	0.94 (1.07)	0.05 (0.85)	0.00 (0.53)	0.84 (1.90)	0.84 (0.55)	0.95 (2.59)	0.10 (0.91)	0.91 (1.28)	0.65 (1.09)	0.94 (0.55)	0.10 (1.43)	0.07 (1.16)	0.52 (0.39)
OAL	0.97 (0.72)	0.99 (0.59)	1.00 (1.01)	0.18 (1.00)	0.00 (0.74)	1.00 (1.33)	0.97 (0.76)	0.99 (1.38)	0.14 (1.11)	0.99 (1.23)	0.98 (0.71)	0.99 (0.58)	0.02 (1.32)	0.04 (1.31)	0.89 (0.67)
RF full	0.00 (0.46)	0.04 (0.45)	0.93 (0.49)	0.04 (0.49)	0.00 (0.47)	0.94 (0.57)	0.82(0.48)	0.93 (0.62)	0.04 (0.65)	0.89 (0.62)	0.00 (0.45)	0.00 (0.45)	0.00 (0.82)	0.00 (0.44)	0.38 (0.35)
RRF	0.00 (0.46)	0.04 (0.45	0.94 (0.49)	0.05 (0.49)	0.00 (0.47)	0.94 (0.58)	0.83 (0.48)	0.92 (0.63)	0.04 (0.65)	0.89 (0.62)	0.00 (0.46)	0.00 (0.45)	0.00 (0.83)	0.00 (0.44)	0.38 (0.35)
OARF	0.83 (0.67)	0.92 (0.58)	0.96 (0.51)	0.13 (0.51)	0.03 (0.48)	0.97 (0.66)	0.87 (0.48)	0.95 (0.62)	0.07 (0.70)	0.92 (0.66)	0.82 (0.68)	0.93 (0.55)	0.23 (1.02)	0.63 (0.75)	0.71 (0.36)
N = 2000															
IPW	0.54 (0.49)	0.96 (0.25)	0.89 (0.51)	0.05 (0.43)	0.00 (0.26)	0.68 (1.20)	0.48 (0.27)	0.95 (1.28)	0.00 (0.43)	0.94 (0.66)	0.5 (0.48)	0.92 (0.26)	0.12 (0.63)	0.04 (0.90)	0.19 (0.28)
OAL	1.00 (0.59)	1.00 (0.47)	1.00 (0.91)	0.20 (1.15)	0.00 (0.58)	1.00 (1.72)	1.00 (0.60)	1.00 (0.83)	0.00 (0.97)	1.00 (1.19)	1.00 (0.59)	1.00 (0.47)	0.01 (0.95)	0.06 (1.29)	0.95 (0.62)
RF full	0.20 (0.23)	0.00 (0.22)	0.93 (0.24)	0.00 (0.24)	0.00 (0.23)	0.92 (0.29)	0.55 (0.24)	0.93 (0.31)	0.00 (0.32)	0.88 (0.31)	0.00 (0.23)	0.00 (0.22)	0.00 (0.45)	0.00 (0.32)	0.12 (0.24)
RRF	0.14 (0.23)	0.00 (0.22)	0.93 (0.24)	0.00 (0.24)	0.00 (0.23)	0.92 (0.29)	0.55 (0.24)	0.93 (0.31)	0.00 (0.32)	0.88 (0.31)	0.00 (0.23)	0.00 (0.22)	0.00 (0.45)	0.00 (0.32)	0.12 (0.24)
OARF	0.96 (0.38)	0.95 (0.30)	0.98 (0.33)	0.14 (0.31)	0.00 (0.25)	0.99 (0.42)	0.94 (0.25)	0.97 (0.22)	0.00 (0.33)	0.95 (0.33)	0.85 (0.38)	0.95 (0.29)	0.25 (0.62)	0.59 (0.55)	0.78 (0.27)

using the OARF is both, a decrease in variance and bias. To visualize this result we show boxplots in Figure 3.16 and 3.17 in the Appendix. They show the ATE estimation using the three random forest algorithms under varying sample sizes for each of the 400 Monte Carlo iterations. While the variance decreases in all algorithms when increasing the sample size, only the OARF has a noticeable decrease in bias.



(a) Setting 4

(b) Setting 5

Figure 3.7: MSE given different sample sizes.

In the Appendix, we discuss the possibility of tuning certain parameters in the random forest while keeping the objective to get balanced covariates rather than maximize the prediction accuracy. We also discuss a generalization of the OARF to other methods like the double machine learning, introduced by Chernozhukov et al. (2018). We find that using the OARF to estimate the propensity score decreases the bias compared to a full RF. In settings where even the OARF is biased, we additionally use the OARF to estimate the conditional mean of Y. This approach further decreases the bias.

3.5 Empirical Examples

In this section, we revisit two empirical examples. Both datasets are observational and contain a rich set of characteristics that may lead to selection into treatment. We use the same 5 methods as in the simulation study above. We report the ATE and a 95% confidence interval (CI). The datasets we use are freely accessible: RHC (https://hbiostat.org/data/), birth weight (http://www.stata-press.com/data/r13/cattaneo2.dta).

3.5.1 Re-analysis of SUPPORT data on Right Heart Catheterization

Right heart catheterization (RHC) is a diagnostic procedure used for critically ill patients. Connors et al. (1996)used a propensity score matching approach to study the effectiveness of right heart catheterization in an observational setting. The authors found that after controlling for selection bias by using a rich set of covariates, RHC appeared to lead to higher mortality than not performing RHC. This conclusion is in contrast to popular belief among practitioners that RHC was beneficial.

The SUPPORT study collected data on hospitalized adult patients at five medical centers in the US. Based on information from a panel of experts, a rich set of characteristics, believed to be related to the decision on whether to perform the right heart catheterization or not, was collected. The study consists of data on 5735 individuals, RHC was performed on 2184 (the treatment group) while the remaining 3551 individuals did not get RHC (the control group). Treatment is equal to 1 if right heart catheterization was applied within 24 hours of admission, and 0 otherwise. The outcome of interest is an indicator for survival at 30 days. In total, we observe 72 covariates (including many dummy variables). After excluding variables with more than 50% missing values, we have 68 covariates left.

Connors et al. (1996) matched treated and untreated patients based on the propensity score, with each unit, matched at most once. Hirano and Imbens (2001) use different matching estimators (one of them is the exact matching) and their regression adjusted estimates range from $(-0.062, -0.053)$. Crump et al. (2009) apply different samples based on the propensity score overlap and estimate an ATE from -0.059 to -0.060. Ramsahai et al. (2011) use propensity score and generic matching to investigate the effect of RHC on mortality within 180 days (note that the outcome variable is now an indicator for mortality, not survival). Their results are consistent with the previous findings, namely an estimated ATE using propensity score matching of 0.063 and 0.046 when using generic matching. A more recent study by Li et al. (2018) considers different weighting strategies for covariate balancing. These strategies are based on the propensity score to estimate different target parameters. Their results also confirm previous estimates. For example, they estimate an ATE using overlap weighting of -0.065 and -0.067 when using optimal matching. All methods mentioned above use a logit model to estimate the propensity score. Given the rich set of covariates that consists of characteristics like age, race, income, and medical characteristics, there might be interaction effects and non-linear dependence. It would also be useful to know which covariates are true confounders and are selected when we do not assume any parametric form for both, the outcome model and the propensity score model.

As shown in Table 3.2, all methods estimate a negative treatment effect, suggesting that performing RHC does decrease survival within the first 30 days. The results from the IPW and OAL are consistent with results from previous findings where the propensity score is estimated via a logit model. The RF results are smaller in magnitude with the lowest ATE of −0.041 by the OARF. The confidence intervals for the RF methods are tighter compared to the IPW or OAL method, while the latter method even includes zero. Among the 68 covariates, we find that the full RF selects the same variables as the RRF, in total 42 covariates with a proportion of 90 – 100%. The OAL method selects only 3 variables and 6 variables around 50% of the time (proportion between 50 – 60%). The OARF selects 9 variables in almost all iterations. All variables selected by OAL are also selected by OARF and additional variables are age, Duke Activity Status Index (DASI), APACHE score, Glasgow Coma Score (scoma1), white blood cell count Day 1 (wblc1), and bilirubin Day 1 (bili1). Variables that are not selected by OAL and OARF but with the RF and RRF are, among others, sex, race, education, and income. Especially characteristic variables that might be uncorrelated with a person's well being are not selected by the OARF but are still selected using the RRF using the same weights as the OARF. The complete list of covariates and their inclusion proportion is listed in Table 3.7. Compared to OAL, the OARF includes age, DASI, white blood cells, and three other covariates.

Table 3.2: Estimates for average treatment effects in RHC study

Method	Estimate	95% CI
IPW	-0.055	(-0.087, -0.026)
OAL	-0.057	(-0.158, +0.031)
RF full	-0.045	(-0.074, -0.026)
RRF	-0.045	(-0.074, -0.027)
OARF	-0.041	(-0.071, -0.022)

Table 3.3: Covariate selection (at least 90%)

Method	# Covariates	excluded covariates
OAL	3	e.g. sex, race, education, income
RF full	42	e.g. trauma, rental, hema
RRF	42	e.g. trauma, rental, hema
OARF	9	e.g. sex, race, education, income

3.5.2 Effect of smoking on birth weight

In this example, we reinvestigate the effect of maternal smoking status during pregnancy, the treatment variable, on babies' birth weight, the outcome variable. We use a publicly available dataset that consists of 4642 singleton births in the USA. Additionally, we observe a rich set of characteristics like age, marital status, race, education, number of prenatal care visits, months since last birth, an indicator of firstborn infant, and indicator of alcohol consumption during pregnancy. All covariates are for the mother, except education, which we also observe for the father. The full dataset, containing more observations, was first used by Almond et al. (2005) who found a strong negative effect of maternal smoking during pregnancy on the weights of babies (about 200 – 250 gram lighter for a baby with a mother smoking). In their study, the authors use a logit model to estimate the propensity score. We focus on estimating the propensity score without assuming any parametric form and again base the variable selection on features that are associated with the outcome. Table 3.4 shows the ATE estimates along with 95% CI's. We find similar negative effects as Almond et al. (2005). The OARF estimates a decreased birth weight of -224 grams. Compared to the other RF models and the OAL method, the OARF has the tightest confidence intervals. Only the classic IPW estimator has a slightly tighter upper bound.

The OAL model and the full RF include 17 out of 19 variables while the RRF model includes 16 covariates. The OARF only includes 7 covariates and excludes, for example, an indicator for marital status, the education of mother and father, and if there were prenatal visits. The excluded variables are shown in Table 3.5 and the complete list with inclusion probability is shown in Table 3.8. The OARF is the only method that excludes the variable alcohol, meaning that it is not a confounding variable nor is it predictive on the outcome. The results, from a recent study by Lundsberg et al. (2015), suggest low-to-moderate alcohol exposure during early and late gestation is not associated with increased risk of low birth weight. Such findings are quite interesting since they allow a better understanding and interpretation of true confounding variables and of such that are not informative.

Table 3.4: Estimates for average treatment effects in birth weight study

Method	Estimate	95% CI
IPW	-236	(-286,-187)
OAL	-236	(-357,-115)
RF full	-221	(-287,-162)
RRF	-205	(-347,-83)
OARF	-224	(-286,-165)

Table 3.5: Covariate selection (at least 90%)

Method	# Covariates	excluded covariates
OAL	17	mother age, birth month
RF full	17	mother hispanic, foreign,
RRF	16	mother hispanic, father hispanic, foreign,
OARF	7	all the above and married, alcohol, m. and f. education, first baby, prenatal visit

3.6 Discussion

We propose a non-parametric variable selection procedure for the estimation of treatment effects from observational studies. Building on outcome-adaptive penalization, we use a random forest to define variables that are predictive of the outcome. This allows the outcome model to deviate from a linear dependence on the covariates. We use a modified variable importance score to generate coefficients from the outcome model. We use the importance score to penalize variables that are spurious or that only predict the treatment but not the outcome. We show how to use penalty weights for each covariate to regularize the random forest that estimates the propensity score. Additional to the penalty, we show the importance of an initial feature space and how to include it in the RF. A Monte Carlo simulation shows that our proposed method, the OARF, has a smaller variance and produces unbiased estimates. In cases where all estimators are biased, the OARF produces the smallest bias and variance. The second goal of our proposed approach is to select only variables that have a relationship with the outcome (including all confounding variables) while excluding variables that predict the treatment and unimportant variables. Based on the simulation, we find that only the OARF selects the correct covariates and disregards all others. This holds for linear and non-linear settings and even if there is a strong correlation between the variables.

We apply the OARF and all other benchmark methods in two empirical examples. The ATE is comparable between all methods while the OARF shows tighter confidence intervals compared to the OAL and other RF methods. Regarding the variable selection, we find that the OARF selects and drops different variables compared to all other methods. This allows for a detailed evaluation of which variable might be responsible for selection bias and which variables are just spurious or only affect the propensity score.

Bibliography

Almond, D., Chay, K. Y., & Lee, D. S. (2005). The costs of low birth weight. *The Quarterly Journal of Economics*, *120*(3), 1031–1083. https://doi.org/10.3386/w10552

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, *21*(1), C1–C68. https://doi.org/10.1111/ectj.12097

Connors, A. F., Speroff, T., Dawson, N. V., Thomas, C., Harrell, F. E., Wagner, D., Desbiens, N., Goldman, L., Wu, A. W., Califf, R. M., et al. (1996). The effectiveness of right heart catheterization in the initial care of critically iii patients. *Journal of the*

American Medical Association, *276*(11), 889–897. https://doi.org/10.1001/jama.1996.03540110043030

Crump, R. K., Hotz, V. J., Imbens, G. W., & Mitnik, O. A. (2009). Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, *96*(1), 187–199. https://doi.org/10.1093/biomet/asn055

Deng, H. (2013). Guided random forest in the rrf package. *arXiv preprint arXiv:1306.0237*.

Deng, H., & Runger, G. Feature selection via regularized trees. In: *The 2012 international joint conference on neural networks (ijcnn)*. IEEE. 2012, 1–8. https://doi.org/10.1109/ijcnn.2012.6252640.

Deng, H., & Runger, G. (2013). Gene selection with guided regularized random forest. *Pattern Recognition*, *46*(12), 3483–3489. https://doi.org/10.1016/j.patcog.2013.05.018

Efron, B. (2014). Estimation and accuracy after model selection. *Journal of the American Statistical Association*, *109*(507), 991–1007. https://doi.org/10.1080/01621459.2013.823775

Hernán, M. A., & Robins, J. M. (2006). Estimating causal effects from epidemiological data. *Journal of Epidemiology & Community Health*, *60*(7), 578–586. https://doi.org/10.1136/jech.2004.029496

Hirano, K., & Imbens, G. W. (2001). Estimation of causal effects using propensity score weighting: An application to data on right heart catheterization. *Health Services and Outcomes research methodology*, *2*(3), 259–278. https://doi.org/10.1023/A:1020371312283

Lee, B. K., Lessler, J., & Stuart, E. A. (2009). Improving propensity score weighting using machine learning. *Statistics in Medicine*, *29*(3), 337–346. https://doi.org/10.1002/sim.3782

Li, F., Morgan, K. L., & Zaslavsky, A. M. (2018). Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, *113*(521), 390–400. https://doi.org/10.1080/01621459.2016.1260466

Lunceford, J. K., & Davidian, M. (2004). Stratification and weighting via the propensity score in estimation of causal treatment effects: A comparative study. *Statistics in Medicine*, *23*(19), 2937–2960. https://doi.org/10.1002/sim.1903

Lundsberg, L. S., Illuzzi, J. L., Belanger, K., Triche, E. W., & Bracken, M. B. (2015). Low-to-moderate prenatal alcohol consumption and the risk of selected birth outcomes: A prospective cohort study. *Annals of epidemiology*, *25*(1), 46–54. https://doi.org/10.1016/j.annepidem.2014.10.011

Nembrini, S., König, I. R., & Wright, M. N. (2018). The revival of the gini importance? *Bioinformatics*, *34*(21), 3711–3718. https://doi.org/10.1093/bioinformatics/bty373

Ramsahai, R. R., Grieve, R., & Sekhon, J. S. (2011). Extending iterative matching methods: An approach to improving covariate balance that allows prioritisation. *Health Services* *and Outcomes Research Methodology*, *11*(3), 95–114. https://doi.org/10.1007/s10742-011-0075-5

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*(1), 41–55. https://doi.org/10.1093/biomet/70.1.41

Rubin, D. B. (1980). Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American Statistical Association*, *75*(371), 591–593. https://doi.org/10.2307/2287653

Shortreed, S. M., & Ertefaie, A. (2017). Outcome-adaptive lasso: Variable selection for causal inference. *Biometrics*, *73*(4), 1111–1122. https://doi.org/10.1111/biom.12679

Westreich, D., Lessler, J., & Funk, M. J. (2010). Propensity score estimation: Neural networks, support vector machines, decision trees (cart), and meta-classifiers as alternatives to logistic regression. *Journal of clinical epidemiology*, *63*(8), 826–833. https://doi.org/10.1016/j.jclinepi.2009.11.020

Zhao, P., Su, X., Ge, T., & Fan, J. (2016). Propensity score and proximity matching using random forest. *Contemporary clinical trials*, *47*, 85–92. https://doi.org/10.1016/j.cct.2015.12.012

3.A Tables

Table 3.6: Data generating processes.

DGP	Propensity score model	Outcome model
1	$\sum_{j=1}^{p} \nu_j X_j; \quad \nu = (1,1,0,0,1,1,0,...,0)$	$\theta A + \sum_{j=1}^{p} \beta_j X_j + \varepsilon$
2	$\sum_{j=1}^{p} \nu_j X_j; \quad \nu = (0.4,0.4,0,0,1,1,0,...,0)$	$\theta A + \sum_{j=1}^{p} \beta_j X_j + \varepsilon$
3	$\sum_{j=1}^{p} \nu_j X_j; \quad \nu = (1,1,0,0,1,1,0,...,0)$	$\theta A + 0.8X_1 \otimes X_2 + 0.8X_3 \otimes X_4 + \varepsilon$
4	$X_1(1 - X_2) + X_5(1 - X_6)$	$\theta A + 0.8X_1 \otimes X_2 + 0.8X_3 \otimes X_4 + \varepsilon$
5	$0.8X_1 \otimes X_2 + 0.8X_5 \otimes X_6$	$\theta A + 0.8X_1 \otimes X_2 + 0.8X_3 \otimes X_4 + \varepsilon$
6	$2\cos(X_2) + \sum_{j=1}^{p} \nu_j X_j; \quad \nu = (1,1,0,0,1,1,0,...,0)$	$\theta A + 0.8X_1 \otimes X_2 + 0.8X_3 \otimes X_4 + \varepsilon$
7	$2\mathbb{1}_{\{X_1>0\}}\mathbb{1}_{\{X_2>1\}} + 2\mathbb{1}_{\{X_5>0\}}\mathbb{1}_{\{X_6>1\}} + X_1 \otimes X_6$	$\theta A + 0.8X_1 \otimes X_2 + 0.8X_3 \otimes X_4 + \varepsilon$
8	$2X_2 \otimes (1 - X_6) + 2X_1\mathbb{1}_{\{X_9>1\}} + \sum_{j=1}^{p} \nu_j X_j;$	$\theta A + 0.8X_1 \otimes X_2 + 0.8X_3 \otimes X_4 +$
	$\nu = (1,1,1,1,1,1,0,0,1,1,0,...,0)$	$0.8X_5 \otimes X_6 + 0.8X_7 \otimes X_8 + \varepsilon$
9	$0.5X_1^2 + 0.5X_2 - X_3 \otimes X_4 + 0.5X_5 + 0.5X_6 + 0.5X_9^2 + 0.5X_{10}$	$\theta A + X_1 \otimes X_2 + X_3 \otimes X_4 +$
		$0.5X_2 + 0.5X_6 + X_7 \otimes X_8 + \varepsilon$
10	$-e^{(X_1)} + 0.4X_2 + e^{(X_3)} + 0.4X_4 + 0.5X_5^2 + X_6 \otimes X_9 + 0.4X_{10}$	$\theta A + 0.8X_1 \otimes X_2 + 0.8X_3 \otimes X_4 +$
		$0.8X_5 \otimes X_6 + 0.8X_7 \otimes X_8 + \varepsilon$
11	As setting 1 but with a correlation between the variables of around 0.2	
12	As setting 2 but with a correlation between the variables of around 0.2	
13	$X_1(1 - X_2) + X_5(1 - X_6)$	$X_1(1 - X_2) + X_3(1 - X_4)$
14	$2\cos(X_2) + \sum_{j=1}^{p} \nu_j X_j; \quad \nu = (1,1,0,0,1,1,0,...,0)$	$2\cos(X_2) + \sum_{j=1}^{p} \beta_j X_j;$
		$\beta = (0.6,0.6,0.6,0.6,0,0,0,...,0)$
15	$2\mathbb{1}_{\{X_1>0\}}\mathbb{1}_{\{X_2>1\}} + 2\mathbb{1}_{\{X_5>0\}}\mathbb{1}_{\{X_6>1\}} + X_1 \otimes X_6$	$2\mathbb{1}_{\{X_1>0\}}\mathbb{1}_{\{X_2>1\}} + 2\mathbb{1}_{\{X_3>0\}}\mathbb{1}_{\{X_4>1\}} + X_1 \otimes X_4$

Notes: Only setting 1 and 2 have a linear DGP. Setting 1 to 7 and 11 to 15 set $X_c = X_o = X_t = 2$ while setting 8 to 10 set $X_c = 6, X_o = X_t = 2$.

Table 3.7: RHC study: Selected covariates by categories

	% Selected					% Selected			
Covariates	RF full	RRF	OARF	OAL	Covariates	RF full	RRF	OARF	OAL
age	1.0	1.0	1.0	0.0	DASI	1.0	1.0	1.0	0.0
sex	1.0	1.0	0.0	0.0	APACHE score	1.0	0.8	1.0	1.0
raceblack	0.6	0.4	0.0	0.0	ca_yes	0.8	0.7	0.0	0.2
raceother	0.0	0.0	0.0	0.0	ca_meta	0.0	0.0	0.0	0.2
edu	1.0	1.0	0.0	0.0	surv2md1	1.0	1.0	1.0	1.0
income1	0.9	0.8	0.0	0.0	aps1	1.0	1.0	1.0	0.0
income2	0.9	0.9	0.0	0.0	scoma1	1.0	1.0	1.0	0.0
income3	0.3	0.2	0.0	0.0	wtkilo1	1.0	1.0	0.0	0.0
ins_care	0.9	0.9	0.0	0.0	temp1	1.0	1.0	0.0	0.0
ins_pcare	1.0	0.8	0.0	0.0	meanbp1	1.0	1.0	0.4	0.0
ins_caid	0.7	0.4	0.0	0.0	resp1	1.0	1.0	0.0	0.0
ins_no	0.0	0.0	0.0	0.0	hrt1	1.0	1.0	0.0	0.0
ins_carecaid	0.1	0.0	0.0	0.0	pafi1	1.0	1.0	0.3	0.0
cat1_copd	1.0	0.9	0.0	0.0	paco21	1.0	1.0	0.7	0.0
cat1_mosfsep	1.0	1.0	0.0	0.0	ph1	1.0	1.0	0.0	0.8
cat1_mosfmal	0.0	0.0	0.8	0.5	wblc1	1.0	1.0	1.0	0.0
cat1_chf	0.9	0.9	0.0	0.0	hema1	1.0	1.0	0.0	0.0
cat1_coma	0.2	0.1	1.0	1.0	sod1	1.0	1.0	0.0	0.0
cat1_cirr	0.1	0.1	0.0	0.0	pot1	1.0	1.0	0.0	0.0
cat1_lung	0.0	0.0	0.0	0.0	crea1	1.0	1.0	0.6	0.0
cat2_mosfsep	1.0	1.0	0.0	0.0	bili1	1.0	1.0	1.0	0.0
cat2_coma	0.0	0.0	0.0	0.0	alb1	1.0	1.0	0.0	0.0
cat2_mosfmal	0.0	0.0	0.2	0.2	cardiohx	1.0	0.9	0.0	0.0
cat2_lung	0.0	0.0	0.0	0.0	chfhx	1.0	0.9	0.0	0.1
cat2_cirr	0.0	0.0	0.0	0.0	dementhx	0.4	0.4	0.0	0.2
resp	1.0	1.0	0.0	0.0	psychhx	0.1	0.1	0.0	0.5
card	1.0	1.0	0.0	0.0	chrpulhx	1.0	0.8	0.0	0.0
neuro	1.0	1.0	0.0	0.0	renalhx	0.0	0.0	0.0	0.6
gastr	0.9	0.9	0.0	0.0	liverhx	0.1	0.1	0.0	0.4
renal	0.1	0.0	0.0	0.0	gibledhx	0.0	0.0	0.0	0.5
meta	0.0	0.0	0.0	0.0	malighx	1.0	0.9	0.0	0.7
hema	0.1	0.0	0.0	0.0	immunhx	1.0	0.9	0.0	0.0
seps	1.0	1.0	0.0	0.0	transhx	1.0	1.0	0.0	0.7
trauma	0.0	0.0	0.0	0.0	amihx	0.0	0.0	0.0	0.6

Table 3.8: Birth weight study: Selected covariates by categories

	% Selected			
Covariates	RF full	RRF	OARF	OAL
married	1.0	1.0	0.8	1.0
mhisp	0.4	0.3	0.0	0.9
fhisp	0.9	0.8	0.0	0.9
foreign	0.8	0.7	0.1	0.9
alcohol	1.0	1.0	0.0	0.9
deadkids	1.0	1.0	0.6	0.9
mage	1.0	1.0	1.0	0.8
medu	1.0	1.0	0.6	0.9
fage	1.0	1.0	1.0	0.9
fedu	1.0	1.0	0.6	0.9
nprenatal	1.0	1.0	1.0	1.0
monthslb	1.0	1.0	1.0	0.9
order	1.0	1.0	0.2	1.0
mrace	1.0	1.0	1.0	1.0
frace	1.0	1.0	1.0	1.0
prenatal	1.0	1.0	0.5	1.0
birthmonth	1.0	1.0	1.0	0.8
fbaby	1.0	1.0	0.1	1.0
prenatal1	1.0	1.0	0.1	1.0

3.B Proofs

Proof of unconfoundedness based on the propensity score.

We show that $Pr(D_i = 1|Y_i^0, Y_i^1, e(X_i)) = Pr(D_i = 1|e(X_i)) = e(X_i)$, implying independence of (Y_i^0, Y_i^1) and D_i conditional on $e(X_i)$. First, note that

$$\begin{aligned} \mathsf{E}_{D|e(X)}\left[D_i \mid e\left(X_i\right)\right] &= \mathsf{E}_{X|e(X)}\left\{\mathsf{E}_{D|X,e(X)}\left[D_i \mid X_i, e\left(X_i\right)\right] \mid e\left(X_i\right)\right\} \\ &= \mathsf{E}\left[e\left(X_i\right) \mid e\left(X_i\right)\right] = e\left(X_i\right) \end{aligned} \tag{3.14}$$

For simplification, we show the proof for Y^0 and note that the same logic follows for Y^1 or both. Using the law of iterated expectation and noticing that $e(X_i)$ is a function of X_i, it follows that,

$$\begin{aligned} \mathsf{E}\left[D_i \mid Y_i^0, e\left(X_i\right)\right] &= \mathsf{E}_{X|Y_i^0,e(X)}\left\{\mathsf{E}_{D|Y^0,X,e(X)}\left[D_i \mid Y_i^0, X_i, e\left(X_i\right)\right] \mid Y_i^0, e\left(X_i\right)\right\} \\ &= \mathsf{E}_{X|Y_i^0,e(X)}\left\{\mathsf{E}_{D|Y^0,X}\left[D_i \mid Y_i^0, X_i\right] \mid Y_i^0, e\left(X_i\right)\right\} \end{aligned} \tag{3.15}$$

Using the assumption of conditional independence of D_i and Y_i^0 given X_i allows us to neglect the conditioning in equation 3.15:

$$\begin{aligned} &\mathsf{E}_{X|Y_i^0,e(X)}\left\{\mathsf{E}_{D|Y^0,X}\left[D_i \mid Y_i^0, X_i\right] \mid Y_i^0, e\left(X_i\right)\right\} = \\ &\quad \mathsf{E}_{X|Y_i^0,e(X)}\left\{\mathsf{E}_{D|X}\left[D_i \mid X_i\right] \mid Y_i^0, e\left(X_i\right)\right\} = \\ &\qquad \mathsf{E}_{X|Y_i^0,e(X)}\left[e\left(x_i\right) \mid Y_i^0, e\left(X_i\right)\right] = e\left(X_i\right) \end{aligned} \tag{3.16}$$

Combining equation 3 and 2 shows that, $\mathsf{E}\left[D_i \mid Y_i^0, e\left(X_i\right)\right] = e(X_i)$. Based on this result and using equation 1 shows that given the propensity score, D_i is independent from Y_i^0.

Propensity score model for linear example:

Dependence is linear: $a(x) = 1X_1 + 1X_2 + 1X_5 + 1X_6$. Calculate the probability distribution for the vector a from the logit distribution function:

$$e_0(x) = \frac{exp\{a(x)\}}{(1 + exp\{a(x)\})}$$

Apply a random number generator from a Binomial function $B\{N, e_0(x)\}$ with probability for success = $e_0(x)$. This creates a vector $D \in \{0;1\}$ such that

$$D \overset{ind.}{\sim} \text{Bernoulli}\{e_0(x)\}.$$

Min-Max normalization to the interval $[0,1]$**:**

$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$$

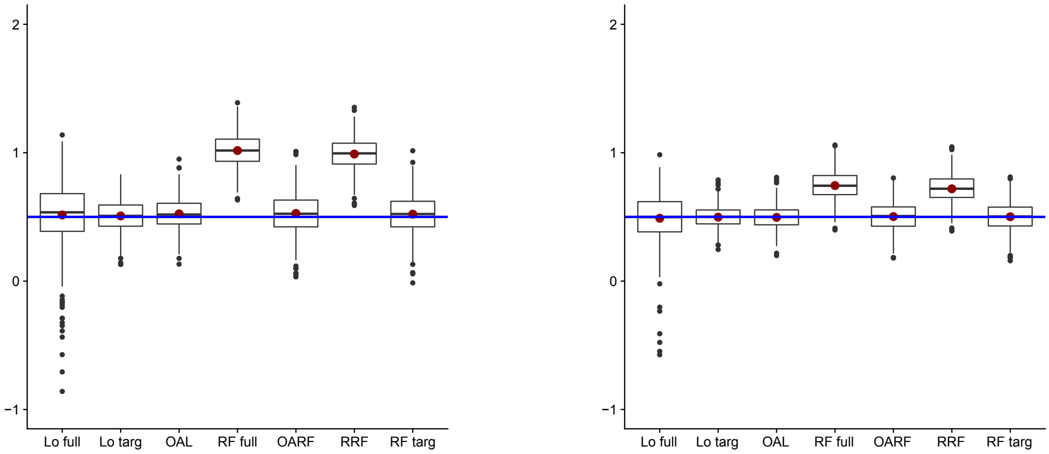
Since the $min(VarImp) = 0$, $a = 0$ and $b = 1$, the expression simplifies to

$$x' = \frac{x}{\max(x)}$$

If $min(VarImp) \neq 0$:

$$x' = \frac{(x - \min(x))}{\max(x) - \min(x)}$$

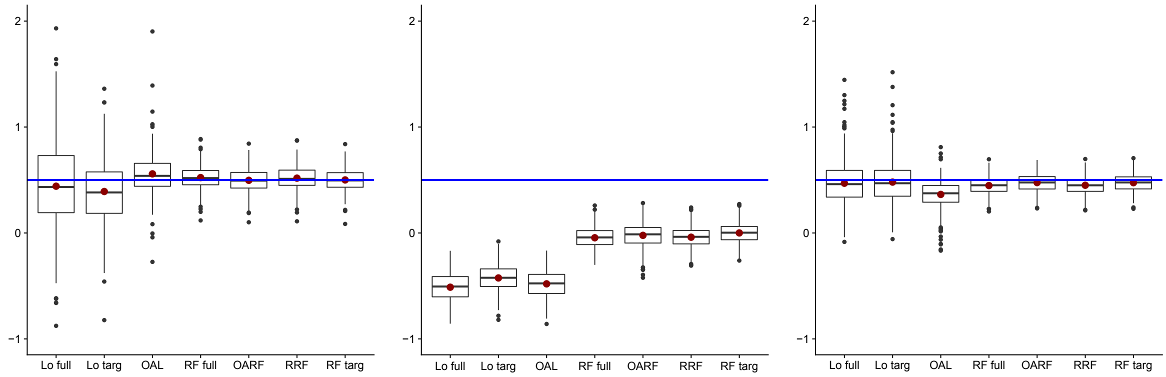
3.C Figures



(a) Linear Setting 1

(b) Linear Setting 2

Figure 3.8: Illustrations in linear settings. ATE via IPTW.

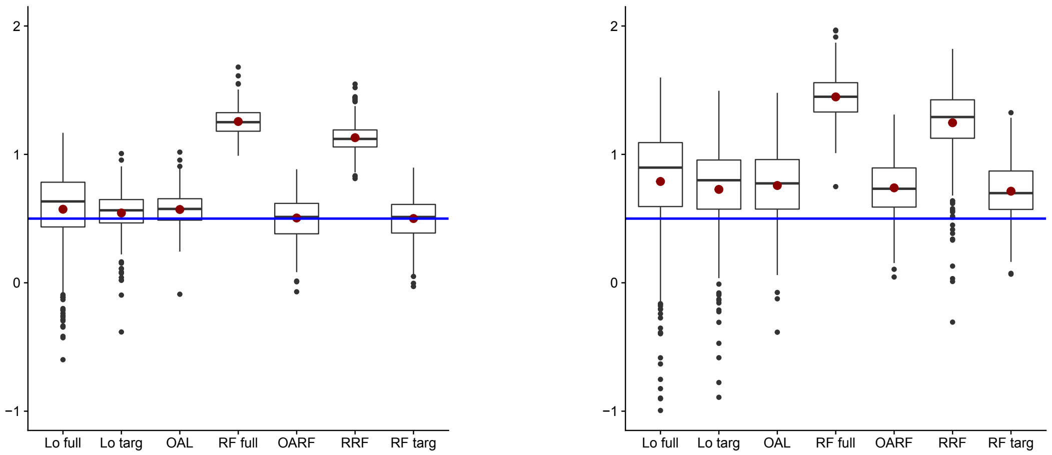


(a) Setting 8

(b) Setting 9

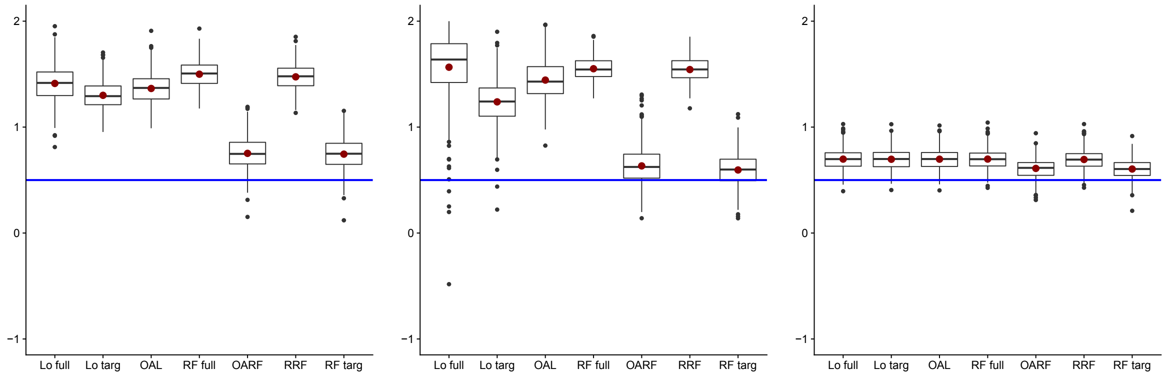
(c) Setting 10

Figure 3.9: Illustrations with more dependent covariates.



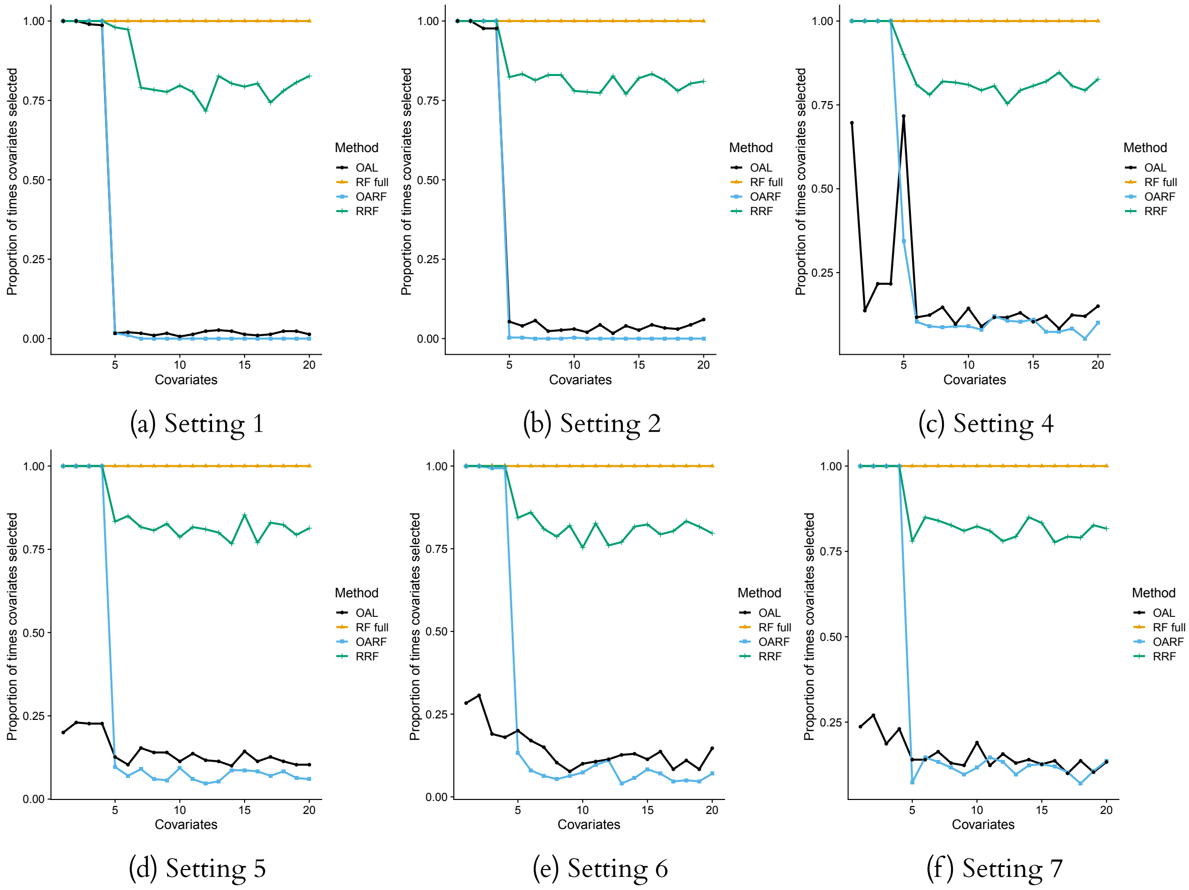
(a) Setting 1 with correlation $\rho = 0.2$ (b) Setting 1 with correlation $\rho = 0.5$

Figure 3.10: Illustrations with positive correlation between the covariates.



(a) Setting 13 (b) Setting 14 (c) Setting 15

Figure 3.11: Illustrations with complex outcome function.



(a) Setting 1

(b) Setting 2

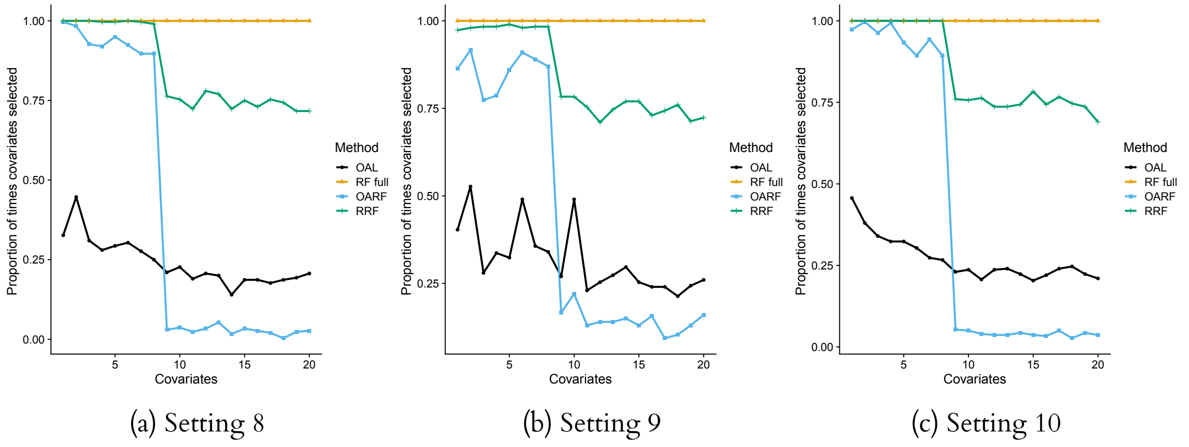
(c) Setting 4

(d) Setting 5

(e) Setting 6

(f) Setting 7

Figure 3.12: Illustrations of selected variables over 500 simulations.

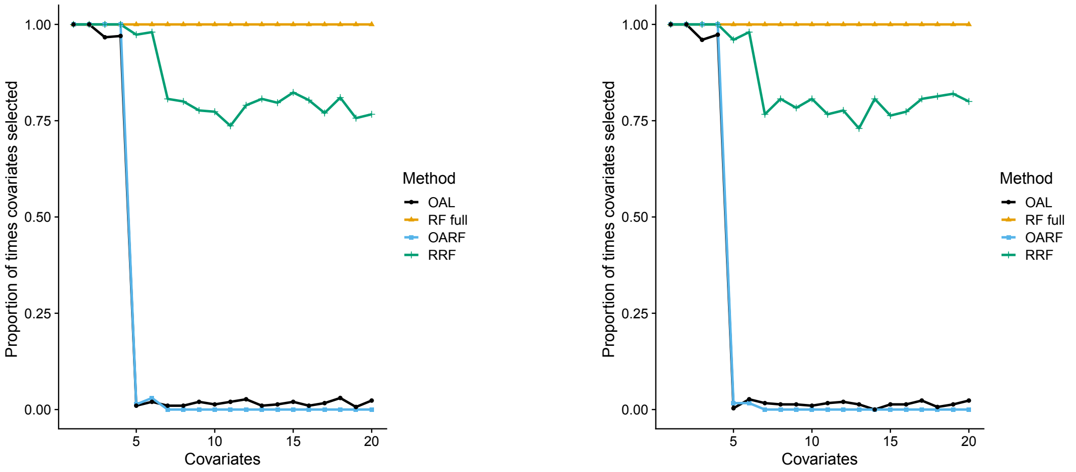


(a) Setting 8

(b) Setting 9

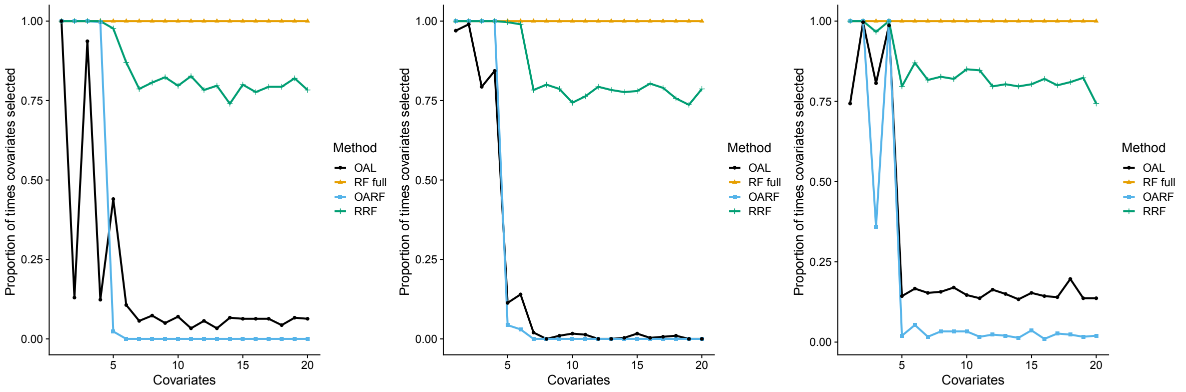
(c) Setting 10

Figure 3.13: Illustrations of selected variables over 500 simulations. Favourable covariates are X_1 to X_8.



(a) Setting 1 with correlation $\rho = 0.2$ (b) Setting 1 with correlation $\rho = 0.5$

Figure 3.14: Illustrations of selected variable coverage over 500 simulations and under positive correlation. Favourable covariates are X_1 to X_4.



(a) Setting 13 (b) Setting 14 (c) Setting 15

Figure 3.15: Illustrations of selected variable coverage over 500 simulations. Favourable covariates are X_1 to X_4.

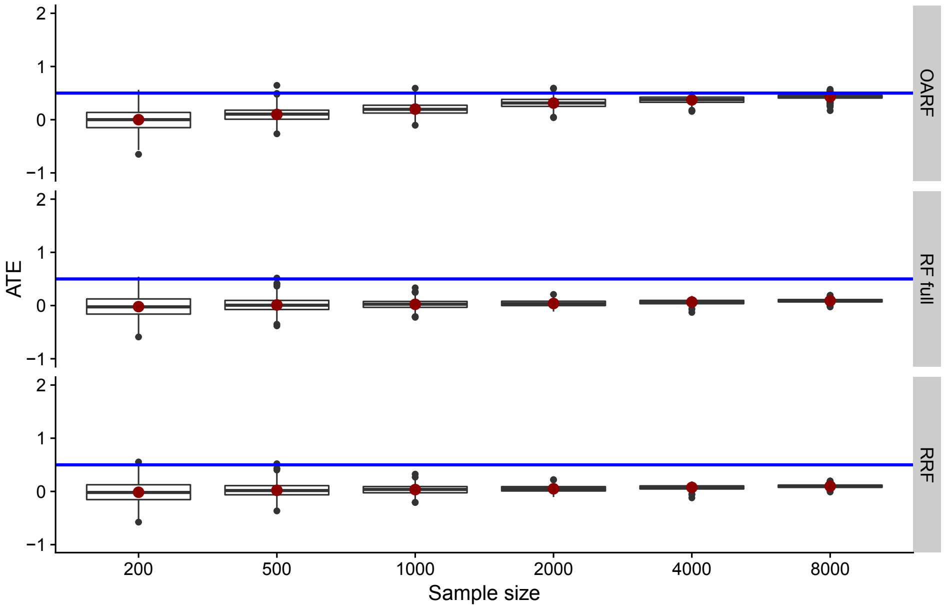


Figure 3.16: Setting 4: Boxplots of ATE for varying sample size.

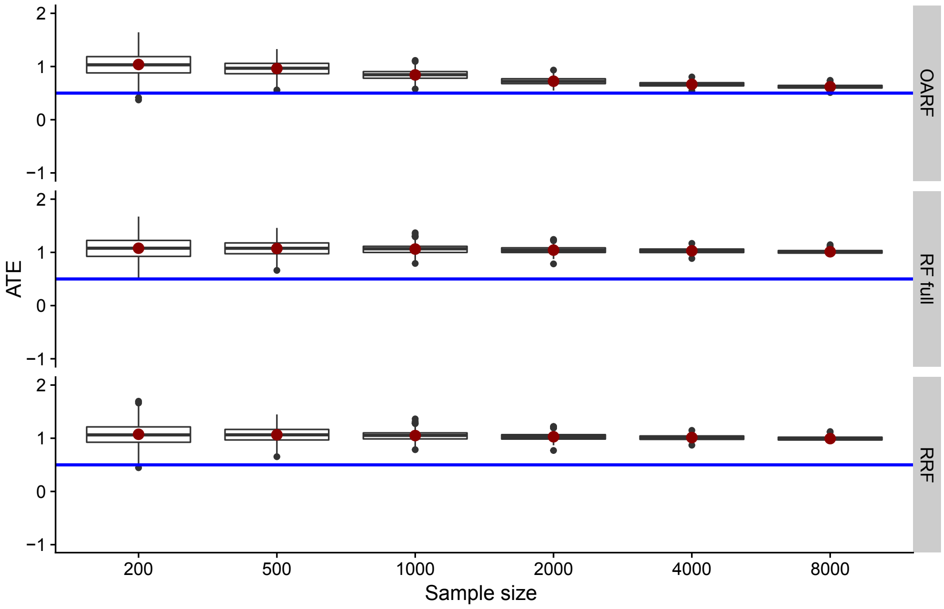


Figure 3.17: Setting 5: Boxplots of ATE for varying sample size.

3.D Selection of tuning parameters

The outcome-adaptive lasso (OAL) approach has to important tuning parameters. The parameter λ_n is the regularization parameter that needs to be optimised while the parameter γ is set to fulfil $\lambda_n n^{\gamma/2-1} = n^2$, with $N = n$. In the IPTW estimator, the propensity score is used to balance the covariate distribution between the treatment and the control group. Shortreed and Ertefaie (2017) propose to select λ_n by minimizing a weighted absolute mean difference (wAMD) using the covariates and the propensity score for the treatment and control group:

$$\text{wAMD}\left(\lambda_n\right) = \sum_{j=1}^{d} \left|\widetilde{\beta}_j\right| \left| \frac{\sum_{i=1}^{n} \widehat{\tau}_i^{\lambda_n} X_{ij} D_i}{\sum_{i=1}^{n} \widehat{\tau}_i^{\lambda_n} D_i} - \frac{\sum_{i=1}^{n} \widehat{\tau}_i^{\lambda_n} X_{ij}\left(1 - D_i\right)}{\sum_{i=1}^{n} \widehat{\tau}_i^{\lambda_n}\left(1 - D_i\right)} \right| \tag{3.17}$$

$$\widehat{\tau}_i^{\lambda_n} = \frac{D_i}{\hat{e}_i^{\lambda_n}\left\{\mathbf{X}_i, \hat{\alpha}(OAL)\right\}} + \frac{1 - D_i}{1 - \hat{e}_i^{\lambda_n}\left\{\mathbf{X}_i, \widehat{\alpha}(OAL)\right\}} \tag{3.18}$$

Equation 3.18 represents the IPT-weights obtained from the propensity score model using the OAL method for variable selection. The λ_n value that minimizes the wAMD is used to estimate the ATE using the propensity score estimates given the specific λ_n and γ.

In equation 3.17 the beta coefficients are used to weight the covariate balancing based on the strengths of the coefficients. Since we do not require exact coefficients from a linear model for the weighting, we could use the wAMD to tune the OARF. Instead of the coefficients, the $|\widetilde{\beta}_j|$ could contain the variable importance scores (they don't even need to be standardized). Since we mainly want to find a good penalization procedure for the propensity score function, possible tuning parameters could include the threshold for the initial feature space, the normalization of the importance score, and whether to apply different penalty weights based on the depth of the tree. For example, now we use the penalty Imp_j^* for each node that contains the variable j and is not in the initial feature set $\mathbb{F}$. Another possibility would be to use $(Imp_j^*)^{\xi}$ where ξ states the depth of the node. Considering the depth of a node would make sense if we believe that splits near the end of a tree are less important and hence are heavier penalized. In the simulation settings that we consider, we find that the default values lead to a significant decrease in bias and variance. Even if the `ranger` implementation of the RF is quite fast, the tuning of parameters is computationally expensive (in comparison to the tuning of the lasso). These are the main reasons why we do not consider parameter tuning at this stage but provide a possible approach to do so if necessary.

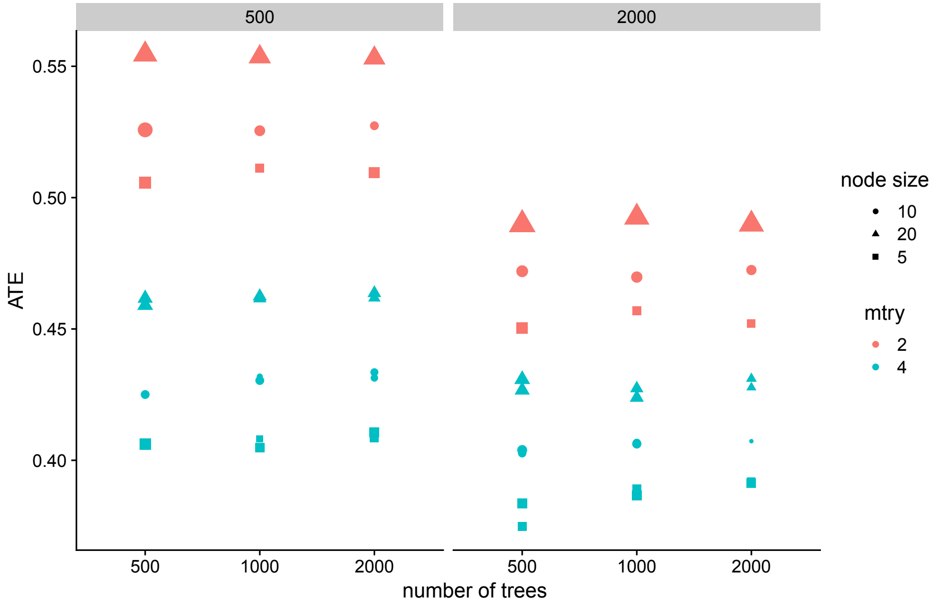


Figure 3.18: ATE for different tuning parameters based on wAMD.

The RF tuning parameters aim to get high prediction accuracy. We do not aim for a high classification rate using the propensity score rather that it balances the covariate distribution between the treated and control group observations. Still, we can use the wAMD and tune, for example, the number of random variables to choose at each split (mtry), the number of trees, the node size, or imbalance methods, as long as we maximize the weighted absolute mean difference. Figure 3.18 shows the ATE applying different tuning parameters on the OARF. The ATE is a median ATE over 200 Monte Carlo iterations. We also show the relative amount of tuning combinations that were chosen when minimizing the wAMD. We have 18 combinations of tuning parameters to choose from. The more often each combination was chosen based on minimizing the wAMD, the bigger the symbol. We find that, with 500 observations, the most often chosen combination has a node size of 20 and selected 2 variables at each split. The same holds when increasing the sample size to 2000 observations. With the latter amount of observations, the best tuning parameter combination is closest to the true ATE of 0.5. The number of trees seems not to be important since the results are mainly constant.

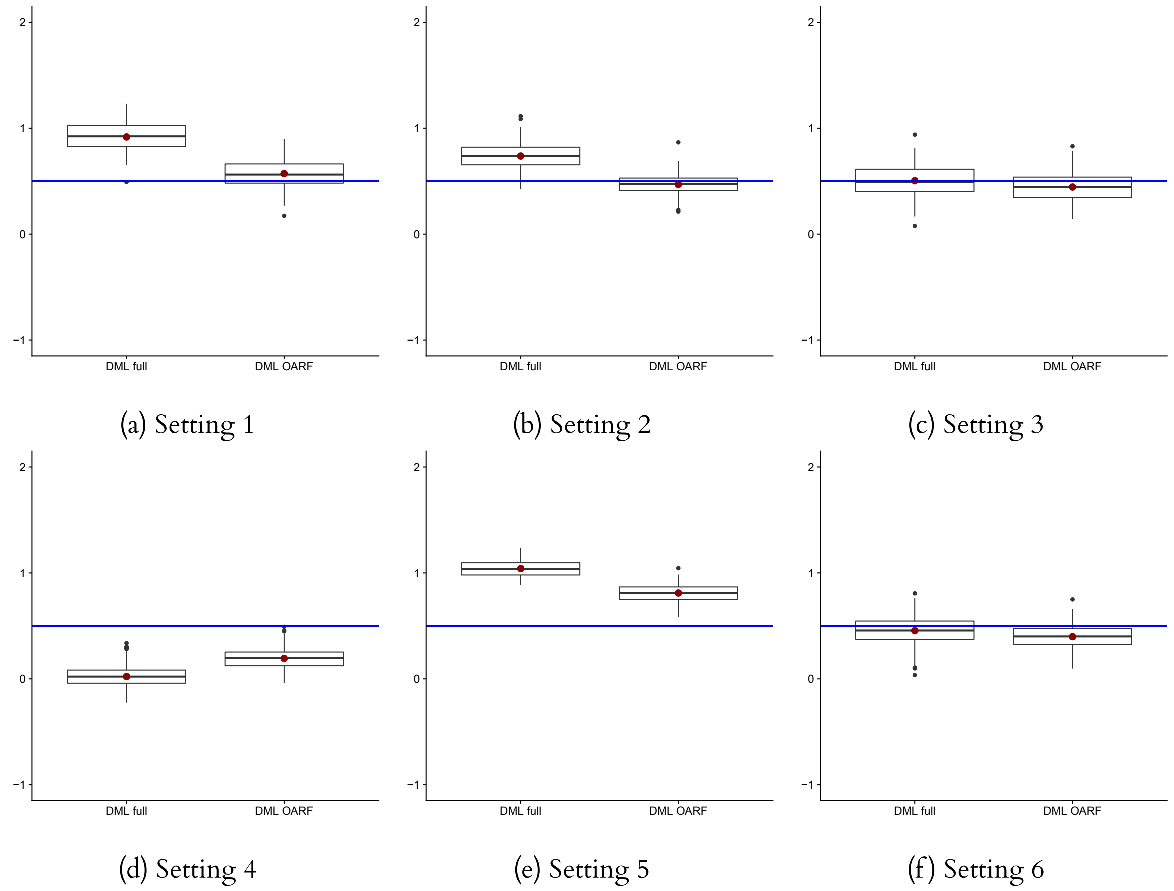
3.E Generalization of OARF

The balancing of covariates through the propensity score generalizes to other methods besides the IPTW estimator. We illustrate the ATE estimation using the double machine learning (DML) approach proposed by Chernozhukov et al. (2018). The estimation is based on the residual-on-residual approach to cancel out the effect from confounding covariates. If more variables are at choice from which only a few are true confounders it might be more beneficial to select variables for the propensity score estimation. Since this approach needs to estimate two functions (the conditional mean of the outcome and the propensity score function) we do not consider this approach as a direct comparison in the main part of the paper.

The treatment effect is estimated as follows: First, we estimate the conditional mean of the outcome by regressing X on Y. This results in the function $\hat{\ell}(X)$. Second, we estimate two propensity score models. One that uses all covariates and the second based on the OARF. We then estimate the residuals $\hat{U} = Y - \hat{\ell}(X)$ and $\hat{V}_m = Y - \hat{e}_m(X)$. Note that only $\hat{V}$ depends on the method m but $\hat{U}$ is only estimated once. The treatment effect is then estimated by:

$$\check{\theta} = \left(\frac{1}{n} \sum_{i=1} \hat{V}_i \hat{V}_i \right)^{-1} \frac{1}{n} \sum_{i=1} \hat{V}_i \left(\hat{U} \right) \tag{3.19}$$

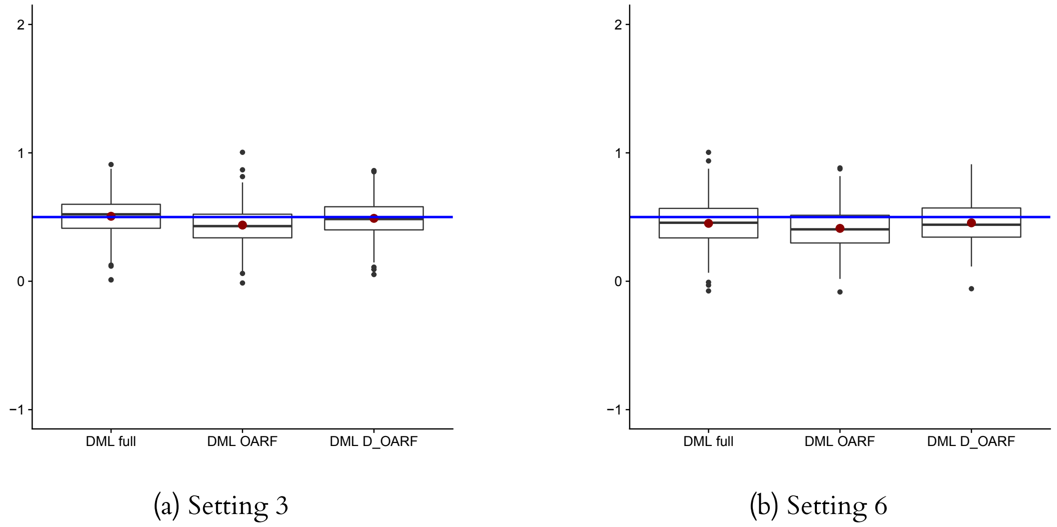
As in the previous settings we use sample splitting and cross-fitting to estimate the final parameter. Figure 3.19 shows the ATE estimated using the full RF and the OARF to estimate the propensity score. We find that using the OARF, the ATE is less biased and has a smaller variance in most settings. In settings 3 and 6, however, we find that both methods have the same variance but using the full RF might be less biased. The reason might be that if both functions, $\mathsf{E}[Y|X]$ and $\mathsf{E}[D|X]$ are quite complicated it might be desirable to also regularize the first function. This means that we only use the variables that are confounders and predictive on Y to estimate the outcome function. In Figure 3.20 we show that using the regularized RF to estimate the outcome model, we can decrease the bias. This is as expected since we want to exclude variables that have no association on Y. We call the additional approach where we regularize both functions based on outcome variables, “double OARF” (DOARF).



(a) Setting 1 (b) Setting 2 (c) Setting 3

(d) Setting 4 (e) Setting 5 (f) Setting 6

Figure 3.19: Illustrations of DML using all covariates and OARF based propensity score.



(a) Setting 3 (b) Setting 6

Figure 3.20: D_OARF uses the OARF for both functions $\hat{\ell}(X)$ and $\hat{e}(X)$

Chapter 4

Cross-Fitting and Averaging for Machine Learning Estimation of Heterogeneous Treatment Effects

Abstract

We investigate the finite sample performance of sample splitting, cross-fitting and averaging for the estimation of the conditional average treatment effect. Recently proposed methods, so-called meta-learners, make use of machine learning (ML) to estimate different nuisance functions and hence allow for fewer restrictions on the underlying structure of the data. To limit a potential overfitting bias the data is usually split into different folds. Averaging over folds aims to restore efficiency. We employ a Monte Carlo study 15 different data generating processes and consider twelve different estimators that vary in sample-splitting, cross-fitting and averaging procedures. We investigate the performance of each estimator on four different meta-learners: the doubly-robust, R-, T-, and X-learner. Together with four different sample sizes, we have 240 different settings to evaluate on. We find that the performance of all meta-learners heavily depends on the procedure of splitting and averaging. The best performance in terms of mean squared error among the sample split estimators can be achieved when applying cross-fitting and median averaging. Cross-fitting can decease the MSE often by more than 50%. Additional median averaging further decreases the MSE due to a decrease in variance. We illustrate the difference between the estimators in two empirical examples and confirm the results from the simulation study.

JEL Classification: C14, C31, C51, C63

Replication code is available on Quantlet .

4.1 Introduction

Recent methods to estimate the conditional average treatment effect (CATE) propose using machine learning (ML) methods when the underlying data is high-dimensional or has non-linear dependencies. The simplest approach is to estimate two conditional mean functions, one for the treated observations and one for the non-treated, and then take the difference. Even if this method works for observational studies, where the treatment assignment mechanism depends on observed covariates (self-selection into treatment), there are more efficient methods to estimate the CATE. One way is to explicitly control for such selection bias by estimating the probability of treatment and control for it. The building of such a model or function is a classic classification task for which machine learning methods are well suited to find generalizable predictive patterns. Since we are only interested in getting a good prediction of the probability of treatment, we do not need to know the underlying structural form of this function which enables black-box ML methods to be sufficient. Such a function is called a nuisance function. Another nuisance function in this context is, for example, the conditional mean function from the outcome variable.

ML methods use regularization to decrease the variance of an estimator. However, there is a trade-off between an introduced bias on the parameter of interest through regularization and overfitting. Sample splitting helps to limit overfitting and allows for less restrictive assumptions on the nuisance functions. The idea is to use at least two different samples or folds (say, A and M), one to fit (or train) the nuisance functions (fold A), and one for the estimation of the parameter of interest (fold M).

To overcome a potential loss in efficiency, since only a subset of the data is used when estimating the CATE, cross-fitting is an increasingly popular approach to combine ML methods with semi-parametric estimation problems; see, for example, Chernozhukov, Chetverikov, et al. (2018), Newey and Robins (2018) and Athey and Wager (2017). Cross-fitting estimates the parameter of interest using the subset M and then switches the roles of the sets now using subset M for training and subset A for estimation. The two results are then averaged. If we want to use this procedure to make predictions, we would build two prediction models based on the roles of the samples and average the resulting values for each observation.

So far there are no clear proposals on how to exactly conduct sample splitting and cross-fitting. We can split the sample in two or more folds and average among those folds as suggested by Chernozhukov, Chetverikov, et al., 2018 for the average treatment effect (ATE) or Nie and Wager (2020) as they do for their R-learner method to estimate the CATE. The former uses two- and five-folds, while the latter uses five- and ten-folds. In both cases, the estimation of models for the nuisance parameters (we sometimes refer to this as training of a model or function) is done on all folds but the one which is used for the prediction of the nuisance parameters and hence for the estimation of the CATE. This is quite similar to cross-validation where one uses part of the data to train different models and the remaining part for validating them. Newey and Robins (2018) and Kennedy (2020) suggested for the doubly-robust (DR) estimator to not only use different folds for training and estimation but also to train each nuisance function on a different fold. Zivich and Breskin (2020) provide a simulation study on the aforementioned so-called double sample splitting as well as cross-fitting estimators and demonstrate the performance for the ATE.

While the sample splitting procedure allows for less restrictive assumptions on the ML estimators, sample splitting can introduce a new bias due to a specific sample. This can become more problematic the smaller the whole sample is. If we have a sample with only 500 observations, 5-fold cross-fitting would use 400 observations for training and 100 for estimation. If we furthermore train each nuisance function with a different fold, we could only use 125 observations for building any estimator that includes up to four different functions to estimate. To average any potential bias from sample splitting, we can repeat the estimation multiple times and take the mean or the median over all the estimators. This approach was first proposed for the ATE by Chernozhukov, Chetverikov, et al. (2018) as well as for the CATE when the nuisance functions might be misspecified due to high-dimensional problems as in Chernozhukov, Demirer, et al. (2018).

For all the different approaches, 2-fold vs. K-fold, double sample splitting, cross-fitting, and repeated sample splitting and averaging, there is little to no guidance for practitioners. In this paper, we consider different variations of the approaches and evaluate them independently on a class of recent meta-learners that estimate the CATE. Taking all suggestions from the literature into account we end up with twelve different estimators. The meta-learners we use are the T-learner, DR-learner, a recent version of the class of orthogonal learners (R-learner), and X-learner. We use all twelve estimators on all meta-learners in a Monte Carlo study to assess the accuracy of the CATE predictions on an independent test-set.

The paper is structured as follows: First, we motivate the topic through an overview of the recent literature. Second, we briefly describe the four different meta-learners and discuss the idea of cross-fitting and double sample splitting as well as the purpose of taking the median over multiple iterations. Third, we describe in detail how we generate our data and why - especially since our results are based on such simulations. In section four we show and discuss our results for twelve different estimators that are based on variations of splitting, cross-fitting, and averaging for each of the meta-learners. Section five provides two empirical examples where we use all estimators to predict the CATE. Last, we summarize the results and provide guidelines for practitioners.

4.2 Methods

We base all our estimators on the potential outcome framework for which we use the following notations: Each observation has two potential outcomes, Y^1 and Y^0 of which we only observe one, namely the former if someone was treated or the latter if not. We denote this by the binary treatment indicator $D \in \{0;1\}$ and denote observed covariates $X \in \mathbb{R}^p$. To interpret the estimated parameter as a causal relationship, the following assumptions are needed; see, for example, Rubin (1980):

1. Conditional independence (or conditional ignorability/exogeneity or conditional unconfoundedness), $(Y_i^1, Y_i^0) \perp\!\!\!\perp D_i | X_i$. 2. Stable Unit Treatment Value Assumption , $Y_i = Y_i^0 + D_i(Y_i^1 - Y_i^0)$. 3. Overlap Assumption and the definition of the propensity score:

$$\forall x \in supp(X_i), \quad 0 < \mathsf{P}(D_i = 1 | X_i = x) < 1, \tag{4.1}$$

$$\mathsf{P}(D_i = 1 | X_i = x) \equiv e(x). \tag{4.2}$$

4. Exogeneity of covariates:

$$X_i^1 = X_i^0. \tag{4.3}$$

Assumption 1 together with Assumption 4 states that the treatment assignment is independent of the two potential outcomes and that the covariates are not affected by the treatment. Assumption 2 ensures that there is no interference, no spillover effects, and no hidden variation between treated and non-treated observations. Assumption 3 states that no subpopulation defined by $X_i = x$ is entirely located in the treatment or control group, hence the treatment probability needs to be bounded away from zero and one. Equation 4.2 is the propensity score.

We define the conditional expectation of the outcome for the treatment or control group as

$$\mu_d(x) = \mathsf{E}[Y_i | X_i = x, D_i = d] \quad with \quad D \in \{0, 1\}. \tag{4.4}$$

If we don't use any subscript, we refer to this function as the general conditional expectation.

Our parameter is interest is the CATE ($\tau(x)$), which is formally defined as:

$$\tau(x) = E\left[Y_i^1 - Y_i^0 \mid X_i = x\right] = \mu^1(x) - \mu^0(x) \tag{4.5}$$

When reviewing recently proposed methods for the estimation of the CATE, we can categorize them into two groups. The first group contains methods that transform the variables to a pseudo-outcome which is used as a proxy for the CATE function (the literature calls these transformed outcome approaches, meta-learners, or generic ML algorithms). In a second step, off-the-shelf machine learning methods can be used to estimate the final CATE. The second group of methods leaves the variables untouched but alters existing machine learning methods in a way that they can be used to estimate the CATE directly (examples are Causal Boosting by Powers et al. (2018), Causal Forest by Athey et al. (2019) or the Bayesian Regression Tree Models for Causal Inference by Hahn et al. (2020)). See Künzel et al. (2019) for a comparison between the S-, T-, and X-learner as well as the Causal Forest in a simulation study. Knaus et al. (2020) compare the inverse probability weighting (IPW) estimator, doubly-robust (DR), modified covariate method (MCM), R-learner and different versions of the Causal Forest in an empirical Monte Carlo study while Nie and Wager (2020) compare their R-learner with the S-, T-, X- and U-learner as well as causal boosting. Regarding the base learners (the ML methods), Künzel et al. (2019) use a Random Forest (RF) and a Bayesian Additive Regression Trees (BART) algorithm. Knaus et al. (2020) use RF and lasso while Nie and Wager (2020) use boosting and lasso for the estimation of the nuisance functions.

In this paper, we concentrate only on methods from the first group, hence meta-learners that are flexible in the use of machine learning methods. Since these methods need to estimate different nuisance functions, we have more options to split and average samples and hence to estimate the CATE.

How we choose ML methods:

The accuracy of the CATE estimation depends on the accuracy of the nuisance functions and hence on the choice of the ML method. For example, Knaus et al. (2020) find that when using the lasso, the estimators can have heavy tails in smaller samples. To minimize the dependence of the ML methods one method might be to consider a range of different popular methods. To choose which ML method to use for each nuisance function as well as for any additional functions, one could use an ensemble and stacking method. In such a setting, not only one ML method may be chosen but an ensemble of methods that are stacked together with different weights. One problem with this approach is that it is computationally expensive. For example, in our case, we have 12 estimators that we want to compare. Given the meta-learner, we need to estimate several nuisance functions. If we choose to provide an ensemble for each nuisance function that contains the random forest, lasso and gradient boosted trees, the estimation takes about 72 hours for 300 iterations. This is just for one DGP. Our aim is to compare estimators based on different sample splitting techniques, not different functionals. Using the same method for all functions should not influence the performance between the estimators. Therefore, we use the random forest algorithm (ranger package in R) to estimate all nuisance functions and to predict the CATE.

4.2.1 Meta-Learners

In the following, we briefly describe the considered meta-learners. Except for the T-learner, all other methods generate a pseudo-outcome in the first step which can be seen as an approximation of the conditional average treatment effect. The last step regresses this function on the covariates to get the final estimate. The DR-, R- and X-learner also require to estimate the propensity score as an additional nuisance function if the data does not come from a randomized control trial (RCT).

Two-model learner (T-learner):

The T-learner is a two step approach where the conditional mean functions $\mu_1(x) = \mathsf{E}[Y^1|X_i = x]$ and $\mu_0(x) = \mathsf{E}[Y^0|X_i = x]$ are estimated separately with any generic machine learning algorithm. The difference between the two functions results in the CATE as shown in Table 4.1.

One problem with the T-learner is that it aims to minimize the mean squared error for each separate function rather than to minimize the mean squared error of the treatment effect. See, for example, Kennedy (2020) and Künzel et al. (2019) for settings when the T-learner is not the optimal choice.

Doubly-robust learner (DR-learner):

A more efficient method than the T-learner can be the DR-learner. It builds on the T-learner and adds a version of inverse probability weighting (IPW) scheme on the residuals of both regression functions $\{Y - \mu_D(x)\}$. We can think of it as combining two different models and hence avoid drawbacks like the minimization goal from the T-learner and a potentially high variance from an IPW model when some propensity scores are small. The doubly-robust learner takes its name from a double robustness property which states that the estimator remains consistent if either the propensity score model or the conditional outcome model is correctly specified. This is at least true for the average treatment effect (Lunceford & Davidian, 2004). Recently, this estimator has gained popularity to estimate the CATE, especially in high-dimensional settings. See, for example, the work by Fan et al. (2020) and Zimmert and Lechner (2019). Kennedy (2020) which find that for estimating the CATE, the finite-sample error-bound from the DR-learner at most deviates from an oracle error rate by the product of the mean squared error of the propensity score and the conditional mean estimator.

Orthogonal-learners (here: R-learner):

The orthogonal learner makes use of the idea of orthogonalization to cancel out any selection bias that may arise in observational studies from observed covariates. Here, the residuals from the regression of Y on X are regressed on the residuals from the regression of D on X and weighted by the squared residuals, $\{D - \hat{e}(x)\}^2$. This is similar to the double machine learning approach from Chernozhukov, Chetverikov, et al. (2018) where their estimator of interest is the ATE. Nie and Wager (2020) develop a general class of two-step algorithms for the estimation of the CATE. Their so-called "R-learner", as from residualization and a homage to Robinson 1988, makes explicit use of machine learning methods.

Achieving Neyman orthogonality using a residuals-on-residuals (or debiasing) approach has a long history in econometrics (see the Frisch–Waugh–Lovell theorem from the 1930s for linear regression) and mainly builds on the work by Robinson (1988) who replaces the linear parts by non-parametric kernel regression. Chernozhukov, Demirer, et al. (2018) adopt the debiasing approach using ML methods in RCTs where the parameter of interest is some feature (like a best linear predictor) of the CATE.

X-learner:
Künzel et al. (2019) propose the X-learner which estimates a treatment effect separately for the control and the treatment group. This might be especially helpful in situations where the proportion of the two groups is highly imbalanced. The X-learner has several steps. The first step is identical to the T-learner, namely estimating the two conditional mean functions. In the second step, however, the difference is found in the observed outcome for the treated and control group, respectively. The two imputed treatment effects ($\hat{\psi}_X^1 := Y^1 - \hat{\mu}_0(x^1)$ and $\hat{\psi}_X^0 := \hat{\mu}_1(x^0) - Y^0$) are now used in a third step to regress them individually on the covariates to obtain $\hat{\tau}_0(x)$ (the CATE for the control group) and $\hat{\tau}_1(x)$ (the CATE for the treatment group). The final estimator combines the two estimators plus some weights, $g(x)$:

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

The weights can, for example, be set to $1 - \hat{e}(x)$ for the treatment group and $\hat{e}(x)$ for the control group estimate, respectively.

Summary of meta-learners:

We summarise the considered meta-learners in Table 4.1 where $\hat{\psi}$ states the pseudo-outcome or estimator for each of the learners. The last column counts the number of nuisance functions to train to estimate the pseudo-outcome or estimator and in brackets, we state the total number of models needed to get the final CATE estimate on the whole dataset. Note that the X-learner is regressed only for the treated observations and again only for the observations in the control group. This is why we need two more additional models for the final estimate.

The estimators from Table 4.1 can be represented as a weighted minimization problem which solves the following:

$$\min_{\tau}\left\{\frac{1}{N}\sum_{i=1}^{N} w_i\left[\hat{\psi}_i - \tau(x)\right]^2\right\}.$$

4.2.2 Sample splitting and cross-fitting

To aim for a consistent estimator, we need to assume certain complexity conditions on the nuisance functions. Specifically, we want them to be smooth (i.e. differentiable) and the

Table 4.1: Summary of meta-learners

Method	Estimator/Pseudo-outcome	Weights (w_i)	# of Models
T-learner	$\hat{\psi}_T = \hat{\mu}_1(x) - \hat{\mu}_0(x)$	1	2 (3)
DR-learner	$\hat{\psi}_{DR} = \hat{\psi}_T + \frac{D\left(Y - \hat{\mu}_1(x)\right)}{\hat{e}(x)} - \frac{(1-D)\left(Y - \hat{\mu}_0(x)\right)}{\left(1 - \hat{e}(x)\right)}$	1	3 (4)
R-learner	$\hat{\psi}_R = \frac{\left(Y - \hat{\mu}(x)\right)}{\left(D - \hat{e}(x)\right)}$	$\left(D - \hat{e}(x)\right)^2$	2 (3)
X-learner	$\hat{\psi}_X^1 := Y^1 - \hat{\mu}_0\left(x^1\right)$ $\hat{\psi}_X^0 := \hat{\mu}_1\left(x^0\right) - Y^0$	1	3 (5)

Notes: Considered meta-learners that estimate the CATE. # of Models counts the number of nuisance functions to estimate the pseudo-outcome. Numbers in brackets count the total number of models to train to get the final CATE estimate.

entropy of the candidate nuisance functions to be small enough to fulfill Donsker conditions (e.g. if we assume Lipschitz parametric functions or VC classes). In high-dimensional settings (p>n) or when using ML methods that are complex or adaptive, the Donsker conditions might not hold; see, for example, Robins et al. (2013), Chernozhukov et al. (2016) and Rotnitzky et al. (2017). As Chernozhukov, Chetverikov, et al. (2018) noticed, verification of the entropy condition is so far only available for certain classes of machine learning methods, such as lasso and post-lasso. For classes that employ cross-validation or for hybrid methods (like the SuperLearner), it is likely difficult to verify such conditions. Luckily, there is an easy solution available: sample-splitting. When splitting the sample we can use independent sets for estimating the nuisance functions and constructing the estimating equation. By using different sets, we can treat the nuisance functions as fixed functions which allows avoiding conditions on the complexity. It also allows us to use any ML method such as random forest or boosting or even an ensemble of different methods. The sample splitting approach to avoid smoothness conditions dates back at least to Bickel (1982) and was extended to also use cross-fitting by Schick (1986).

Concerning sample splitting and cross-fitting, there are different strategies on how to split the data and average the results. First, consider the 50:50 sample splitting where the data is split into two equal folds, namely an auxiliary sample (A) (which is used to train all nuisance functions) and the main sample (M) for the estimation. Second, the data can be split into that many equal parts as we have functions to estimate and we use each fold to train a different nuisance function. Newey and Robins (2018) use this approach with undersmoothing to reduce bias (they call this approach double cross-fitting). Kennedy (2020) adopts this approach and shows that one can achieve faster rates in estimating the CATE using the DR-learner. He restricts the estimation of the two conditional mean functions ($\hat{\mu}_1(x)$ and $\hat{\mu}_0(x)$) to the same fold, the propensity score to another fold and the estimation of the parameter to the remaining fold. Hence we need three folds for the DR-learner. To not get confused about the names of the different approaches, we call the technique where we use different folds on each nuisance function but without cross-fitting "double sample splitting" (following Kennedy, 2020).

Both approaches can be extended to not only two or three folds but K folds. The idea here is to use more observations to train the nuisance functions. For example, we can assign $(100 - \frac{100}{K})$% of the observations to sample A and $(\frac{100}{K})$% to sample M. For simplicity, we refer to the two- and K-fold estimator as $\hat{\tau}_K$ while K denotes the number of folds. This is independent of whether we use standard or double sample splitting.

The two estimators above might however not be efficient. Sample splitting reduces the available amount of data used for both, the estimation of the nuisance function and the final parameter by construction. This leads to a loss in efficiency and statistical power in finite samples. If we want to make use of the full sample to restore efficiency, we can switch the roles of the samples thereby using sample M for training and sample A for estimation and again to regress the pseudo-outcome on the covariates. Prediction on the whole data or an independent test set leads to two estimates which are then averaged. If we use K folds, we repeat the procedure until each fold was used for estimation and average over the K estimates. Let $S = \{Y_i, D_i, X_i\}_{i \in \{1...N\}}$ be the set of observations from the whole training sample. Then, we define $S_k = \{Y_i, D_i, X_i\}_{i \in N_k}$ as the set of observations for each fold using only observations N_k while $\biguplus_{k=1}^{K} N_k = \{1...N\}$. We use data from the set S_k of the training sample to estimate the pseudo-outcome while using independent test data to estimate the final estimator:

$$\hat{\tau}_k(x) = \mathsf{E}[\hat{\psi}(S_k) \mid X_i = x], \tag{4.6}$$

$$\tilde{\tau}_K(x) = \frac{1}{K} \sum_{k=1}^{K} \hat{\tau}_k(x). \tag{4.7}$$

We give an example of the benefit from cross-fitting in Figure 4.1. We show the MSE from the true treatment effect for a single estimator ($\hat{\tau}_2(x)$) and the cross-fit estimator ($\tilde{\tau}_2(x)$)

based on a 50:50 sample split. We used the R-learner as the meta-learner and create 50 Monte Carlo replications of the data using the same data generating process (DGP) which simulates a RCT and has the following properties: $N = 2000$, $X = \mathbb{R}^{10}$, $e_0(X) = 0.5$, and $\tau(x) = X_1 + \mathbb{1}(X_2 > 0) + W \quad \text{with} \quad W \sim \mathcal{N}(0, 0.5)$. Using cross-fitting decreases the MSE compared to the single estimator in about 90% of the cases. We also find that the variance is smaller compared to the single estimator.

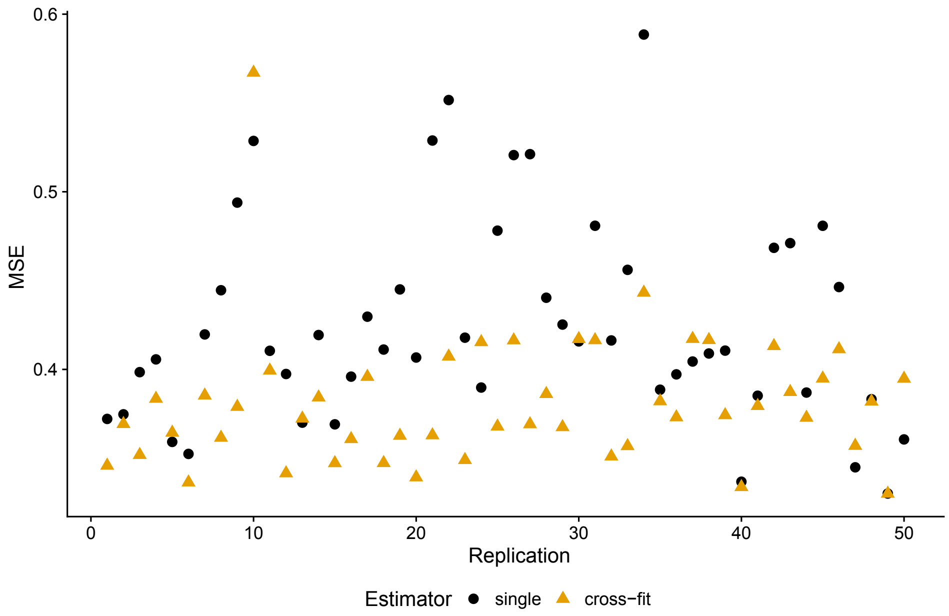


Figure 4.1: Single vs. cross-fit estimation of CATE.

Figure 4.2 illustrates the double sample splitting with cross-fitting in detail. Let fold 1 to fold 3 denote three independent samples that include the set of observations S_k as stated above. Additional subscripts indicate on which sample we train the corresponding function. The fold in brackets indicates the fold on which we predict. For example, we use fold 1 to train $\hat{e}_1(x)$ and fold 2 to train $(\hat{\mu}_0, \hat{\mu}_1)_2$. We use fold 3 for the prediction and to calculate the pseudo-outcome $\hat{\psi}_3(S_3)$. It has the same subscript as we use the same fold for the regression on the covariates. We then use all the data S to predict the CATE. The resulting estimates from the three pseudo-outcomes are averaged to get the final CATE estimate.

Splitting the sample in K folds has the consequence that only $\frac{1}{K}$ of the observations is used for estimation. Chernozhukov, Chetverikov, et al. (2018) find that this approach could lead to an unstable empirical Jacobian in the estimation sample when the sample size is small. To overcome this problem, they suggest calculating the pseudo-outcome (the residuals in the case of double machine learning) over all folds and then estimate the final parameter using all observations. We call this the combined approach and state the following notation:

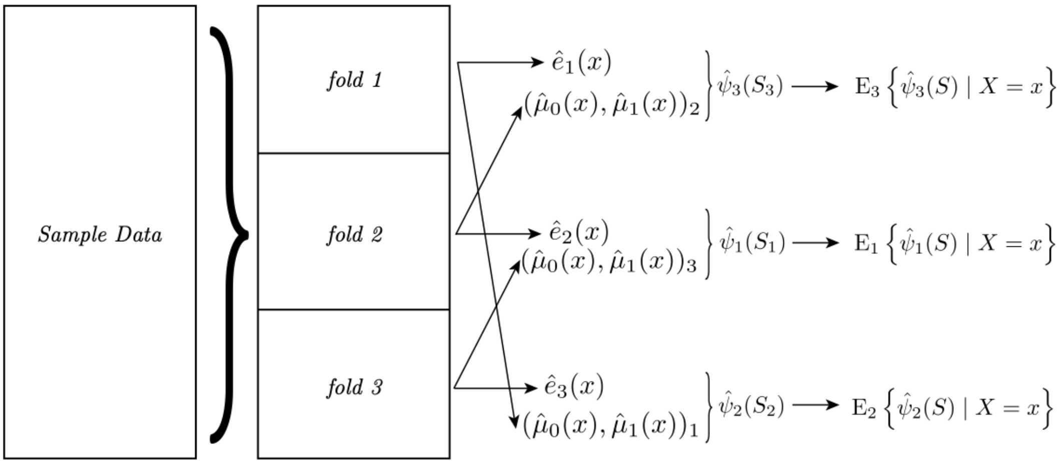

Sample Data
fold 1
fold 2
fold 3
$\hat{e}_1(x)$
$(\hat{\mu}_0(x), \hat{\mu}_1(x))_2$
$\hat{\psi}_3(S_3)$
$\mathrm{E}_3\left\{\hat{\psi}_3(S) \mid X = x\right\}$
$\hat{e}_2(x)$
$(\hat{\mu}_0(x), \hat{\mu}_1(x))_3$
$\hat{\psi}_1(S_1)$
$\mathrm{E}_1\left\{\hat{\psi}_1(S) \mid X = x\right\}$
$\hat{e}_3(x)$
$(\hat{\mu}_0(x), \hat{\mu}_1(x))_1$
$\hat{\psi}_2(S_2)$
$\mathrm{E}_2\left\{\hat{\psi}_2(S) \mid X = x\right\}$

Figure 4.2: Double sample splitting procedure and cross-fitting.

$$\hat{\hat{\tau}}(x) = \mathsf{E}[\hat{\psi}(S_c) \mid X_i = x]. \tag{4.8}$$

with $\hat{\psi}(S_c) = \{\hat{\psi}(S_1), ..., \hat{\psi}(S_K)\}$ being the combined pseudo-outcomes. Note that this is different from the naive approach where we use the whole training sample to estimate the pseudo-outcome (i.e. $\hat{\psi}(S)$).

While we can use the combined approach for any number of K splits, we only focus on 5-fold splits since in this case, we have more observations for estimating the nuisance functions and might see a bigger difference when combining the pseudo-outcomes over 5-folds vs. using $\frac{1}{5}$ for the estimation of the CATE.

4.2.3 Bias reduction due to specific sample splitting

Since we only partition our sample once in K folds, we end up with a specific sample used for estimation. Even if the splitting is random and if the specific partition has no impact on the results asymptotically, in finite samples the effect of the specific partition can lead to a bias. To see this, we show the distribution of over 50 estimates for the CATE for 3 representative observations from the test set in Figure 5.4. Different from the example above, we only create one dataset but repeat the sample splitting in fold A and fold M 50 times. Each time we estimate the CATE for each observation. The simulated data has the same properties as the cross-fit example. The blue lines show the true treatment effect for each observation. We can distinguish three different cases. First, consider the left plot: The right tail of the distribution is above zero, leading to the wrong sign of the treatment effect in some cases (the correct effect is −1 as indicated by the blue line). Through averaging (either with the mean or the median) we would neglect those wrong predictions and at least get the correct sign of the value. The plot in the middle is similar and again shows a more heavy right tail. Here we would get the correct sign and come close to the true value if we take the median instead of the mean. The plot on the right is quite normally distributed and by taking the median we would even get an unbiased estimate. The plot shows that the sample splitting plays a role in the variance in the final estimates, even if the nuisance functions are not complicated (the propensity score is a constant and the treatment effect is linear). Figure 4.11 in the Appendix shows 49 randomly selected observations and their distribution of the CATE due to sample splitting. While for most of the observations the distribution is concentrated on either positive or negative values, some observations are centered around zero and hence show positive and negative values given a particular sample split.

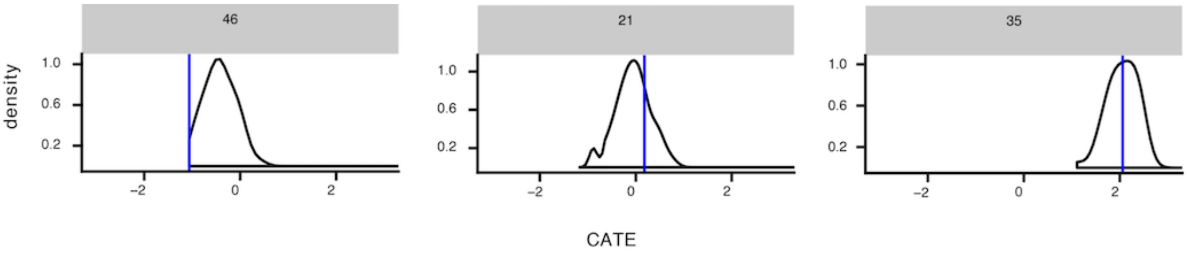


Figure 4.3: Distribution of estimated CATE for 3 selected individuals.

As suggested by Chernozhukov, Chetverikov, et al. (2018), we take the median over all iterations (B) for each observation. This may lead to a more stable conditional average treatment effect function:

$$\tilde{\tau}_{median}(x) = median\{\tilde{\tau}_b(x)\}^{1:B} \tag{4.9}$$

We neglect the subscript for the number of folds for readability. Here we show the median estimator based on cross-fitting estimators.

4.3 Simulation Study

To evaluate the performance of the estimators, we perform a Monte Carlo study in which we repeat the estimation of all parameters 300 times. In the following, we describe the data generating process and show the variations that we consider. Since simulations can be biased towards a specific setting, we consider 6 different DGPs and also two different sample sizes. We include two settings for randomized control trials (one with a balanced

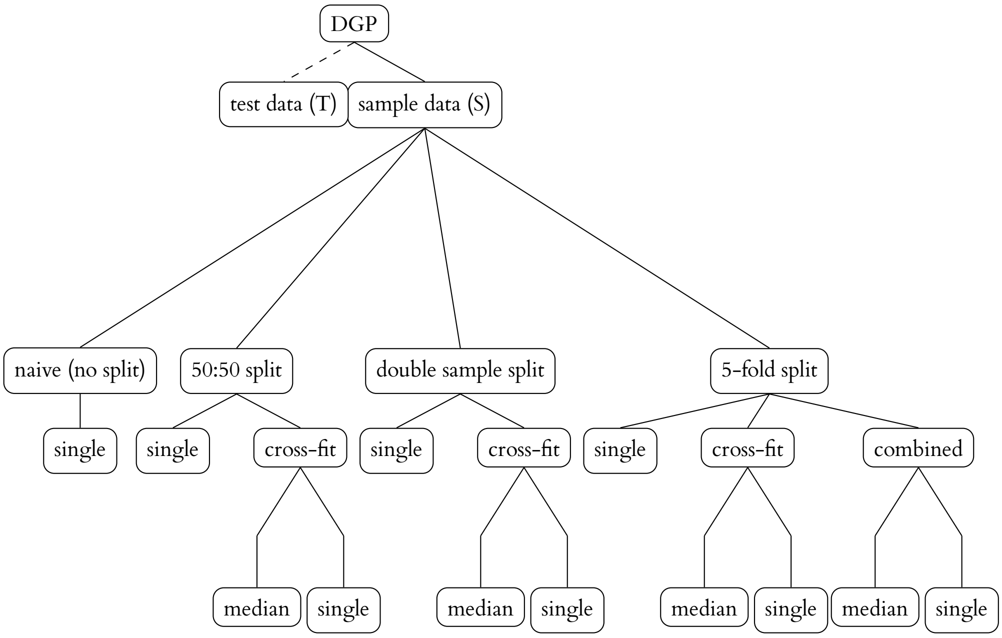


Figure 4.4: Structure of the different splitting and averaging scenarios.

treatment assignment and one where only 20% of the observations are treated). In all the other settings we assume selection into treatment and vary the difficulty to estimate the propensity score function as well as the treatment effect function. We are interested in finite sample performance and evaluate all estimators and configurations for $N = 2000$ (setting A:F) and $N = 500$ (setting G:L) observations, respectively. In total, we use 12 different settings on which we compare the estimators and meta-learners.

Figure 4.4 shows the different splitting procedures that we consider in our study. We first set aside a test set (T) of 10,000 observations which we use to compare the performance measures. We choose a higher amount of observations to limit the noise from our Monte Carlo study. We consider 100 replications of the sample data (S) for each DGP but keep the test set fixed conditional on the DGP.

The naive setting uses the whole sample data for training and estimation during all steps. In the 2-fold setting, we split the data into equal parts and use one subset for training and the remaining for estimation. When we use more than two folds, we consider two different strategies. First, we consider the setting where we split the sample data into three folds (the double sample splitting setting). Next, we assign folds to train a specific nuisance function. We group the nuisance function in 1. the propensity score, 2. the conditional mean function (either for both, the treatment and control group or separately), and 3. the regression of the pseudo-outcome on the covariates. For the X-learner, we also assign one specific fold to estimate the two imputed treatment effects. In the cross-fitting setting, we then switch the roles of the three folds and repeat the process until each fold estimated the CATE function.

When we use five-folds, we do not split the training and estimation sample into equal parts but use four out of the five folds for training the nuisance functions and the remaining fold for estimation. In the cross-fitting setting, we repeat this process until all of the five folds are used for the estimation. It is also possible to use any K-fold splitting together with the double sample splitting. We would use the same proportion $(1/K)$ for the estimation but split the remaining folds for training into two parts to using different folds for each nuisance function. Due to computational reasons, we only consider three folds for the double sample splitting approach in our analysis.

Since cross-fitting uses the whole data and hence increases efficiency, we only apply the median procedure on such cross-fitted estimators. It would also be possible to use the median based on single estimators. Preliminary simulations show that these estimators at most behave equally compared to cross-fit estimators if we take the median over at least 30 iterations. Given these results, we do not consider this class of estimators in our Monte Carlo study. Algorithm 11 in the Appendix shows the pseudo-code for the procedure of double sample splitting, cross-fitting, and taking the median.

Data generating process:

The basic model used in this simulation study is a partially linear regression model based on Robinson (1988):

$$Y = \tau(X)D + g(X) + U, \qquad \mathsf{E}[U|X, D] = 0, \tag{4.10}$$

$$D = e(X) + V, \qquad \mathsf{E}[V|X] = 0, \tag{4.11}$$

$$\tau(X) = t(X) + W \qquad \mathsf{E}[W|X] = 0, \tag{4.12}$$

with Y being a continuous outcome variable (this can also be a binary variable for all the methods we consider). $\tau(X)$ is the true treatment effect or population uplift, while D is the treatment status. The vector $X \in \mathbb{R}^p = (X_1, ..., X_p)$ consists of p different features, covariates or confounders. Let $X_i \overset{iid}{\sim} \mathcal{N}(0, \Sigma)$, where Σ is a correlation matrix U, V and W are error terms which follow a normal distribution with mean zero and variance 1 (if not specified otherwise).

Equation 5.25 is the propensity score. In the case of completely random treatment assignment, the propensity score $e(X_i) = c$ for all units $(i = 1, ..., N)$. The scalar c can take any value within the interval (0,1). In the simulation we consider $c = 0.5$ (balanced) and $c = 0.2$ (imbalanced). The imbalance in treatment assignment given a RCT can often be observed in practice since treatment is generally costly.

We assume a correlation of the covariates through a uniform distribution of the covariance matrix which is then transformed into a correlation matrix. Correlated characteristics are more common in real datasets and help to investigate the performance of ML algorithms, especially the regularization bias, in a more realistic manner.

The function $g(X)$ takes the following form:

$$g(X) = X_1 \otimes X_2 + X_3 \otimes X_4 + X_5. \tag{4.13}$$

In the simulation, we focus on different functions of the **treatment assignment**. We use the cumulative density function (CDF) of the standard normal distribution to create probabilities which are then used in a binomial function to create a binary treatment variable. The dependence of covariates within the CDF is defined as $a(X)$, for which we use a variety of functions, namely random assignment with balanced and imbalanced groups, a linear dependence, interaction terms, and non-linear dependence.

$$e_0(X) = \Phi\left(\frac{a(X) - \mu(a(X))}{\sigma(a(X))}\right), \tag{4.14}$$

$$D \overset{ind.}{\sim} \text{Bernoulli}(e_0(X)) \quad \text{such that} \quad D \in \{0;1\}. \tag{4.15}$$

$$\begin{aligned}
\text{random assigment:} &\quad e_0(X) = c \quad \text{with} \quad c \in (0,1),\\
\text{linear:} &\quad a(X) = X_1 + X_2 + X_3 - X_4,\\
\text{interaction:} &\quad a(X) = X_1 \otimes X_2 + X_3 \otimes X_4,\\
\text{non-linear:} &\quad a(X) = \sin(X_1) + \sin(X_2) + \cos(X_3 \otimes X_4).
\end{aligned}$$

The **treatment effect** takes four different settings where we also vary the degree of heterogeneity. It might not always be the case that there is significant heterogeneity in the treatment effect which is why we also consider the case of a binary effect and even no effect at all. We generate the four different settings as follows:

$$\begin{aligned}
\text{linear:} \quad & \tau(X) = 0.6\sum_{j=1}^{4} X_j + X_5 + W \quad \text{with} \quad W \sim N(0, 0.5), \\
\text{binary:} \quad & \tau(X) = \begin{cases} 0.5, & \text{if } X_p > 0 \\ -0.5, & \text{otherwise,} \end{cases} \\
\text{interaction:} \quad & \tau(X) = X_1 \otimes X_2 + X_2 + X_3 \otimes X_4 + X_5, \\
\text{non-linear:} \quad & \tau(X) = \sin(X_{1:3} \times b_{1:3}) + 1.5\cos(X_4) + X_5, \\
\text{zero:} \quad & \tau(X) = 0, \\
\text{GATE:} \quad & \tau(X) = X_5; \quad X_5 \in \{-0.5, 0, 0.5, 1\}
\end{aligned}$$

The vector $b = \frac{1}{l}$ with $l \in \{1, 2, ..., p\}$ represents weights for every covariate. Similar settings for the treatment assignment and with binary and zero treatment effects are used by Künzel et al. (2019), Nie and Wager (2020), and Powers et al. (2018).

4.4 Simulation Results

For each meta-learner, we evaluate the splitting and averaging procedures based on the mean squared error (MSE) from the test data. Following Knaus et al. (2020), we also show the absolute bias and the standard deviation. For the absolute bias and the standard deviation, we first average the predicted CATE for each individual over 300 repetitions. In order to summarise the results, we then average all three performance measures over the whole test data observations (N_T):

$$\begin{aligned}
MSE_i &= \frac{1}{R}\sum_{r=1}^{R} [\hat{\tau}(X_i)_r - \tau(X_i)]^2 \\
\overline{MSE} &= \frac{1}{N_T}\sum_{i=1}^{N_T} MSE_i
\end{aligned} \tag{4.16}$$

$$\begin{aligned}
|Bias_i| &= |\underbrace{\frac{1}{R}\sum_{r=1}^{R} \hat{\tau}(X_i)_r}_{\bar{\hat{\tau}}(X_i)} - \tau(X_i)| \\
\overline{|Bias|} &= \frac{1}{N_T}\sum_{i=1}^{N_T} |Bias_i|
\end{aligned} \tag{4.17}$$

$$SD_i = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left[\hat{\tau}\left(X_i\right)_r - \bar{\hat{\tau}}\left(X_i\right)\right]^2}$$

$$\overline{SD} = \frac{1}{N_T}\sum_{i=1}^{N_T} SD_i \tag{4.18}$$

The estimator $\hat{\tau}(X_i)$ refers to either the naive estimator without sample splitting $\hat{\tau}_{naive}(X_i)$, the single estimators without cross-fitting ($\hat{\tau}_K(X_i)$), the cross-fit estimators which takes the average over K folds ($\tilde{\tau}(X_i)$) or the median estimators ($\tilde{\tau}_{median}(X_i)$) which result only from cross-fit estimators.

Table 4.2 shows the specifications of the 15 DGP for each of the 15 settings which we refer to in the results.

Table 4.2: DGP settings

Scenarios	A	B	C	D	E
P(D = 1)	0.5	0.5	0.5	0.5	0.5
$\tau(x)$	zero	binary	GATE	linear	interaction
Scenarios	F	G	H	I	J
P(D = 1)	0.5	0.8	linear	linear	interaction
$\tau(x)$	non-lin	non-lin	linear	non-lin	binary
Scenarios	K	L	M	N	O
P(D = 1)	interaction	non-lin	non-lin	non-lin	non-lin
$\tau(x)$	interaction	linear	non-lin	GATE	zero

Notes: Simulations have 20 covariates and varying sample size (500, 2000, 4000, 8000).

Since we have 15 different data generating processes, we summarise the results by counting which estimator has the smallest MSE for each dataset, for settings 500, 2000, 4000, and 8000 observations, respectively. Table 4.3 shows that the 5-fold cross-fit estimator with median averaging outperforms all other estimators in at least 8 cases for the DR-learner and in at least 9 when using the R-learner. Overall, we find that the best performance is among the estimators which use cross-fitting and median averaging. None of the single or cross-fit only estimators shows the smallest MSE among the 60 settings and between the meta-learners. There are some settings in which the naive estimator performs best. These are primarily settings with an easy data generating process, like the two GATES settings. In all other settings the difference between the MSE from the naive and the 5-fold cross-fit estimator with median averaging is quite low (e.g. in Setting 4 and 5 the difference is 0.01). This finding suggests that using the latter estimator results in a comparable MSE. For the

T-learner, we find that the best estimator uses cross-fitting and median averaging on a 50:50 split, at least for settings up to 4000 observations. For 8000 observations we again find the 5-fold cross-fit with median averaging the best estimator. The X-learner does behave similarly to the T-learner with respect to the 50:50 split estimator.

Table 4.3: CATE: Best estimator based on MSE.

	N = 500				N = 2000				N = 4000				N = 8000			
Estimator	DR	R	T	X	DR	R	T	X	DR	R	T	X	DR	R	T	X
naive	6	2	1	1	4	2	3		1	2	4		1	2	4	
50:50																
50:50 CF																
double			—				—				—				—	
double CF			—				—				—				—	
5fold																
5fold CF																
5fold comb			—				—				—				—	
50:50 CF median	1	1	**10**	5	2	1	**8**	**8**	2		**6**	**9**	2		3	**9**
double CF median		1	—	3			—	3			—	3			—	3
5fold CF median	**8**	**11**	4		**9**	**9**	4		**9**	**10**	5		**8**	**10**	**8**	
5fold comb CF median			—	**6**	3	3	—	4	3	**3**	—	3	4	3	—	3

Notes: Averages over 15 datasets.

In Figures 4.5, 4.6, 4.7, and 4.8 we show the exact MSE for each setting and estimator, respectively for each meta-learner. Depending on the data generating process, the difference between the estimators is significantly high. For example, in setting J with 500 observations and for the DR-learner, the naive estimator has an MSE of 2.4, while the 5fold CF with median averaging estimator only has 1.7. This is a decrease of about 28%. Looking at the R-learner, we even find settings where we can decrease the MSE by about 50%. For example, with 500 observations, in setting A the MSE ranges from 0.6 to 0.1, and for setting L from 1.27 to 0.32, both are between the double and the 5-fold cross-fit with a median averaging estimator. For 500 observations, this decrease in MSE holds for at least 8 out of the 15 settings that we consider. The T-learner, however, does not show much variance between the estimators but has in general a slightly higher MSE. The smaller variance between the estimators when using the T-learner might indicate that the instability of other methods comes from the propensity score. This is because only the T-learner does not use the propensity score at all. Lastly, for the X-learner, the variance in MSE depends on whether the treatment assignment is random or observational as well as how complicated the nuisance functions are.

In general, if we simulate a randomized control trial, the variance is smallest among all meta-learners, whereas it increases if the propensity score function or the treatment effect function gets more complicated. Increasing the sample size does not change the above-observed behaviors. Even with 8000 observations, the median averaging estimators can decrease the MSE. The overall MSE decreases most for the DR-learner, especially for setting K, where we observe the highest MSE. Our findings show that the higher the sample size, the lower the MSE. Based on the results, the double sample splitting estimator often performs worst compared to other estimators with or without cross-fitting.

Now, let us look at what drives the decrease in MSE given the different estimators. To do so we plot the absolute bias in Figures 4.12, 4.13, 4.14, and 4.15, and the standard deviation in Figures, 4.16, 4.17, 4.18, and 4.19, again respectively for all meta-learners, in the Appendix. We observe that the T-learner has the highest bias among the meta-learners but the lowest standard deviation. Applying cross-fitting and median averaging on all other meta-learners can decrease the standard deviation towards the one from the T-learner. While the absolute bias differs slightly among all estimators, the standard deviation seems to be the main driver for the difference in MSE. This is especially visible for the R-, T-, and X-learner. However, it is a combination of decreased bias and variance that results in more accurate estimates when using the class of estimators with cross-fitting and median averaging.

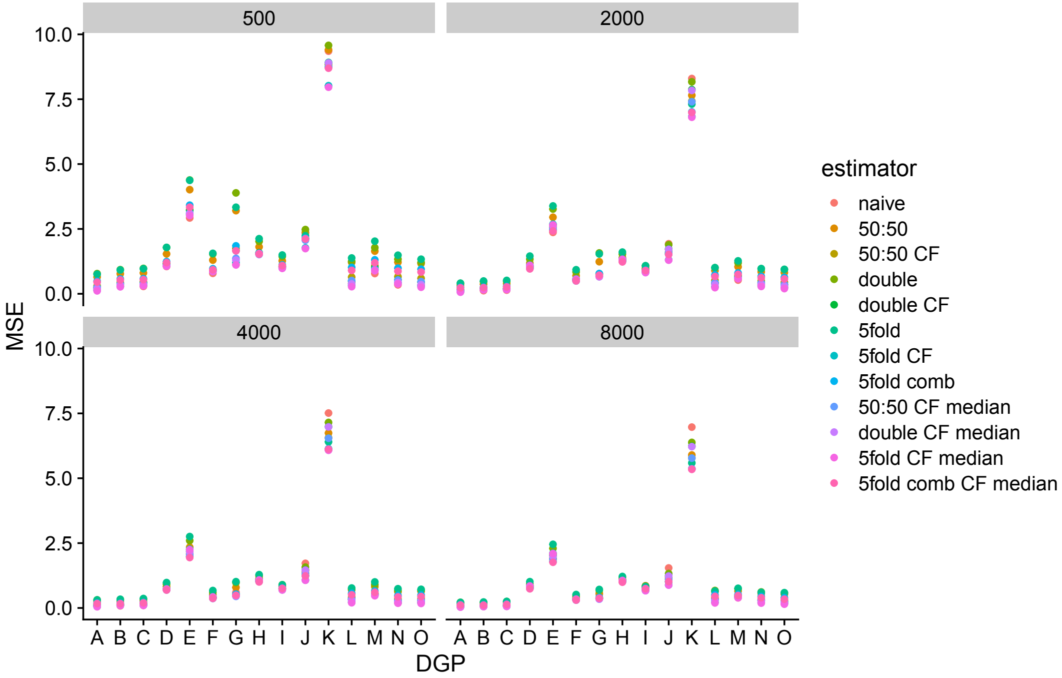


Figure 4.5: MSE from independent test set for the DR-learner.

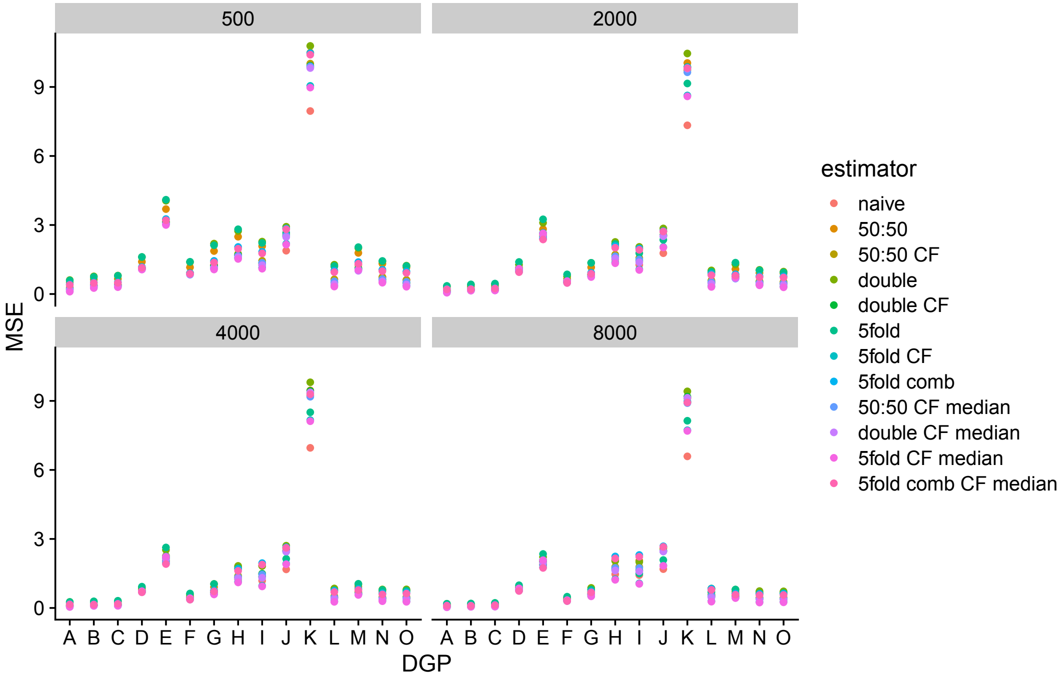


Figure 4.6: MSE from independent test set for the R-learner.

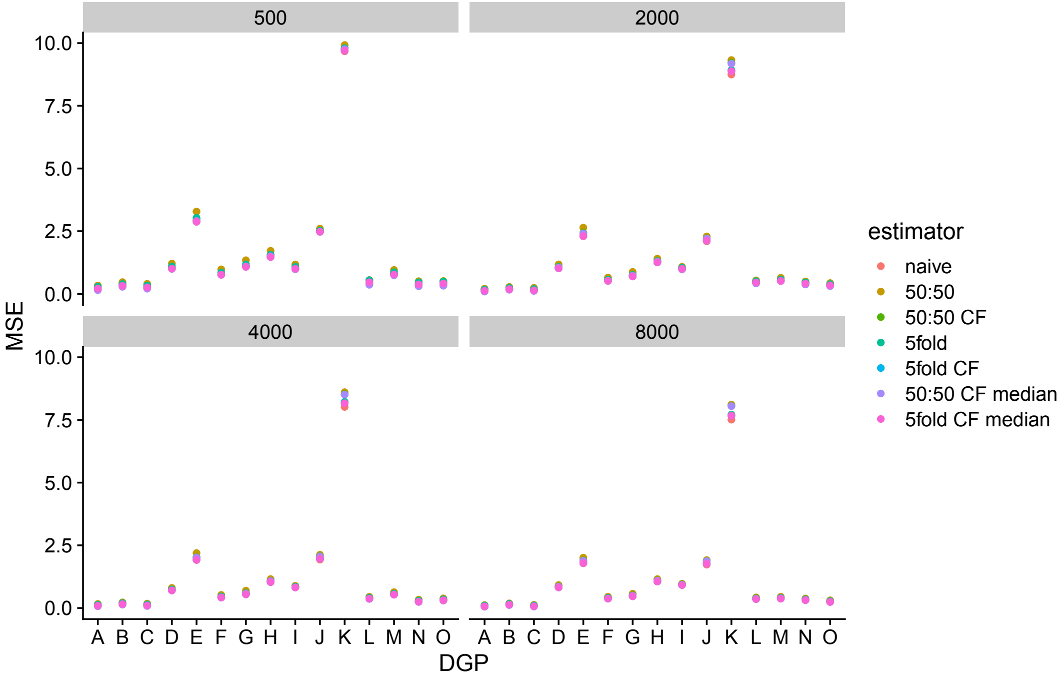


Figure 4.7: MSE from independent test set for the T-learner.

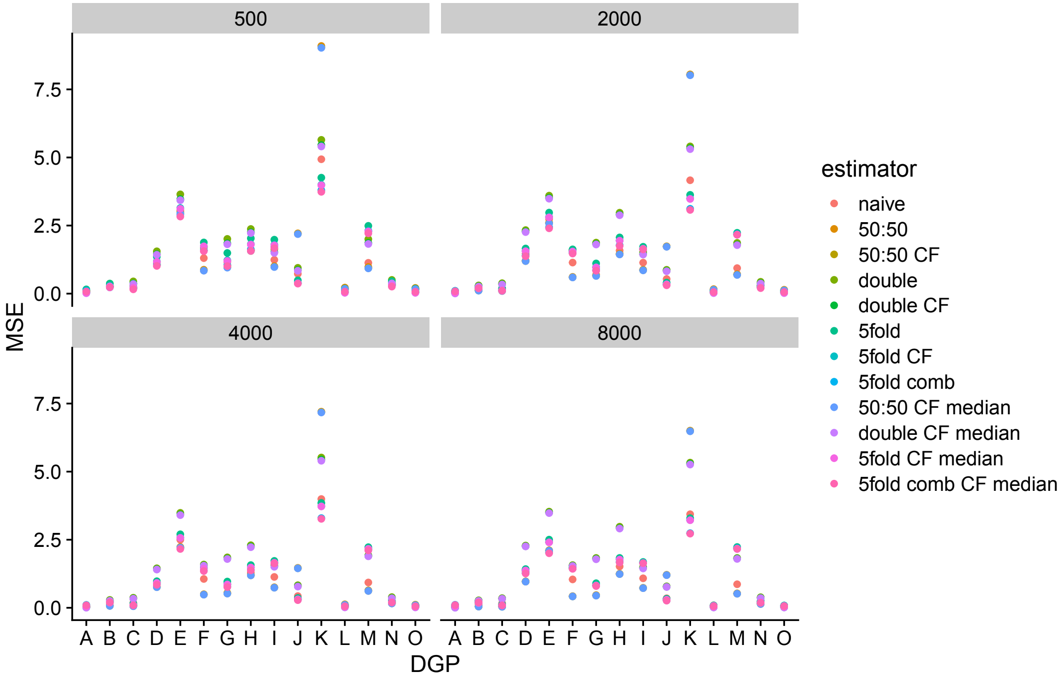


Figure 4.8: MSE from independent test set for the X-learner.

4.5 Empirical Examples

To illustrate the magnitude of applying different estimators, we consider two empirical examples. In the first example, we use data to estimate the effect of job training on real earnings. This program randomly assigned applicants to the job training program (or out of the job training program). In the second example, the treatment is not randomly assigned. Here we re-examine the effect of 401(k) eligibility, and 401(k) participation on accumulated assets. To provide a simple comparison with estimates from other papers that use the same data we show the mean of the CATE and the standard deviation. Since the CATE is the treatment effect conditional on the covariates, taking the mean can be seen as an estimate for the ATE. The SD then gives some insights into the estimated heterogeneity. For estimators that only use one iteration out of B we always show the result from the $B - th$ iteration. The same holds for estimators that use cross-fitting. Estimators that use median averaging use all results from the B iterations. The results are based on the random forest as a machine learning method for estimating the nuisance functions and for mapping the estimated treatment effect on the covariates. In the empirical analysis, we only use meta-learners on which we can apply all 12 estimators and hence neglect the T-learner.

4.5.1 Estimated effect of job training on real earnings.

In this example, we use data from the National Supported Work Demonstration (NSW) to estimate the treatment effect of job training on real earnings. The advantage of this training program is that the assignment of qualified applicants was applied randomly. Many papers have used this dataset to estimate the average treatment effect and even expanded the randomized data with observational survey data. We use the original, randomized, data with 185 treated people and 260 in the control group. This sample, whit 445 observations in total, is a sub-sample of the original data and contains information about earnings in 1974. The outcome variable is real earnings and the treatment variable is participation in the job training. To investigate heterogeneity we further use pre-treatment characteristics like age, years of schooling, an indicator variable for blacks, an indicator variable for Hispanics, marital status, an indicator for high school diploma, and two variables containing real earnings in 1974 or 1775, respectively.

Table 4.4 shows the mean and SD from the CATE estimates for all estimators and three meta-learners. Dehejia and Wahba (1999) finds an average treatment effect in the experimental data of \$1655 when controlling for all pretreatment covariates. This is in line with the results of most of our estimators. However, the mean treatment effect ranges from -1123 to 3120. Especially the estimators that use median averaging show similar results while the single estimators heavily differ from each other. Their estimate also depends on the used meta-learner. The median estimators show the smallest difference in terms of ATE

Table 4.4: Estimated effect of job training on real earnings.

	DR-learner		R-learner		X-learner	
Estimator	Mean	SD	Mean	SD	Mean	SD
naive	1,149	3,144	1,761	4,024	2,172	2,019
50:50	2,009	5,054	2,661	4,124	2,026	2,214
50:50 CF	2,268	4,368	1,621	3,378	1,742	1,944
double	2,979	6,012	530	3,015	572	967
double CF	1,860	2,930	1,547	2,578	1,243	1,551
5fold	3,120	4,197	-1,123	3,836	1,884	1,265
5fold CF	1,312	2,646	1,591	2,329	1,528	813
5fold comb	1,618	5,157	1,632	4,647	1,280	1,543
50:50 CF median	1,914	3,554	1,647	3,054	1,523	1,666
double CF median	1,888	2,876	1,589	2,329	1,207	945
5fold CF median	1,776	2,593	1,695	2,068	1,537	804
5fold comb CF median	1,726	4,911	1,713	4,542	1,635	1,434

Notes: Table shows the mean effect (ATE) and the standard deviation (SD) for the R-and the DR-learner.

compared to the findings of Dehejia and Wahba (1999). Since we show the Bth result for the single estimators, we noticed that the extreme values depend on which value out of the Bth iterations we use. For the single estimators, it is the case that there are extreme mean estimates, as the value of -1123 for the R-learner. This again shows the importance of average estimates by taking the median instead of the mean.

In Figure 4.9 we plot the densities for each estimator and three meta-learners. We group the estimators into three categories. Single estimators are those that use only one iteration and no cross-fitting. The second category is the cross-fit (CF) estimators and the last ones are those with median averaging. The findings show that the single estimators differ most within the category and across the meta-learners. We also see that the X-learner has a higher density around the mean, while the DR- and R-learner show heavier tails.

4.5.2 Effect of 401(k) eligibility and participation on net financial assets

Whether someone is eligible for a 401(k) pension plan is not randomly assigned. We use data from the 1991 Survey of Income and Program Participation and follow the argument by Poterba et al. (1995). Their strategy to treat eligibility as exogenous is to condition on a few covariates that are related to job choice (e.g. income). The data used recently in the paper by Chernozhukov, Chetverikov, et al. (2018) to estimate the ATE. We use net financial assets as the outcome variable and the eligibility to enroll in a 401(k) plan as the treatment variable. As pretreatment covariates we use age, years of education, whether a person is married, income, family size, a two-earner status dummy, a defined benefit pension status

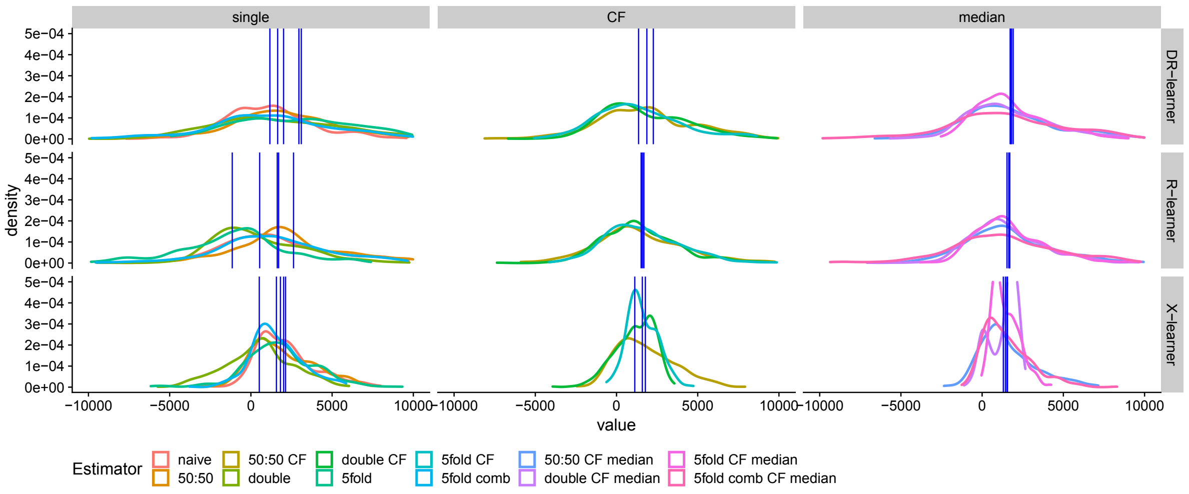


Figure 4.9: Distribution of estimated CATE for each estimator.

dummy, an individual retirement account participation dummy, and a home-ownership dummy. The dataset contains 9915 observations.

Table 4.5: Estimated effect of 401(k) eligibility on net financial assets.

	DR-learner		R-learner		X-learner	
Estimator	Mean	SD	Mean	SD	Mean	SD
naive	5,342	35,158	9,197	50,927	5,393	17,061
50:50	8,116	38,779	8,127	39,871	9,015	14,087
50:50 CF	8,459	31,972	7,704	32,263	8,745	13,465
double	7,111	37,221	8,416	41,640	2,309	10,707
double CF	7,718	26,067	8,195	24,061	1,638	5,842
5fold	10,880	34,765	3,736	25,100	5,369	20,403
5fold CF	8,598	19,824	9,064	17,404	6,660	16,184
5fold comb	8,224	53,707	8,432	52,955	6,900	21,784
50:50 CF median	8,016	31,663	8,292	32,337	8,767	12,677
double CF median	8,112	23,763	8,457	23,016	1,838	2,082
5fold CF median	8,318	18,290	9,097	17,349	7,285	16,762
5fold comb CF median	8,238	51,387	8,512	52,590	7,954	20,334

Notes: Table shows the mean effect (ATE) and the standard deviation (SD) for the R- and the DR-learner.

In Table 4.5 we show the mean and standard deviation of the conditional average treatment effect, separately using the DR-, R- and X-learner. First, it is reassuring that the mean estimates are broadly consistent with each other and are in line with the results from other papers. Depending on the meta-learner the results range from 3736 to 10880. Since we are primarily interested in heterogeneity it is worth looking at the standard deviation. We notice that our preferred estimator (the 5 fold cross-fitting plus median averaging) has the smallest SD when using the doubly-robust or the R-learner. The second smallest is produced by the 5 fold cross-fit estimator. It is also interesting to see that the more often we use the data (with cross-fitting and with averaging) the more stable the results are.

Estimators that only use one specific sample splitting show the highest variation, both in mean and in standard deviation. In this setting, the X-learner has the highest variation in terms of the mean estimates across the 12 estimators (a range of 7377), while the range for the other meta-learners is around 5000. Figure 4.10 shows the densities as in Figure 4.9. Especially for the DR-, and R-learner, we find that the variance can be decreased the more averaging we use. This result does not necessarily hold for the X-learner where we find that the double splitting estimator produces the lowest estimates even under cross-fitting and median averaging. All other estimators do, however, get closer together when moving from single, over CF towards median averaging.

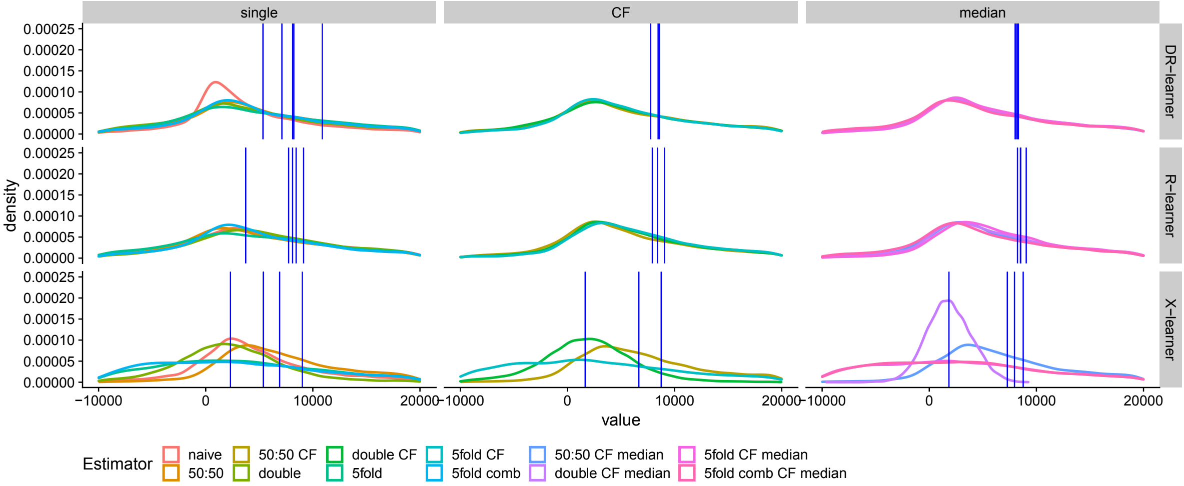


Figure 4.10: Distribution of estimated CATE for each estimator.

4.6 Discussion

This paper studies the finite sample performance of estimators based on meta-learners for the estimation of heterogeneous treatment effects. Such meta-learners rely on the estimation of different nuisance functions which facilitates different ways of sample splitting, cross-fitting and averaging. Samples can be equally or unequally split into folds. Each fold can be used to train all nuisance functions or we can select different folds for each function. Given specific folds, we can use cross-fitting to average the results. Furthermore, we can repeat this process to generate new folds and take the median over a sufficiently large number of iterations.

To study the aforementioned estimators, we generate 15 different artificial datasets that vary in their complexity of the propensity score and the treatment effect as well as the number of observations. We perform a Monte Carlo study with 300 repetitions of each DGP and evaluate each estimator on different performance measures. We further use four popular meta-learners: the DR-learner, R-learner, T-learner, and X-learner and apply all estimators based on the splitting and averaging procedures on each of them independently. For each estimator, we have 60 different settings given a specific meta-learner and 240 different settings in total to evaluate on.

For the T, DR, and R-learner, we can group our findings into three categories. First, if we have an RCT, the median estimators (50:50, double split, 5-fold, and combined) perform similarly in terms of mean MSE, mean absolute bias, and mean standard deviation, and the difference in MSE is not that high compared to the single estimators. Second, if we deviate from an RCT towards an observational study, the performance measures differ between the estimators and the learners. Using the DR-learner and R-learner we find that the 5-fold cross-fit with median averaging estimator performs best while the results for the T- and X-learner suggest using a 50:50 split or even the combined method. The effect from cross-fitting is substantial, decreasing the MSE by more than 50%.

Additional median averaging over at least 20 iterations decreases the MSE further compared to other estimators. The more complicated the functions are and the more confounding we introduce, the harder it is for the ML methods to learn the functions. This is why assigning more observations to train the nuisance functions is especially helpful in such complicated settings. Using 80% of the observations for training also decreases the bias from a particular sample. We see this since the MSE decreases less when we take the median over the 5-fold sample instead of the 2-fold sample (where we only use 50% for training).

In RCTs, we find that the performance between the T-, DR- and R-learner is competitive. This finding vanishes as soon as selection into treatment is present. Comparing all meta-learners, we find the lowest MSE (both in terms of bias and variance) for the DR-learner. The results are quite similar to findings from Kennedy (2020) for the comparison of the T and DR-learner and findings from Knaus et al. (2020) who compares the DR- and R-learner.

Based on our findings, we recommend not only rely on a specific sample split but use cross-fitting and multiple iterations over which one takes the median. In our simulation, we use a maximum of 50 iterations and find that after around 20, the result stabilizes if using the 5-fold cross-fit estimator. If we have prior knowledge about the structural form (i.e. whether the data is based on an RCT or an observational study), we should deviate from the 50:50 splitting and instead use more observations for the training of the nuisance functions. The number of observations we should use for training might also depend on how many parameters we have to tune while training the nuisance functions.

Bibliography

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *Annals of Statistics*, *47*(2), 1148–1178. https://doi.org/10.1214/18-AOS1709

Athey, S., & Wager, S. (2017). Efficient policy learning. *arXiv preprint arXiv:1702.02896.*

Bickel, P. J. (1982). On adaptive estimation. *Annals of Statistics*, *10*(3), 647–671. https://doi.org/10.1214/aos/1176345863

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, *21*(1), C1–C68. https://doi.org/10.1111/ectj.12097

Chernozhukov, V., Demirer, M., Duflo, E., & Fernández-Val, I. (2018). Generic machine learning inference on heterogeneous treatment effects in randomized experiments, with an application to immunization in india. https://doi.org/10.3386/w24678

Chernozhukov, V., Escanciano, J. C., Ichimura, H., Newey, W. K., & Robins, J. M. (2016). Locally robust semiparametric estimation. *arXiv preprint arXiv:1608.00033*. https://doi.org/10.1920/wp.cem.2016.3116

Dehejia, R. H., & Wahba, S. (1999). Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American statistical Association*, *94*(448), 1053–1062. https://doi.org/10.1080/01621459.1999.10473858

Fan, Q., Hsu, Y.-C., Lieli, R. P., & Zhang, Y. (2020). Estimation of conditional average treatment effects with high-dimensional data. *Journal of Business & Economic Statistics*, 1–15. https://doi.org/10.1080/07350015.2020.1811102

Hahn, P. R., Murray, J. S., & Carvalho, C. M. (2020). Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects. *Bayesian Analysis*, *15*. https://doi.org/10.1214/19-BA1195

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

Knaus, M. C., Lechner, M., & Strittmatter, A. (2020). Machine Learning Estimation of Heterogeneous Causal Effects: Empirical Monte Carlo Evidence. *The Econometrics Journal*. https://doi.org/10.1093/ectj/utaa014

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, *116*(10), 4156–4165. https://doi.org/10.1073/pnas.1804597116

Lunceford, J. K., & Davidian, M. (2004). Stratification and weighting via the propensity score in estimation of causal treatment effects: A comparative study. *Statistics in Medicine*, *23*(19), 2937–2960. https://doi.org/10.1002/sim.1903

Newey, W. K., & Robins, J. R. (2018). Cross-fitting and fast remainder rates for semiparametric estimation. *arXiv preprint arXiv:1801.09138*. https://doi.org/10.1920/wp.cem.2017.4117

Nie, X., & Wager, S. (2020). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*. https://doi.org/10.1093/biomet/asaa076

Poterba, J. M., Venti, S. F., & Wise, D. A. (1995). Do 401(k) contributions crowd out other personal saving? *Journal of Public Economics*, *58*(1), 1–32. https://doi.org/10.1016/0047-2727(94)01462-w

Powers, S., Qian, J., Jung, K., Schuler, A., Shah, N. H., Hastie, T., & Tibshirani, R. (2018). Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in Medicine*, *37*(11), 1767–1787. https://doi.org/10.1002/sim.7623

Robins, J. M., Zhang, P., Ayyagari, R., Logan, R., Tchetgen, E. T., Li, L., Lumley, T., & van der Vaart, A. (2013). New statistical approaches to semiparametric regression with application to air pollution research. *Research report (Health Effects Institute)*, (175), 3–129.

Robinson, P. M. (1988). Root-n-consistent semiparametric regression. *Econometrica*, 931–954. https://doi.org/10.2307/1912705

Rotnitzky, A., Robins, J., & Babino, L. (2017). On the multiply robust estimation of the mean of the g-functional. *arXiv preprint arXiv:1705.08582*.

Rubin, D. B. (1980). Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American Statistical Association*, *75*(371), 591–593. https://doi.org/10.2307/2287653

Schick, A. (1986). On asymptotically efficient estimation in semiparametric models. *Annals of Statistics*, *14*(3), 1139–1151. https://doi.org/doi:10.1214/aos/1176350055

Zimmert, M., & Lechner, M. (2019). Nonparametric estimation of causal heterogeneity under high-dimensional confounding. *arXiv preprint arXiv:1908.08779*.

Zivich, P. N., & Breskin, A. (2020). Machine learning for causal inference: On the use of cross-fit estimators. *arXiv preprint arXiv:2004.10337*.

4.A Additional Plots

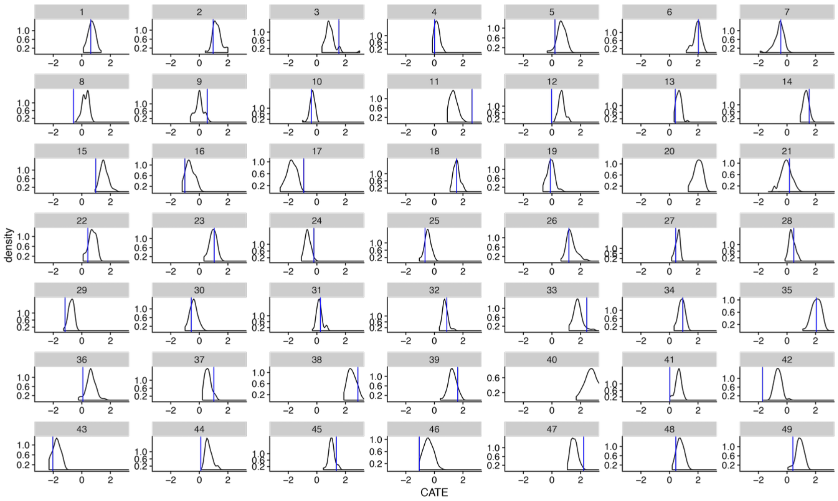


Figure 4.11: Distribution of estimated CATE for randomly selected individuals.

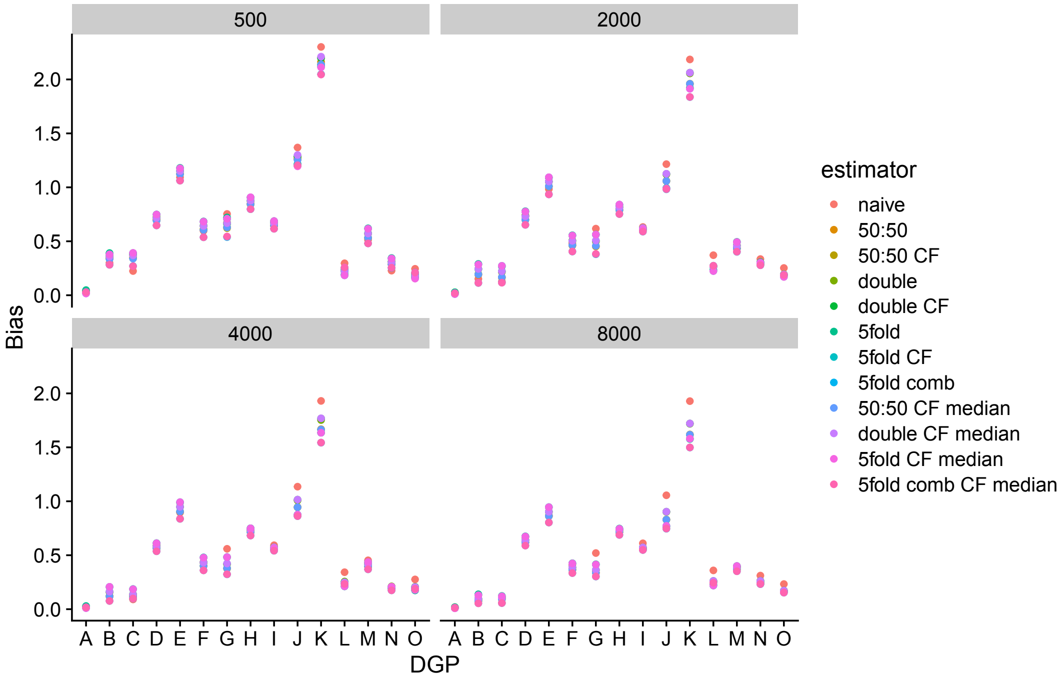


Figure 4.12: Bias from independent test set for the DR-learner.

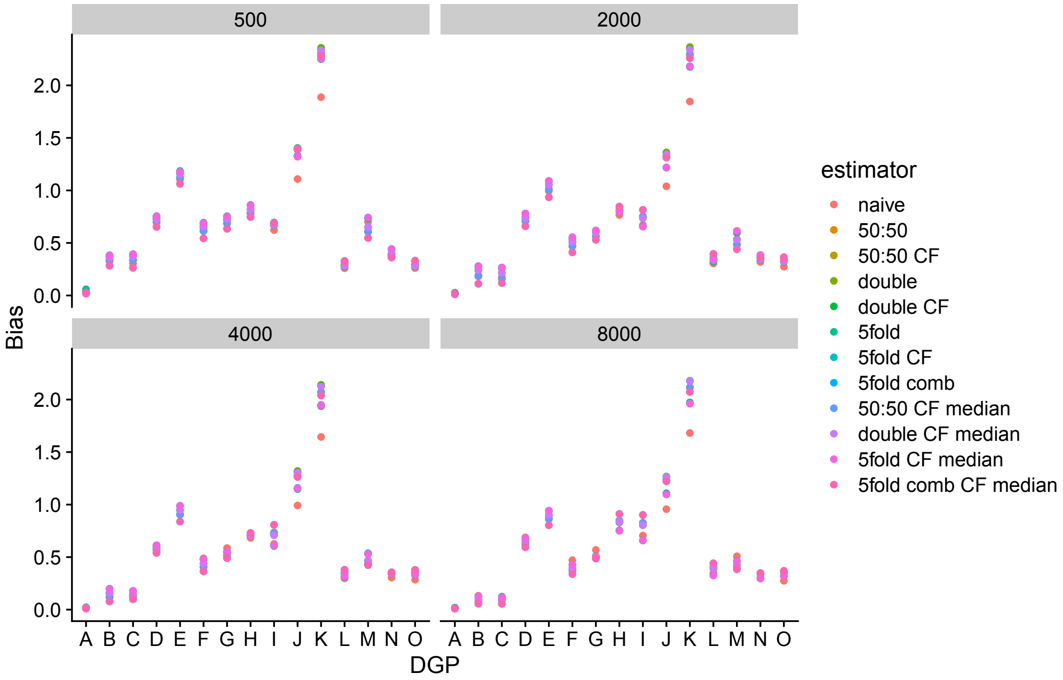


Figure 4.13: Bias from independent test set for the R-learner.

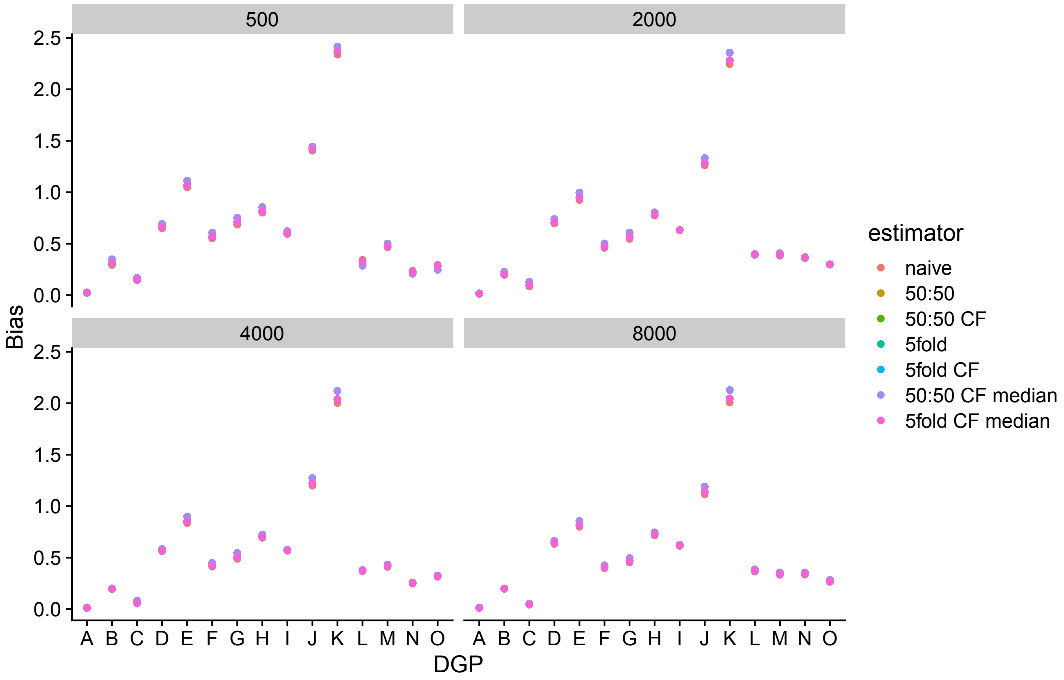


Figure 4.14: Bias from independent test set for the T-learner.

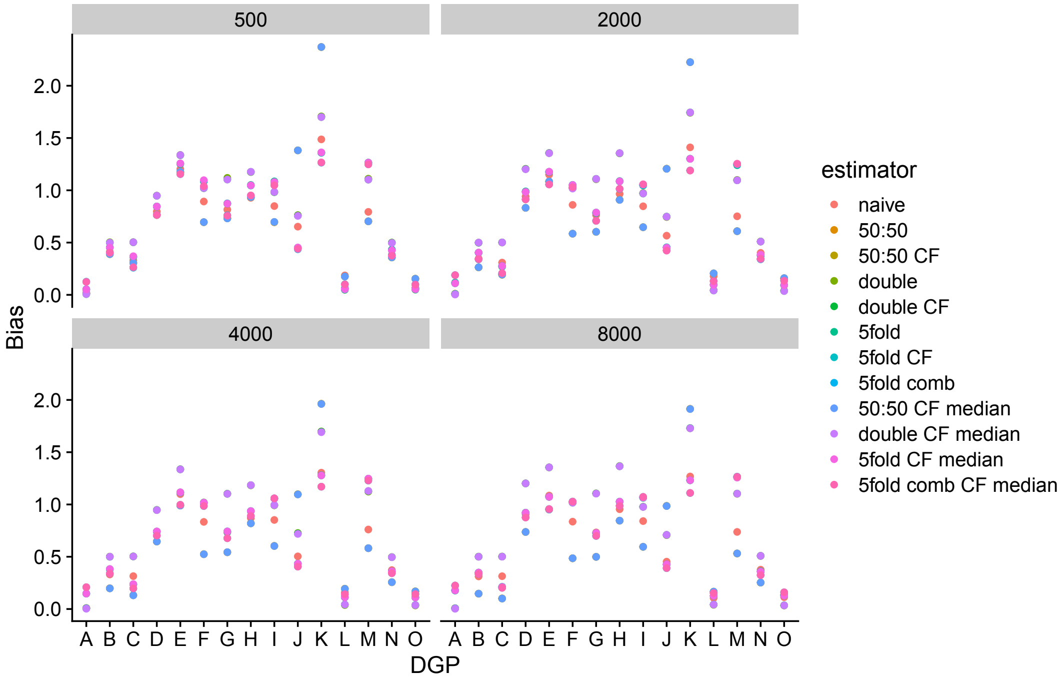


Figure 4.15: Bias from independent test set for the X-learner.

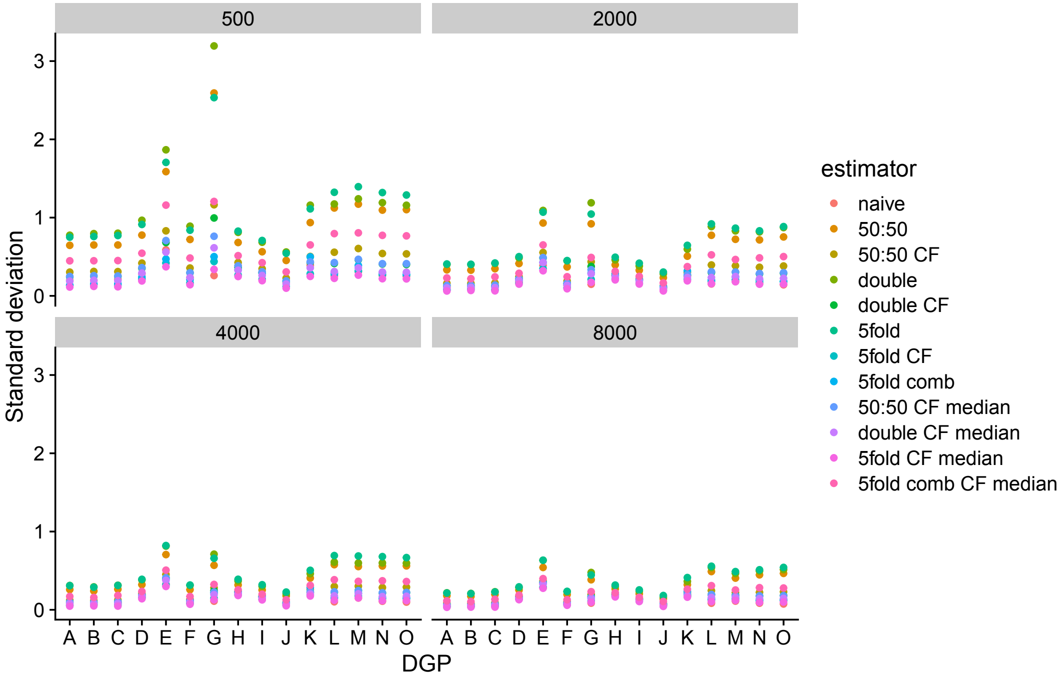


Figure 4.16: Standard deviation from independent test set for the DR-learner.

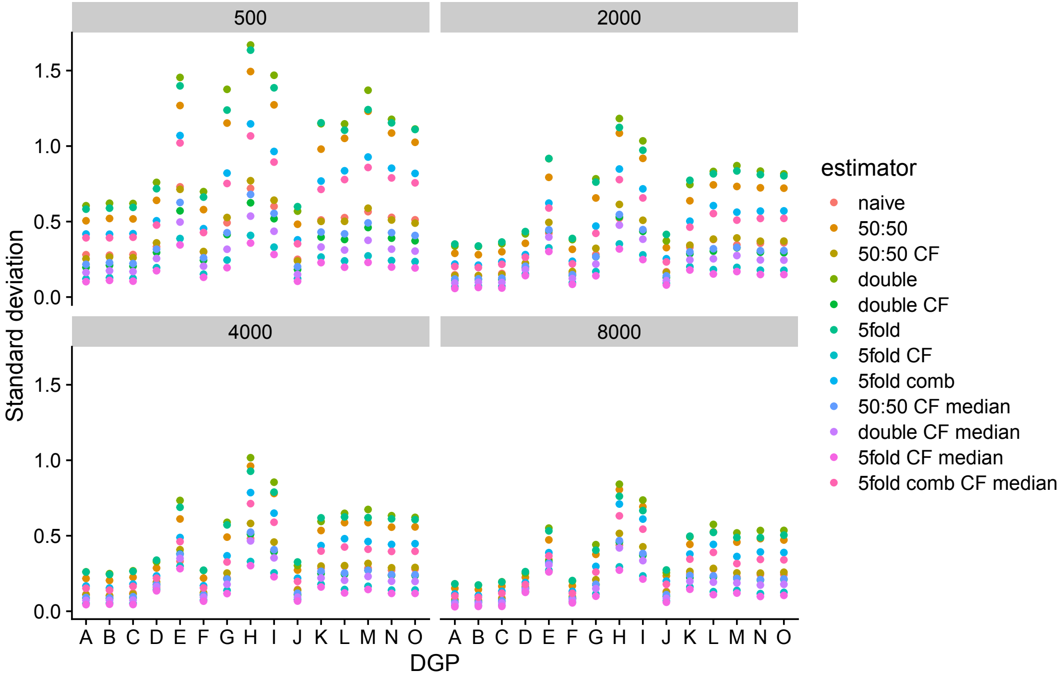


Figure 4.17: Standard deviation from independent test set for the R-learner.

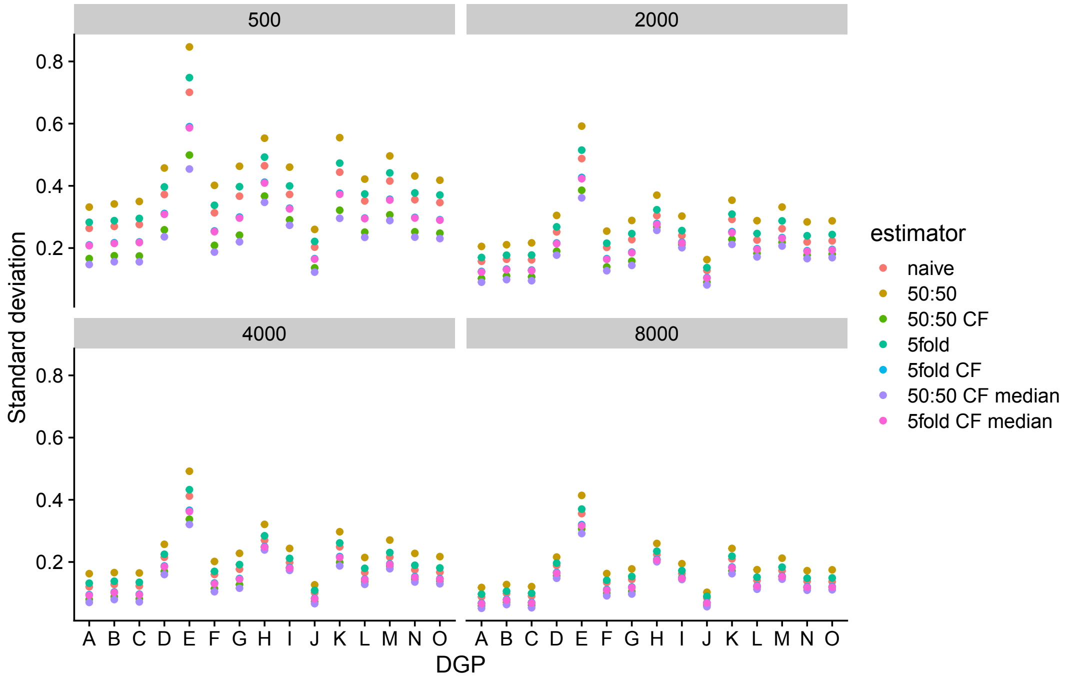


Figure 4.18: Standard deviation from independent test set for the T-learner.

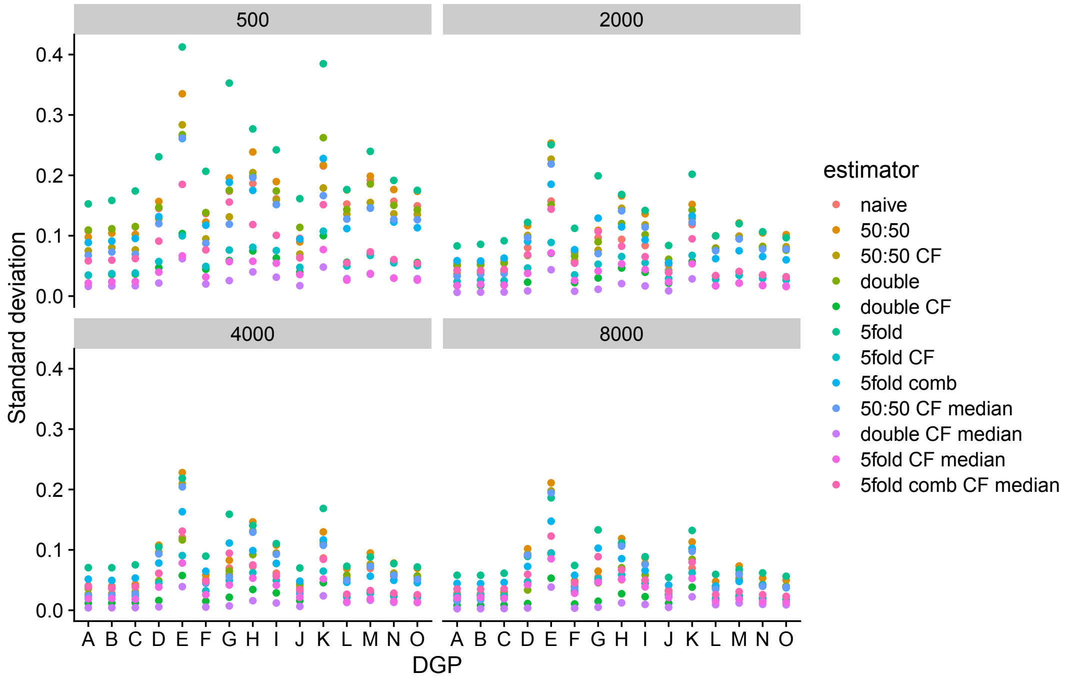


Figure 4.19: Standard deviation from independent test set for the X-learner.

4.B Tables

4.B.1 Additional tables from simulation study

Table 4.6: CATE: Best estimator based on absolute Bias.

	N = 500				N = 2000				N = 4000				N = 8000			
Estimator	DR	R	T	X	DR	R	T	X	DR	R	T	X	DR	R	T	X
naive	2	5	**11**	1		**6**	**11**		1	**6**	**13**			4	**13**	
50:50			1	**5**				2				3				**5**
50:50 CF			1	1			3	**4**				1				4
double			—				—				—	2			—	2
double CF			—	3			—				—				—	
5fold									1	1	1			1		
5fold CF	1					5							2			
5fold comb	**5**	1	—	4	**9**			—	**9**	3	—	2	**8**	**5**	—	2
50:50 CF median			2	1			3	3			1	**6**			2	1
double CF median			—				—	3			—	1			—	1
5fold CF median	3	1			**3**	2			2	1			**1**	3		
5fold comb CF median	4	**8**	—		3	2	—	3	2	4	—		4	2	—	

Notes: Averages over 15 datasets.

Table 4.7: CATE: Best estimator based on Standard Deviation.

	N = 500				N = 2000				N = 4000				N = 8000			
Estimator	DR	R	T	X	DR	R	T	X	DR	R	T	X	DR	R	T	X
naive	1				3				5				5			
50:50																
50:50 CF																
double			—				—				—				—	
double CF			—				—				—				—	
5fold																
5fold CF																
5fold comb			—				—				—				—	
50:50 CF median			**15**				**15**				**15**				**15**	
double CF median			—	**12**			—	**11**			—	**15**			—	
5fold CF median	**14**	**15**		3	**12**	**15**		4	**10**	**15**			**10**	**15**		**15**
5fold comb CF median			—				—				—				—	

Notes: Averages over 15 datasets.

Pseudo-code for the estimation procedure of double sample splitting with cross-fitting and median averaging.

Algorithm 10: Cross-fitting and averaging

```
create hold out sample: Split data in two samples: $S$ and $T$ with $S \dot{\cup} T$
for $b$=1 to $B$ do
    create folds: Split data $S$ in $K$ folds
    for $k$ in 1 to $K$ do
        regress $Y_i^d = \mu_d(X_i) + U_i$, with $i \in S_1$
            predict $\hat{Y}_i^d = \hat{\mu}_d(X_i)$, with $i \in S_3$
        regress $D_i = e(X_i) + V_i$, with $i \in S_2$
            predict $\hat{D}_i = \hat{e}(X_i)$, with $i \in S_3$
        calculate $\hat{\psi}_f$, with $i \in S_3$ (see Table 4.1 for estimators)
        regress $\hat{\psi}_f = l(X_i) + W$ with $i \in S_3$
            predict $\hat{\tau}_k(X_i) = \hat{l}(X_i)$ with $i \in T$
        cross-fitting: Assign $S_1$, $S_2$ and $S_3$ to different folds than before
    average: Take mean over $K$ folds: $\tilde{\tau}_{i,b} = \frac{1}{K} \sum \hat{\tau}_{i,k}$
average: Take median over $B$ repetitions: $\tilde{\tau}_{i,median} = median\{\tilde{\tau}_{i,b}\}$
```

Y_i^d contains either $d = D_i \in \{0, 1\}$ for the two separate conditional mean functions or all observations (like in the R-learner). The example shows the procedure for $K = 3$ folds.

Chapter 5

Group Average Treatment Effects for Observational Studies

Abstract

The paper proposes an estimator to make inference of heterogeneous treatment effects sorted by impact groups (GATES) for non-randomised experiments. The groups can be understood as a broader aggregation of the conditional average treatment effect (CATE) where the number of groups is set in advance. In economics, this approach is similar to pre-analysis plans. Observational studies are standard in policy evaluation from labour markets, educational surveys and other empirical studies. To control for a potential selection-bias, we implement a doubly-robust estimator in the first stage. We use machine learning methods to learn the conditional mean functions as well as the propensity score. The group average treatment effect is then estimated via a linear projection model. The linear model is easy to interpret, provides p-values and confidence intervals, and limits the danger of finding spurious heterogeneity due to small subgroups in the CATE. To control for confounding in the linear model, we use Neyman-orthogonal moments to partial out the effect that covariates have on both, the treatment assignment and the outcome. The result is a best linear predictor for effect heterogeneity based on impact groups. We find that our proposed method has lower absolute errors as well as smaller bias than the benchmark doubly-robust estimator. We further introduce a bagging type averaging for the CATE function for each observation to avoid biases through sample splitting. The advantage of the proposed method is a robust linear estimation of heterogeneous group treatment effects in observational studies.

JEL Classification: C01, C14, C31, C63

Replication code is available on Quantlet .

5.1 Introduction

When evaluating the causal effect of some policy, marketing action or other treatment indicator, it might not be sufficient to report the average treatment effect (ATE). The estimation of heterogeneous effects, e.g. the conditional (on covariates) average treatment effect (CATE), provides further insight into causal mechanisms and helps researchers and practitioners to actively adjust the treatment assignment towards an efficient allocation. The more information in terms of characteristics is provided, the better can heterogeneity be observed. If we have little deterministic information, it might be that heterogeneity effects are overlooked. The trade-off here is that the more covariates datasets have, the more complex they get. We might not know the structural form of our functions of interest or we simply have more covariates than observations. Even if the number of raw covariates is smaller than the number of observations, including quadratic or even higher order interaction increases the number of covariates and can easily exceed the sample size. This is why parametric models are often insufficient when applied on high-dimensional, non-linear datasets (Chernozhukov, Chetverikov, et al., 2018).

Under the assumption of approximate sparsity, meaning that out of our rich covariate space only a few are dependent on our variables of interest (the outcome and the treatment assignment), recent methods for treatment effect estimation use machine learning models that have shown to be superior in high-dimensional prediction problems (Hastie et al., 2009). The idea is to learn nuisance functions and regularize the parameter space while making as little assumptions as possible. This is especially helpful when the data does not come from randomised experiments where treatment is randomly assigned to the individuals. In observational studies, self-selection into treatment can arise and introduce a bias that has to be corrected for (i.e. self-selection bias) (Heckman et al., 1998). For the ATE one would use the nuisance parameter to orthogonalize the effect that covariates have on both, the treatment assignment and the outcome variable. See Chernozhukov, Chetverikov, et al. (2018) for a recent approach, which they call double machine learning.

Methods to estimate the CATE are, among others, the generalized random forest, which builds on the idea of controlling for observed confounders through a tree structure and then estimates the CATE using the estimates from each leafs (Athey et al., 2019). Another approach is causal boosting, which uses the idea of causal trees as a weak learner (Powers et al., 2018). What the aforementioned methods lack, however, is that they are built on tree algorithms and therefore do not allow a flexible estimation of heterogeneous treatment effects in terms of the underlying algorithm (e.g. LASSO or Neural Networks). A recent method called R-learner does provide such flexibility and shows competitive performance in the estimation of the CATE (Nie & Wager, 2020). Other models, known as meta-learners, decompose the modelling procedure into sub-regression functions, which can be solved using any supervised learning method. This can, for example, be done by a two-model approach (TMA) where a response function (conditional mean) on the treated and another one on the non-treated observations is estimated. The difference between the two functions can thus be interpreted as the CATE (Künzel et al., 2019). This approach falls into the category of indirect estimation since its goal is to minimize the squared error loss based on the outcome variable. A more efficient approach is to directly estimate the treatment effect and regularize based on the treatment effect itself rather than the observed outcome. This leads to a reduced variance of the prediction (Hitsch & Misra, 2018). In order to do this, we first have to estimate a proxy function of the treatment effect. The literature refers to this approach as the treatment effect projection (Hitsch & Misra, 2018) or the modified outcome method (Knaus et al., 2020). The idea is to use inverse probability weighting or the doubly-robust estimator as proposed by (Robins & Rotnitzky, 1995). Using the estimates from the two-model approach in combination with inverse probability weighting (IPW) decreases the variance of the estimator and controls for observed confounding (see e.g. Lunceford and Davidian (2004)). Additional orthogonalization using the two conditional mean functions produced by the TMA further decreases the bias of the parameter of interest (Lee et al., 2017). The doubly-robust estimator can be used in high-dimensional settings to estimate a reduced dimensional conditional average treatment effect function. Functional limit theory can be derived for the case where the nuisance functions are trained via machine learning methods, which are then applied on the doubly-robust estimator (Fan et al., 2020; Zimmert & Lechner, 2019). Recent papers study and evaluate different models that are designed for the estimation of heterogeneous treatment effects (see e.g. Knaus (2020), Knaus et al. (2020), Künzel et al. (2019), and Powers et al. (2018).

The difficulty, however, is that machine learning methods are often a black box that is not easy to interpret. This fact hinders the information on drivers for effect heterogeneity. In this paper, we, therefore, build on the ideas of Chernozhukov, Demirer, et al. (2018), who concentrate on estimating group average treatment effects (GATE) in randomised experiments. The groups are built on the distribution from the CATE (e.g. quantiles to get five groups). A semi-parametric model is then used to identify the best linear predictor for the group treatment effect, providing standard errors and confidence intervals. The heterogeneity between the groups can further be interpreted through covariates, which shed some light on the question of what characteristics determine the differences between groups. In this paper, we extend the approach to estimating the GATE parameter towards the use in observational studies and also towards the possibility to estimate a best linear CATE based on group heterogeneity. In a paper that is very close to ours the authors use a linear model with the aforementioned orthogonalization step to estimate group treatment effects based on observed covariates (Park & Kang, 2019). They define the groups by conditioning on observed covariates, for example, for a binary variable they estimate the average treatment effect for the two levels of the variable.

The advantage of the method proposed here is a robust estimation of group heterogeneous treatment effects that is comparable with other models thus keeping its flexibility in the choice of machine learning methods and at the same time its ability to interpret the results. The latter is especially useful in all areas of empirical economics like policy or labour markets interventions. It also has the advantage to control for potential self-selection bias. The idea of going beyond the average, but not as deep as to estimate conditional average treatment effects for many covariates, is first considered in Chernozhukov, Fernández-Val, et al. (2018). They provide standard errors and confidence bands for the estimated sorted group effects and related classification analysis and provide confidence sets for the most and least affected groups. While they only use parametric estimators, a non-parametric attempt to estimate group average treatment effects and also provide insights from the heterogeneity in terms of observed covariates is proposed by Fan et al. (2020) and Zimmert and Lechner (2019). They use a two-step estimator of which the second step consists of a kernel estimator.

Our contribution is to use the doubly-robust estimator and thus keep the flexibility in using any kind of machine learning method to learn the nuisance parameter in the first step. In a second step, we use Neyman-orthogonal scores to cancel out the effect that covariates can have on both, the outcome and the treatment selection. Using the residuals, we set up a linear model to learn the group average treatment effects. When grouping the heterogeneous treatment effect, we find that we can get more precise estimates in terms of mean absolute error (MAE) and bias with our proposed method. Especially the linearization increases the accuracy of our estimates. We also show that the CATE function, which we estimate via a doubly-robust estimator, should be weighted over multiple iterations. This type of bagging further decreases the MAE of the CATE function. We split our paper in three main parts. First, we state the methodology for randomized experiments and second, the extensions for observational studies. Third, we employ a extensive simulation study that include selection bias, high-dimensionality and non-linearity in the data generating process.

5.2 Generic Machine Learning for Group ATE

5.2.1 Potential Outcome Assumptions

Throughout this paper, we make use of the potential outcome theorem (Rosenbaum & Rubin, 1983) and state four necessary assumptions. The first assumption is the ignorability of treatment, conditional on observed covariates (X), from the two potential outcomes. It is also known as unconfoundedness or simply conditional independence:

$$\left(Y_i^1, Y_i^0\right) \perp\!\!\!\perp D_i|X_i. \tag{5.1}$$

With Y^1 denoting the potential outcome under treatment and Y^0 if not being treated. D is the treatment assignment variable.

The second assumption, the Stable Unit Treatment Value Assumption (SUTVA), guarantees that the potential outcome of an individual is unaffected by changes in the treatment assignment of others. This assumption might be violated if individuals can interact with each other (peer and social effects).

The third assumption, called overlap, guarantees that for all $x \in supp(X)$, the probability of being in the treatment group (i.e. the propensity score, $e_0(X)$), is bounded away from 0 and 1:

$$\begin{aligned} 0 < \mathsf{P}(D = 1|X = x) < 1. \\ e_0(X) = \mathsf{P}(D = 1|X = x). \end{aligned} \tag{5.2}$$

We control for the common support by estimating the propensity score and balance the treatment and control group based on the distribution. We hence exclude all observations that have a propensity score lower 0.02 or higher than 0.95. The fundamental problem of causal inference is that we only observe one of the two potential outcomes at the same time. The counterfactual for a nontreated (treated) person, namely, what would have happened if this person were (not) treated, is always missing. We can represent this statement through a switching regression where the observed outcome (Y_i) depends on the two potential outcomes and the treatment assignment:

$$Y_i = Y_i^0 + D(Y_i^1 - Y_i^0). \tag{5.3}$$

We further assume that, for the estimation of standard errors, the following moments exist: $\mathsf{E}\left[|Y^j|^q\right] < \infty$ for $q \geq 4$ and $j = 0, 1$.

5.2.2 Randomized Control Trial

To provide valid estimation and inference for a causal interpretation of parameters, Chernozhukov, Demirer, et al. (2018) focus on features of the CATE. One of the main features is the **Sorted Group Average Treatment Effect (GATES)**. The idea is to find groups of observations depending on the estimated treatment effect heterogeneity. Their proposed method relies on a two-model approach in the first step. Here, two response functions are trained separately for the treated and non-treated observations. This approach will be biased if the data sample is from an observational study. In randomized control trials, the difference between the two functions provides an estimate of the treatment effect for every observation. We refer to $S(X)$ as our proxy-predictor:

$$\begin{aligned} \tau(X) &= \mathsf{E}[Y|D=1,X] - \mathsf{E}[Y|D=0,X], \\ S(X) &= g_1(X) - g_0(X). \end{aligned} \tag{5.4}$$

Here $g_D(X) = \mathsf{E}(Y|D,X)$ is the regression model of the outcome variable on X separately for $D \in \{0,1\}$. The two functions can be estimated with a broad range of supervised machine learning methods. The target parameters are

$$\mathsf{E}[\tau(X)|G_k] \qquad\qquad G_k := \{S(X) \in I_k\}, \quad k = 1,\dots,K, \tag{5.5}$$

where G is an indicator of a group membership with $I_k = [\ell_{k-1}, \ell_k)$ and ℓ_k is the k/K-quantile of $\{S_i\}_{i\in M}$. Subscript M denotes that these are all out-of-sample predictions. We will for readability not always refer to the sets but always make use of sample splitting when using machine learning methods. The groups are ex-post defined by the predicted score function in the first stage. Figure 5.1 shows an illustration of how the GATE parameter are defined.

If the treatment effect for the groups are consistent, it asymptotically holds that

$$\mathsf{E}[\tau(X)|G_1] \leqslant \mathsf{E}[\tau(X)|G_2] \leqslant \dots \leqslant \mathsf{E}[\tau(X)|G_k], \tag{5.6}$$

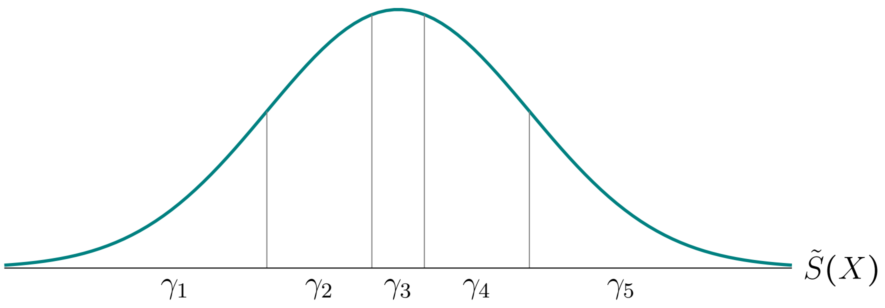


Figure 5.1: Illustration of a treatment effect function and how we define the GATE.

which is the monotonicity restriction. Furthermore, it can be tested whether there is a homogeneous effect if $\mathsf{E}[\tau(X)|G_k]$ would be equal for all k groups. The weighted linear projection equation to recover the GATES parameter is:

$$YH = \beta^{\top} A_1 H + \sum_{k=1}^{K} \gamma_k \cdot \mathbf{I}(S(X) \in I_k) + \nu, \tag{5.7}$$

with $A_1 = (1, B(X))$ and $B(X) = \mathsf{E}[Y|D = 0, X]$ being the baseline function without treatment. See pseudo-code of Algorithm 11, which describes the implementation of this method. The weights H represent the Horvitz-Thompson transformation (Horvitz & Thompson, 1952):

$$H = H(D, X) = \frac{D - e(X)}{e(X)(1 - e(X))}. \tag{5.8}$$

This estimator, which is applied to account for different proportions of observations within strata in a target population, is equivalent to the simple inverse probability weighting estimator. These estimators, however, might exhibit a high variance if the identification (the precision) of the propensity scores is lacking (Lunceford & Davidian, 2004).

The main identification result is that the projection coefficients γ_k can be represented in the following way:

$$\gamma = (\gamma)_{k=1}^{K} = (\mathsf{E}[\tau(X)|G_k])_{k=1}^{K}. \tag{5.9}$$

Algorithm 11: GATES

```
for $b$=1 to $B$ do
    Split Data in $k = 2$ samples: $I^a$ and $M$ with $I^a \uplus M$
    Train $Y_i^0 = g_0(X_i, D = 0) + U_{0i}$, with $i \in I^a$
    Train $Y_i^1 = g_1(X_i, D = 1) + U_{1i}$, with $i \in I^a$
        Predict $\hat{Y}_i^0 = \hat{g}_0(X_i)$, with $i \in M$
        Predict $\hat{Y}_i^1 = \hat{g}_1(X_i)$, with $i \in M$
            Calculate $S_b(X|i) = \hat{Y}_i^1 - \hat{Y}_i^0$
    Train $D_i = e_0(X_i) + V_i$, with $i \in I^a$
        Predict $\hat{D}_i = \hat{e}(X_i)$, with $i \in M$
            Calculate $\hat{V}_i = D_i - \hat{e}(X_i)$, with $i \in M$
    Estimate GATES parameters ($\gamma$) with weighted OLS using $M$ (see equation
     5.7)
Average $\gamma$ over $B$ iterations: $\tilde{\gamma} = median\{\gamma\}$
```

The algorithm is based on Chernozhukov, Demirer, et al. (2018).

There are two potential sources of uncertainty. One is estimation uncertainty regarding our parameter of interest, keeping sample splitting fixed. The second source is exactly due to the sample splitting. To account for this, the p-values, as well as the confidence intervals, need to be adjusted. (Chernozhukov, Demirer, et al., 2018) show, that sample-splitting-adjusted p-values can have the following form.

$$\mathsf{P}(p_A \leq \alpha/2|\text{Data}) \geq 1/2, \tag{5.10}$$

with p_A being the realized p-value given the auxiliary sample and α is the significance level. Given that we use medians to average the parameters γ_k over B bootstrap repetitions it holds that for at least 50% of the random data splits out of B, the p-value is at largest $\alpha/2$. Small values provide evidence that the group parameter is different from zero.

5.2.3 Extensions for Observational Studies

To use the best linear predictor for group heterogeneity in observational studies, we change and extend the TMA in the first step and the linear model in the second step. First, we replace the two-model approach by a doubly-robust estimator. This means we not only weight by the inverse of the propensity score but also orthogonalize the outcome variable by subtracting the conditional mean. We also make use of sample splitting as a form of cross-fitting. The auxiliary sample is applied to estimate the score function via the doubly-robust estimator and the main sample to predict the final score function, which is used in the linear step. In this way, we limit the danger of overfitting. The resulting function is a more robust version of the CATE for each individual as well as for the GATE function. The two steps are described in more detail in the following.

The function in equation 5.12 is calculated using the training data (the I^a sample). In a second step, a new supervised model is trained on the transformed outcome using I^a while predictions are made on the test set M to get an unbiased estimate (see equation 5.12. Algorithm 12 describes this process.

$$\hat{S}_i = \hat{\mu}_1(X_i) - \hat{\mu}_0(X_i) + \frac{D_i(Y_i - \hat{\mu}_1(X_i))}{\hat{e}(X_i)} - \frac{(1 - D_i)(Y_i - \hat{\mu}_0(X_i))}{(1 - \hat{e}(X_i))} \tag{5.11}$$

$$\hat{S}_i = l(X_i) + \omega \tag{5.12}$$

In equation 5.11, $\hat{\mu}_1(X_i) - \hat{\mu}_0(X_i)$ is equivalent to the score-function from the two-model approach. Simulation evidence from Knaus et al. (2020) and Powers et al. (2018) suggests that estimators based on $\hat{S}_i$ might be more stable because of the doubly-robust property and that the performance is competitive for the estimation of heterogeneous treatment effects in observational studies. The doubly-robust property states that the estimator is consistent and unbiased if only one of the models, the regression or the propensity score, is correctly specified (Robins & Rotnitzky, 1995; Robins et al., 1994). Belloni et al. (2013), Lunceford and Davidian (2004), and Williamson et al. (2014) study the theoretical properties of the doubly-robust estimator and highlight implications for practice. One of the findings is that the variance can be decreased when using the doubly-robust estimator instead of a simple inverse probability estimator (Lunceford & Davidian, 2004). Chernozhukov and Semenova (2018) show that equation 5.11 is conditionally locally robust to the estimation error of the nuisance parameter.

Next we state some asymptotic results to recover the CATE. From equation 5.4 it follows that

$$\tau(X) = \mathsf{E}\{\mathsf{E}[Y|D = 1, X] - \mathsf{E}[Y|D = 0, X]|X = x_i\} \tag{5.13}$$

Let $\eta(X) := (e(X), \mu_1(X_i), \mu_0(X_i))$ be the true high dimensional nuisance parameters. Following Fan et al. (2020) we can define

$$\psi(D,Y,X,\eta(X)) = \mu_1(X_i) - \mu_0(X_i) + \frac{D_i(Y_i - \mu_1(X_i))}{e_0(X_i)} - \frac{(1-D_i)(Y_i - \mu_0(X_i))}{(1-e_0(X_i))} \tag{5.14}$$

.

Theorem 2.1
(i) under Assumptions 1,2,3,4

$$\begin{aligned} \mathsf{E}\left[\mu_1(X_i) + \frac{D_i(Y_i - \mu_1(X_i))}{e_0(X_i)} | X = x_i\right] &= \mathsf{E}\left[Y^1 | X = x_i\right], \\ \mathsf{E}\left[\mu_0(X_i) + \frac{(1-D_i)(Y_i - \mu_0(X_i))}{1 - e_0(X_i)} | X = x_i\right] &= \mathsf{E}\left[Y^0 | X = x_i\right] \end{aligned}$$

(ii)$\mathsf{E}\left[\psi(D,Y,X,\eta(X)) - \tau(X) | X = x_i\right] = 0$ given (i).

This moment condition satisfies the Neyman-orthogonality condition. It is a key component in ensuring that the CATE estimators are robust to the regularization bias inherent for the nuisance functions, which are learned via machine learning models.

Next, we set up a linear model for the estimation of the low-dimensional parameters of interest γ_k. Suppose that the data generating process follows a partial linear model that has the following form:

$$Y = \tau(X)D + g_0(X) + U, \qquad E[U|X,D] = 0, \tag{5.15}$$

$$D = e_0(X) + V, \qquad E[V|X] = 0, \tag{5.16}$$

$$\mathsf{E}[\tau(X)] = \frac{1}{K}\sum_{k=1}^{K}\gamma_k(X). \tag{5.17}$$

We define γ as before as $\gamma = (\gamma)_{k=1}^{K} = (\mathsf{E}[\tau(X)|G_k])_{k=1}^{K}$. The outcome variable Y depends not only on the treatment effect parameter but also on observed covariates through the function $g_0(X)$. The second equation displays the setting in observational studies, namely that the treatment assignment also depends on observed covariates through the function $e_0(X)$. In randomized control trials, it is sufficient to directly learn the regression function

$\hat{\tau}(X)D$. We do not need to include the function $\mu_0(X)$ since in RCT the distribution of the covariates are assumed to be the same for both, the control and the treatment group. The linear model in section 5.2.2 does exactly this but also takes into account that the treatment assignment might be different for strata in the covariate space.

In observational settings, confounding through covariates leads to a bias that we have to account for. We wish to partial out the effect from X on D as well as the effect from X on Y. Following Chernozhukov, Chetverikov, et al. (2018), we again make use of Neyman-orthogonal moments, which leads to the orthogonalized regressors $\hat{U} = Y - \hat{\mu}(X)$ and $\hat{V} = D - \hat{e}(X)$. The terms $\hat{U}$ and $\hat{V}$ are the residuals that we use in the linear projection function in Equation 5.19. The propensity-score estimates $(\hat{e}(X))$ can be derived by using the main sample on the already estimated propensity-score function, which is used in the doubly-robust step. The function $\hat{\mu}(X)$ is estimated using any machine learning model on the auxiliary sample. For the estimation of the average treatment effect $\hat{\tau}$, the residualized regression function has the following form:

$$\hat{\tau} = \left(\frac{1}{N} \sum_{i \in M} \hat{V}_i \hat{V}_i \right)^{-1} \frac{1}{N} \sum_{i \in M} \hat{V}_i (Y_i - \hat{\mu}(X_i)). \tag{5.18}$$

In our case, we are not specifically interested in the average treatment effect but the average effect given the quantiles of the CATE function. This leads to the linear projection equation to estimate the group average treatment effect using the main sample of observations:

$$(Y - \mu_0(X)) = \sum_{k=1}^{K} \gamma_k \cdot (D - e_0(X)) \cdot \mathbf{I}(S(X) \in I_k) + \nu. \tag{5.19}$$

Let $\tilde{S}(X) = \hat{l}(X)$ and rewrite the empirical analog for the k-th specific group as:

$$\hat{\gamma}_k = \left(\frac{1}{N} \sum_{i \in M} \hat{V}_i \hat{V}_i \cdot \mathbf{I}(\tilde{S}(X_i) \in I_k) \right)^{-1} \frac{1}{N} \sum_{i \in M} \hat{V}_i \left(Y_i - \hat{\mu}(X_i)\right) \cdot \mathbf{I}(\tilde{S}(X_i) \in I_k). \tag{5.20}$$

Through orthogonalization, we can overcome the regularization bias, present when naively employing ML models to estimate the parameter γ in this setting. Note that we can use different ML algorithms for each function. A trade-off to regularization bias is overfitting, which can be taken care of through sample splitting. However, through sample splitting, we can only use $N - n$ observations, given that n observations are in the training data. To account for the uncertainty through sample splitting, we run multiple repetitions via bootstrapping and split the sample randomly in each bootstrap. We average the GATES parameter by taking the median over B repetitions.

Naturally, we can do the same sort of bagging for the score-function. Given that we split our sample into two equal parts (one for training and one for estimation), we get estimates of the CATE for half of the observations in each iteration. If we repeat this procedure often enough, we get estimates for every observation and decrease the uncertainty bias due to sample splitting. We then take medians for each observation to get the final CATE estimate ($\bar{S}_i(X)$):

$$\bar{S}_i(X) = median\{\hat{\hat{S}}_b(X)\}. \tag{5.21}$$

In our simulation study we show exactly how bagging for CATE can influence the results.

Algorithm 12 shows the steps to identify the group treatment effect and the steps to average the CATE function over all repetitions.

```
Algorithm 12: Extended GATES
for $b$=1 to $B$ do
    Split Data in $k = 2$ samples: $I^a$ and $M$ with $I^a \cup M$
    Train $Y_i^0 = g_0(X_i, D = 0) + U_{0i}$, with $i \in I^a$
    Train $Y_i^1 = g_1(X_i, D = 1) + U_{1i}$, with $i \in I^a$
    Train $D_i = e_0(X_i) + V_i$, with $i \in I^a$
    Train $Y_i = \mu(X_i) + U_i$, with $i \in I^a$
        Predict $\hat{Y}_i^0 = \hat{g}_0(X_i)$, with $i \in I^a$
        Predict $\hat{Y}_i^1 = \hat{g}_1(X_i)$, with $i \in I^a$
        Predict $\hat{D}_i = \hat{e}(X_i)$, with $i \in I^a$
        Predict $\hat{Y}_i = \hat{\mu}(X_i)$, with $i \in M$
            Calculate doubly-robust estimator (see equation 5.11)
            Train $\hat{S}_i = l_0(X_i) + W$ with $i \in I^a$
            Predict $\tilde{S}_i = \hat{l}(X_i)$ with $i \in M$
                Calculate $\hat{V}_i = D_i - \hat{e}(X_i)$, with $i \in M$
                Calculate $\hat{U}_i = Y_i - \hat{\mu}(X_i)$, with $i \in M$
                Store $S_b^*(X)|i = \tilde{S}_i(X)|b$
    Estimate GATES parameters ($\gamma$) with OLS using $M$ (see equation 5.19)
Average $\gamma$ over $B$ iterations: $\tilde{\gamma} = median\{\gamma\}$
Calculate Density for every $i$: $\hat{\hat{S}}(X)|i$ given $S_b^*(X)|i$ over $b$
Calculate Final score-function ($\bar{S}_i(X)$) given density of medians for i = 1 to N
```

The number of groups can, of course, be increased to e.g. 10. In empirical settings, it would depend on the sample size. If we want to have at least 30 observations within a group, we could have $\frac{N}{30 \times \Lambda}$ groups, with Λ-splits or folds of the dataset in the first stage used for training and testing. Here we consider only two-folds. However, there is no general relationship between the number of folds in cross-fitting and the precision of the estimator (see Chernozhukov, Chetverikov, et al. (2018) for an example with different folds).

5.2.4 Alternative Grouping: The Baseline Effect

In this paper, we focus on group average treatment effects that are based on quantiles from the CATE function. In the linear model we, therefore, condition on these quantiles to estimate the GATE parameters. The condition on which the sample is split into groups can, of course, be changed. Of particular interest could be the GATE based on different levels of the baseline outcome (Y^0). This has some similarity with the traditional "Quantile Treatment Effect", which is defined as $\delta(\lambda) = F_{Y^1}^{-1}(\lambda) - F_{Y^0}^{-1}(\lambda) \quad 0 < \lambda < 1$, where $F_{Y^D}^{-1}$ is the unconditional quantile function and λ a specific quantile. Here, however, we only focus on Y^0.

Let us think of Y as being a binary variable, indicating if a customer bought something online or if a person participated in a program (e.g. unemployment training). We might believe that the treatment effect differs among different baseline probabilities. In such cases, we want to find group treatment effects based on quantiles of Y^0. As an estimator for Y^0, we can use the doubly-robust estimator and only concentrate on the baseline potential outcome:

$$\mathsf{E}\left[g_0\left(X_i\right) + \frac{\left(1 - D_i\right)\left(Y_i - g_0\left(X_i\right)\right)}{1 - e_0\left(X_i\right)}|X = x_i\right] = \mathsf{E}\left[Y^0|X = x_i\right] \tag{5.22}$$

The linear model would then have the following form:

$$(Y - \mu_0(X)) = \sum_{k=1}^{K} \gamma_k \cdot (D - e_0(X)) \cdot \mathbf{I}(Y^0 \in I_k) + \nu. \tag{5.23}$$

5.3 Simulation Study

To evaluate the proposed extensions i) doubly-robust first stage, ii) orthogonal semi-parametric second stage, and iii) bagging of CATE, we use simulated data where the true treatment effects are known. In the following, we describe the data generating process (DGP) and show the variations that we consider. In all simulations we consider five groups denoted by γ. For example, γ_1 is the mean from the lower $\frac{100}{K}\%$ quantile from the CATE function while γ_K is the mean of the upper $\frac{100}{K}\%$ quantile. Our method, which we call DO GATES (we refer to the name DO GATES as for Double Orthogonal GATES) estimates γ_k quantiles from a linear model. As a benchmark model, we use the mean-effect from each of the K-quantiles from the final treatment effect function ($\bar{S}(X)$) estimated via the doubly-robust estimator. Below, we show an example of the distribution from the estimated CATE function. A natural question is how the mean for the K-quantile groups, directly build from the CATE function, behaves without the second linear orthogonalization. To answer this, we compare the estimates from the benchmark and our approach with the mean values of $K = 5$ quantiles from the true treatment effect function. For the non-parametric estimation of functions, we use a random forest method (*R-package: ranger*) as the corresponding machine learning algorithm due to its fast performance.

In this simulation, we are interested in finite sample results and estimate our group treatment effects for $N = 2000$ and again for $N = 500$ observations.

5.3.1 Data Generating Process

The basic model used in this simulation study is a partially linear regression model based on Robinson (1988):

$$Y = \tau_0(X)D + g_0(X) + U, \qquad E[U|X, D] = 0, \tag{5.24}$$
$$D = e_0(X) + V, \qquad E[V|X] = 0, \tag{5.25}$$
$$\tau(X) = t_0(X) + W \qquad E[W|X], = 0, \tag{5.26}$$

with Y being a continuous outcome variable. $\tau_0(X)$ is the true treatment effect or population uplift, while D is the treatment status. The vector $X = (X_1, ..., X_p)$ consists of p different features, covariates or confounder. U, V and W are unobserved covariates, which follow a random normal distribution $= N(0, 1)$.

Equation 5.25 is the propensity score. In the case of completely random treatment assignment, the propensity score $e_0(X_i) = c$ for all units $(i = 1, ..., N)$. The scalar c can take any value within the interval (0,1). In the simulation we consider $c = 0.5$ (balanced) and $c = 0.8$ (imbalanced).

The function $g_0(X)$ takes the following form:

$$g_0(X) = X_{p/2} + X_{p/10} + X_{p/4} \times X_{p/10}. \tag{5.27}$$

The vector $b = \frac{1}{l}$ with $l \in \{1, 2, ..., p\}$ represents weights for every covariate.

In the simulation, we focus on different functions of the **treatment assignment**. We use a CDF to create probabilities, which are then used in a Binomial function to create a binary treatment variable. The dependence of covariates within the normal distribution function is represented as $a(X)$, for which we use a variety of functions, namely random assignment with balanced and imbalanced groups, a linear dependence, interaction terms, and non-linear dependence.

$$e_0(X) = \Phi\left(\frac{a(X) - \mu(a(X))}{\sigma(a(X))}\right), \tag{5.28}$$

$$D \overset{ind.}{\sim} \text{Bernoulli}(e_0(X)) \quad \text{such that} \quad D \in \{0;1\}. \tag{5.29}$$

$$\begin{aligned}
\text{random assigment:} \quad & e_0(X) = c \quad \text{with} \quad c \in (0,1),\\
\text{linear:} \quad & a(X) = X_2 + X_{p/2} + X_{p/4} - X_8,\\
\text{interaction:} \quad & a(X) = X \times b + X_{p/2} + X_2 + X_{p/4} \times X_8,\\
\text{non-linear:} \quad & a(X) = X \times b + \sin(X_{p/2}) + X_2 + \cos(X_{p/4} \times X_8).
\end{aligned}$$

The **treatment effect** is heterogeneous and generated with a linear or a non-linear dependency of X:

$$\text{linear:} \quad \tau_u(X) = X_1 + X_2 > 0 + W \quad \text{with} \quad W \sim N(0, 0.5), \tag{5.30}$$

$$\text{non-linear:} \quad \tau_u(X) = \sin(X \times b) + X_{5+p/2}. \tag{5.31}$$

We standardise the treatment effect within the interval $[0.1, 1]$:

$$\tau_0(X) = \frac{\tau_u(X) - min(\tau_u(X))}{max(\tau_u(X)) - min(\tau_u(X))}(1 - 0.1) + 0.1. \tag{5.32}$$

Our aim is to perform a Monte Carlo simulation, in which we fix the distribution and dependence of the covariates (correlation). This is necessary to also fix the treatment effect function for each setting. All other functions can have small deviations within each repetition due to the error term and the random generation for the binomial distribution (for the treatment assignment). The idea is to gather samples to approximate the distribution of the true treatment effect. Figure 5.6 in the Appendix shows the true treatment effect function for the linear and non-linear case. They are quite normally distributed without heavy tails.

5.3.2 Simulation Results

In Table 5.1 and 5.2, we show the different scenarios, specifically for the treatment assignment mechanism. We also consider a simulation where both, the $\hat{e}(X)$ function as well as $\hat{g}_D(X_i)$ (and as a consequence also $\hat{\mu}(X)$) is misspecified. We model misspecification by introducing an unobserved confounder in the DGP (namely X_2), which we exclude in the observed dataset. In scenarios A to F, the data generation process for the treatment effect depends linear on the covariates, whereas in scenarios G to L the dependency is non-linear (see equations 5.30 and 5.31). In Table 5.1 we consider N = 2000 observations. In Table 5.2, we set N =500. All other processes are the same. We repeat every DGP 100 times and estimate the mean absolute error (MAE) over all groups and repetitions. Another performance measurement is the squared bias as a comparison between the expected function and the true function. We aggregate our measurement as follows:

$$MAE_{kj} = |\hat{\gamma}_k - \gamma_k|, \tag{5.33}$$

$$MAE_k = \frac{1}{S}\sum_{j=1}^{S}(MAE_{kj}), \tag{5.34}$$

$$MAE = \frac{1}{K}\sum_{k=1}^{K}(MAE_k). \tag{5.35}$$

$$Bias^2(\hat{\gamma}_k) = (\underbrace{\mathsf{E}[\hat{\gamma}_k]}_{\frac{1}{S}\sum_{j=1}^{S}\hat{\gamma}_{kj}} - \gamma_k)^2, \tag{5.36}$$

$$Bias^2 = \frac{1}{K}\sum_{k=1}^{K}\left(Bias^2(\hat{\gamma}_k)\right). \tag{5.37}$$

Table 5.1: Settings and Monte Carlo averages for N = 2000.

Scenarios	A/G	B/H	C/I	D/J	E/K	F/L
N	2000	2000	2000	2000	2000	2000
$\mathbb{R}^p$	20	20	20	20	20	20
$\mathsf{P}(D = 1)$	0.5	0.2 (imbalanced)	linear	interaction	non-linear	linear
Misspecification	No	No	No	No	No	Yes
MAE CATE	**0.06/0.05**	0.15/0.14	0.62/0.61	0.66/0.67	0.61/0.61	0.91/0.92
MAE DO GATES	0.09/0.10	**0.08/0.10**	**0.33/0.32**	**0.32/0.36**	**0.32/0.31**	**0.75/0.76**
$Bias^2$ CATE	**0.00**/0.00	0.03/0.03	0.47/0.44	0.49/0.50	0.46/0.44	0.99/1.00
$Bias^2$ DO GATES	0.01/0.00	**0.00/0.00**	**0.15/0.15**	**0.17/0.18**	**0.15/0.13**	**0.74/0.75**

Notes: Mean absolute error (MAE) of the average treatment effect (estimated from the quantile estimates) over S = 100 Monte Carlo simulations. Setting G to L consists of a non-linear treatment effect function.

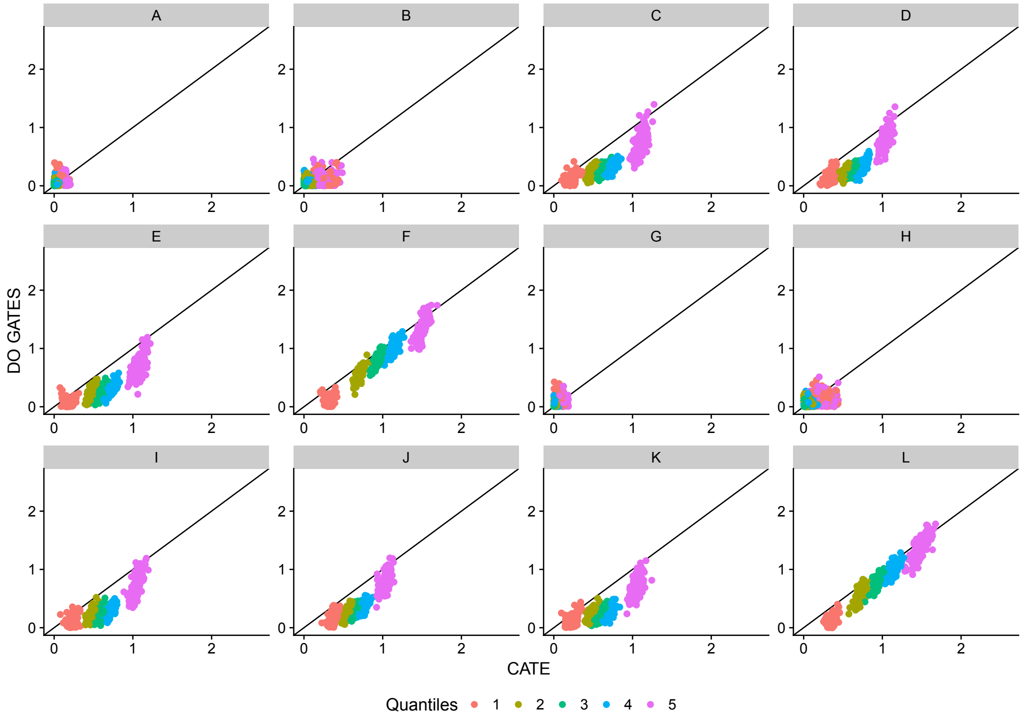


Figure 5.2: Group estimate from the CATE function and DO GATES. Groups are coloured by quantiles (1 = least affected, 5 = most affected). Axes show the absolute error between estimates and true quantile treatment effects. The 45-degree line indicates the equality of both methods. The number of observations equals 2000.

In Figure 5.2 and 5.3 we plot the results for each group of the simulation study for N = 2000 and 500 observations, respectively. We find that especially the lower quantiles show a smaller absolute error of the quantile treatment effect. If the treatment assignment mechanism is randomized, our model performs competitively or slightly worse than the doubly-robust estimator, as can be seen from scenario A, B, G and H. In all observational settings, we find a smaller mean absolute error from our DO GATES method compared to the benchmark model. We find that the highest error from the true treatment effect arises when both models are misspecified as in scenario F and L. There is, however, no significant difference in the MAE if we change the dependency of the treatment effect from linear to non-linear.

The bias behaves like the MAE in the sense that only for the balanced randomization the two models show equal bias. In all other settings, we find a smaller bias for our method and again the highest differences when there is selection bias, which depends linear and or with interactions on the treatment. Table 5.3 and 5.4 in the Appendix show the MAE for each group and DGP over all repetitions. We also notice that the deviation between the two methods is highest for the lowest group and decreases towards the upper quantiles. When we set N = 500, we find that the MAE and the bias increase in both methods but the overall structure stays the same as for N = 2000.

Table 5.2: Settings and Monte Carlo averages for N = 500.

Scenarios	A/G	B/H	C/I	D/J	E/K	F/L
N	500	500	500	500	500	500
$\mathbb{R}^p$	20	20	20	20	20	20
$P(D=1)$	0.5	0.2 (imbalanced)	linear	interaction	non-linear	linear
Misspecification	No	No	No	No	No	Yes
MAE CATE	**0.12/0.14**	**0.16**/0.20	1.13/1.10	1.22/1.21	1.11/1.10	1.44/1.41
MAE DO GATES	0.19/0.16	0.21/**0.19**	**0.64/0.63**	**0.71/0.69**	**0.62/0.64**	**1.13/1.10**
$Bias^2$ CATE	**0.02**/0.02	0.03/0.05	1.47/1.38	1.60/1.54	1.43/1.40	2.34/2.29
$Bias^2$ DO GATES	0.03/0.02	**0.02/0.00**	**0.67/0.65**	**0.60/0.59**	**0.63/0.66**	**1.67/1.69**

Notes: Mean absolute error (MAE) of the average treatment effect (estimated from the quantile estimates) over S = 100 Monte Carlo simulations. Setting G to L consist of a non-linear treatment effect function.

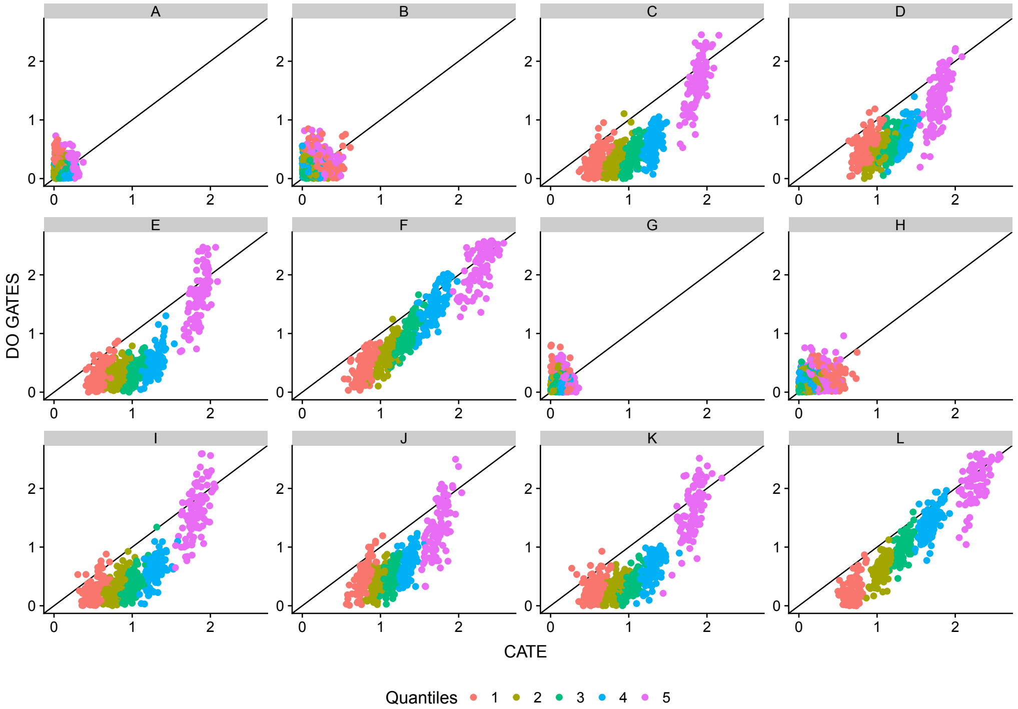


Figure 5.3: Group estimate from the CATE function and DO GATES. Groups are coloured by quantiles (1 = least affected, 5 = most affected). Axes show the absolute error between estimates and true quantile treatment effects. The 45-degree line indicates the equality of both methods. The number of observations equals 500.

5.3.3 Model Averaging of CATE

Figure 5.4 shows the estimates from the conditional average treatment effect over $B = 100$ bootstrap iterations (we split the sample once in equal parts). For every iteration, we either get an estimate for a specific observation i if this observation is in the test-sample or we don't which we label with a "NA" estimate. After 100 iterations we have at least 39 estimates for each observation. If the specific sample we use for training would always be equal in terms of distributions, we would assume that every iteration estimates the same value for a specific observation. Here we show that this is not the case since the sample splitting does matter, especially in finite sample settings.

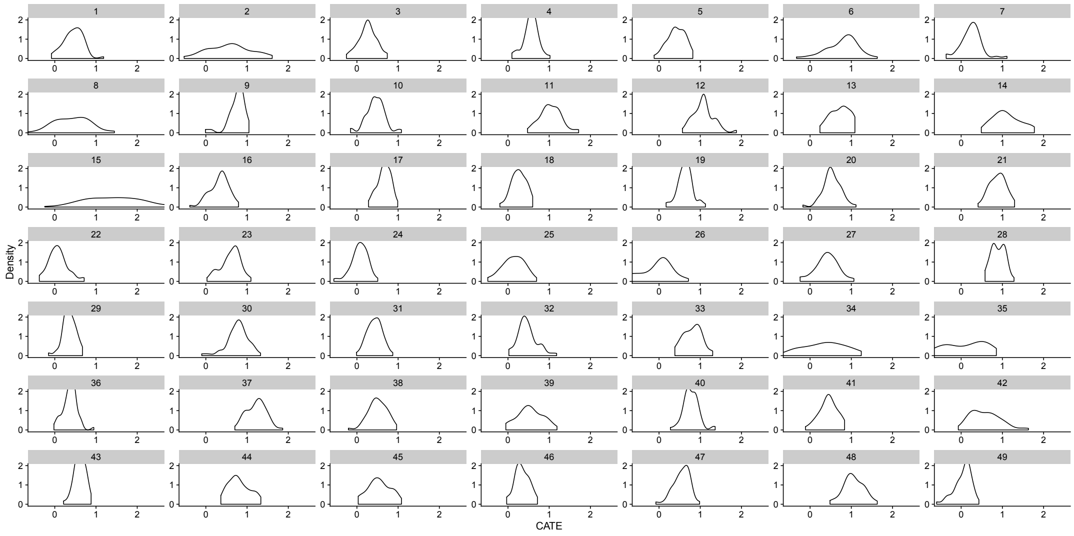


Figure 5.4: Distribution of score function ($\hat{\hat{S}}_b(X)$) for each of 49 randomly selected individuals. Density is estimated over 100 iterations. Due to random sample splitting, the number of estimates for each individual is between 39 and 59 for 100 iterations.

The simulated data, in this case, has the following properties. $N = 1000$, $X = \mathbb{R}^{20}$, $e_0(X) = 0.5$ and $\tau_u(X) = X_1 + \mathbb{1}(X_2 > 0) + W$ with $W \sim N(0, 0.5)$. We standardize τ_u to $\tau(X) \in [0.1, 1]$. The densities for 49 selected observations show that even in randomised experiments, the point estimates differ due to the sample splitting in the first step. We propose to average the estimates by taking the median over each iteration. This leads to a more stable conditional treatment effect function:

$$\bar{S}_i(X) = median\{\hat{\hat{S}}_b(X)\} \tag{5.38}$$

We show in Figure 5.5 how the absolute error (AE) for each group depends on the number of iterations. The upper plot shows the AE for the CATE function and the lower plot for the DO GATES method. The AE for the CATE function stabilizes after around 50 iterations while for the DO GATES we can decrease the AE further as we take the median over more iterations. Interestingly, it seems that the variance in AE for the DO GATES method does not mainly depend on the CATE function as we can see after taking the median of 50 iterations. The CATE function is quite stable but the GATES still show some variance in the AE. It could be that this is caused in the orthogonalization step where we train the two conditional mean functions. We also try to average with the arithmetic mean and find no significant difference between the two approaches. However, it could be that in some scenarios, like a very small sample size, single estimates are far away from the true estimate and should be treated as outliers. Therefore, we suggest to use the median for averaging.

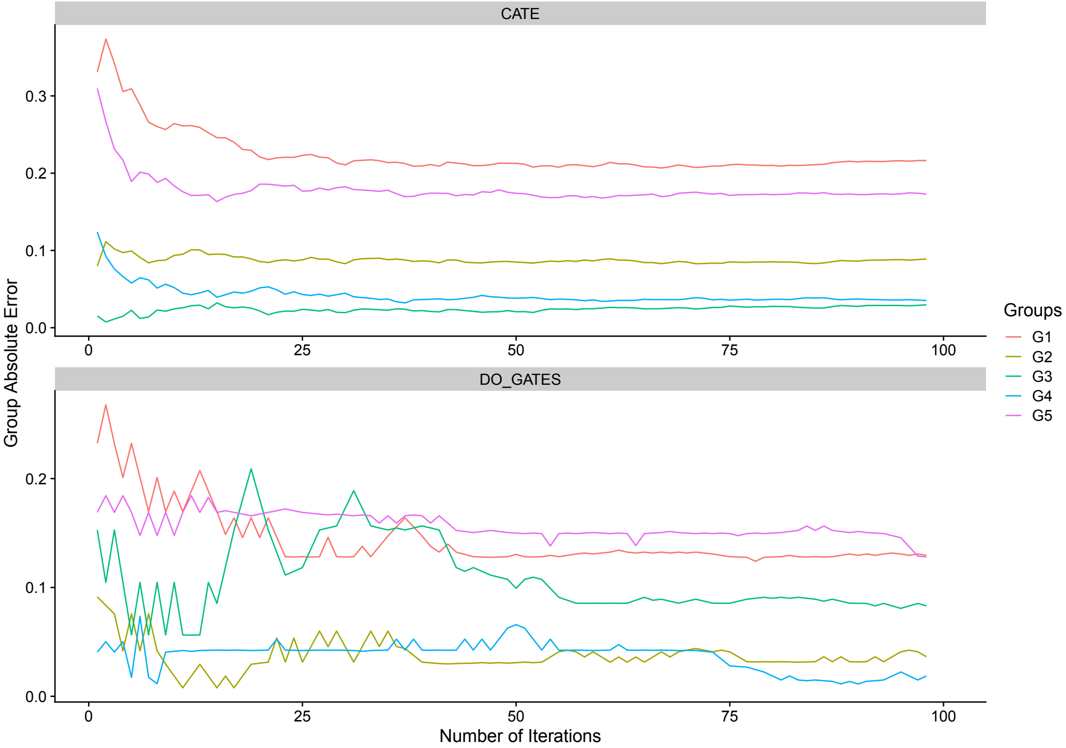


Figure 5.5: Upper plot shows the absolute error for the CATE estimator given B iterations. Lower plot shows the same for the DO GATES method.

5.4 Conclusion

In this paper, we propose a method to estimate group average treatment effects using machine learning methods and linear estimation for non-randomized control trials. Since flexibility in terms of the model choice, as well as interpretability of the results, is our main interest, we extend the idea of the GATES approach by using a doubly-robust estimator in the first, the non-parametric, step. In the second step, the linear projection function, we use Neyman-orthogonal moments to overcome the regularization bias due to the dependency of the covariates on the outcome and the treatment assignment parameter. This ensures to control for self-selection into treatment which is a realistic challenge in observational studies. Our main interest is to find heterogeneity in terms of quantiles from the treatment effect function and estimate them via a best linear projection. We also propose to weight the CATE function by taking the median over B iterations instead of single estimates. Since we are interested in describing the groups in terms of all observations, random sample splitting in half of the sample only gives us estimates from half of the observations. After approximately $B = 10$ iterations we have at least one estimate for each observation. Therefore, we suggest to use at least 50 or preferably 100 iterations in order to cover the whole sample.

We find that our proposed model has a lower MAE and a lower bias for the majority of the groups we consider and among different simulation scenarios, compared to the benchmark model. In randomized control trials, both methods perform competitive. As soon as we include selection-bias, the DO GATES method performs better due to its second orthogonalization step. We note that a natural extension is to estimate the ATE by taking the mean over the group estimates. However, this estimator is not the scope of the paper and there are direct approaches to estimate the ATE.

Bibliography

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *Annals of Statistics*, *47*(2), 1148–1178. https://doi.org/10.1214/18-AOS1709

Belloni, A., Chernozhukov, V., & Hansen, C. (2013). Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, *81*(2), 608–650. https://doi.org/10.1093/restud/rdt044

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, *21*(1), C1–C68. https://doi.org/10.1111/ectj.12097

Chernozhukov, V., Demirer, M., Duflo, E., & Fernández-Val, I. (2018). Generic machine learning inference on heterogeneous treatment effects in randomized experiments, with an application to immunization in india. https://doi.org/10.3386/w24678

Chernozhukov, V., Fernández-Val, I., & Luo, Y. (2018). The sorted effects method: Discovering heterogeneous effects beyond their averages. *Econometrica*, *86*(6), 1911–1938. https://doi.org/10.3982/ecta14415

Chernozhukov, V., & Semenova, V. (2018). Simultaneous inference for best linear predictor of the conditional average treatment effect and other structural functions. (CWP40/18). https://doi.org/10.1920/wp.cem.2018.4018

Fan, Q., Hsu, Y.-C., Lieli, R. P., & Zhang, Y. (2020). Estimation of conditional average treatment effects with high-dimensional data. *Journal of Business & Economic Statistics*, 1–15. https://doi.org/10.1080/07350015.2020.1811102

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer Series in Statistics. https://doi.org/10.1007/978-0-387-84858-7

Heckman, J., Ichimura, H., Smith, J., & Todd, P. (1998). *Characterizing selection bias using experimental data* (tech. rep. No. 5). JSTOR. https://doi.org/10.2307/2999630

Hitsch, G. J., & Misra, S. (2018). Heterogeneous treatment effects and optimal targeting policy evaluation. *Available at SSRN 3111957*. https://doi.org/10.2139/ssrn.3111957

Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, *47*(260), 663–685. https://doi.org/10.1080/01621459.1952.10483446

Knaus, M. C. (2020). Double machine learning based program evaluation under unconfoundedness. *arXiv preprint arXiv:2003.03191*.

Knaus, M. C., Lechner, M., & Strittmatter, A. (2020). Machine Learning Estimation of Heterogeneous Causal Effects: Empirical Monte Carlo Evidence. *The Econometrics Journal*. https://doi.org/10.1093/ectj/utaa014

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, *116*(10), 4156–4165. https://doi.org/10.1073/pnas.1804597116

Lee, S., Okui, R., & Whang, Y.-J. (2017). Doubly robust uniform confidence band for the conditional average treatment effect function. *Journal of Applied Econometrics*, *32*(7), 1207–1225. https://doi.org/10.1002/jae.2574

Lunceford, J. K., & Davidian, M. (2004). Stratification and weighting via the propensity score in estimation of causal treatment effects: A comparative study. *Statistics in Medicine*, *23*(19), 2937–2960. https://doi.org/10.1002/sim.1903

Nie, X., & Wager, S. (2020). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*. https://doi.org/10.1093/biomet/asaa076

Park, C., & Kang, H. (2019). A groupwise approach for inferring heterogeneous treatment effects in causal inference. *arXiv preprint arXiv:1908.04427*.

Powers, S., Qian, J., Jung, K., Schuler, A., Shah, N. H., Hastie, T., & Tibshirani, R. (2018). Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in Medicine*, *37*(11), 1767–1787. https://doi.org/10.1002/sim.7623

Robins, J. M., & Rotnitzky, A. (1995). Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association*, *90*(429), 122–129. https://doi.org/10.1080/01621459.1995.10476494

Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1994). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, *89*(427), 846–866. https://doi.org/10.1080/01621459.1994.10476818

Robinson, P. M. (1988). Root-n-consistent semiparametric regression. *Econometrica*, 931–954. https://doi.org/10.2307/1912705

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*(1), 41–55. https://doi.org/10.1093/biomet/70.1.41

Williamson, E. J., Forbes, A., & White, I. R. (2014). Variance reduction in randomised trials by inverse probability weighting using the propensity score. *Statistics in Medicine*, *33*(5), 721–737. https://doi.org/10.1002/sim.5991

Zimmert, M., & Lechner, M. (2019). Nonparametric estimation of causal heterogeneity under high-dimensional confounding. *arXiv preprint arXiv:1908.08779*.

5.A Figures

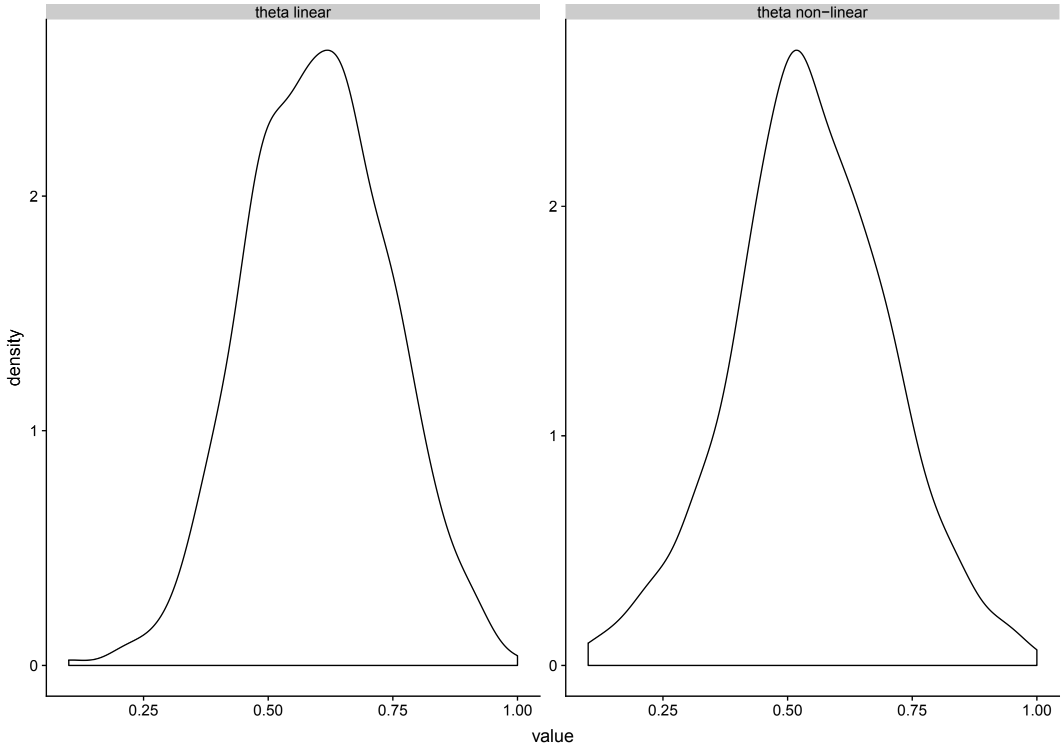


Figure 5.6: Density for the true and fixed heterogeneous treatment effect function. Left plot shows the linear treatment effect dependency and right plot the non-linear dependency.

5.B Tables

Table 5.3: Mean absolute error by setting and groups for N = 2000.

SETTING	G	DO GATES	CATE	SETTING	G	DO GATES	CATE
A	1	0.18	**0.07**	G	1	0.18	**0.07**
A	2	0.07	**0.03**	G	2	0.08	**0.02**
A	3	0.04	**0.02**	G	3	0.05	**0.02**
A	4	0.07	**0.03**	G	4	0.07	**0.04**
A	5	**0.12**	0.13	G	5	0.14	**0.12**
B	1	**0.12**	0.27	H	1	**0.15**	0.25
B	2	0.07	0.07	H	2	0.08	**0.07**
B	3	0.06	**0.03**	H	3	0.06	**0.04**
B	4	**0.07**	0.11	H	4	**0.08**	0.09
B	5	**0.12**	0.29	H	5	**0.14**	0.27
C	1	**0.11**	0.21	I	1	**0.12**	0.23
C	2	**0.25**	0.49	I	2	**0.23**	0.48
C	3	**0.25**	0.61	I	3	**0.23**	0.59
C	4	**0.30**	0.74	I	4	**0.29**	0.71
C	5	**0.73**	1.10	I	5	**0.73**	1.05
D	1	**0.16**	0.34	J	1	**0.16**	0.37
D	2	**0.23**	0.53	J	2	**0.25**	0.55
D	3	**0.27**	0.64	J	3	**0.30**	0.66
D	4	**0.35**	0.75	J	4	**0.35**	0.75
D	5	**0.78**	1.03	J	5	**0.77**	1.03
E	1	**0.09**	0.18	K	1	**0.12**	0.23
E	2	**0.25**	0.48	K	2	**0.24**	0.48
E	3	**0.26**	0.60	K	3	**0.23**	0.59
E	4	**0.30**	0.72	K	4	**0.27**	0.70
E	5	**0.73**	1.08	K	5	**0.70**	1.05
F	1	**0.10**	0.32	L	1	**0.11**	0.36
F	2	**0.54**	0.70	L	2	**0.56**	0.72
F	3	**0.78**	0.92	L	3	**0.78**	0.93
F	4	**1.00**	1.14	L	4	**0.99**	1.13
F	5	**1.36**	1.50	L	5	**1.36**	1.49

Notes: Settings as in Table 5.1. Number of observations: N = 2000. Averages over 100 repetitions.

Table 5.4: Mean absolute error by setting and groups for N = 500.

SETTING	G	DO GATES	CATE	SETTING	G	DO GATES	CATE
A	1	0.34	**0.08**	G	1	0.27	**0.14**
A	2	0.16	**0.05**	G	2	0.16	**0.06**
A	3	0.12	**0.09**	G	3	0.12	**0.10**
A	4	**0.11**	0.14	G	4	**0.09**	0.15
A	5	**0.21**	0.23	G	5	**0.19**	0.23
B	1	0.29	0.29	H	1	**0.25**	0.39
B	2	**0.18**	0.10	H	2	0.15	**0.13**
B	3	0.13	**0.07**	H	3	0.14	**0.07**
B	4	0.17	**0.10**	H	4	0.16	**0.13**
B	5	0.30	**0.25**	H	5	**0.28**	0.29
C	1	**0.31**	0.60	I	1	**0.27**	0.52
C	2	**0.31**	0.83	I	2	**0.31**	0.80
C	3	**0.37**	1.04	I	3	**0.37**	1.01
C	4	**0.60**	1.31	I	4	**0.59**	1.29
C	5	**1.64**	1.88	I	5	**1.63**	1.82
D	1	**0.55**	0.85	J	1	**0.47**	0.79
D	2	**0.44**	1.00	J	2	**0.43**	0.98
D	3	**0.53**	1.14	J	3	**0.50**	1.13
D	4	**0.73**	1.35	J	4	**0.73**	1.34
D	5	**1.30**	1.78	J	5	**1.35**	1.74
E	1	**0.31**	0.58	K	1	**0.29**	0.55
E	2	**0.31**	0.81	K	2	**0.29**	0.82
E	3	**0.33**	1.02	K	3	**0.37**	1.03
E	4	**0.53**	1.29	K	4	**0.62**	1.30
E	5	**1.62**	1.86	K	5	**1.64**	1.82
F	1	**0.38**	0.79	L	1	**0.23**	0.69
F	2	**0.65**	1.09	L	2	**0.62**	1.06
F	3	**0.97**	1.35	L	3	**0.99**	1.33
F	4	**1.42**	1.68	L	4	**1.43**	1.68
F	5	**2.21**	2.30	L	5	**2.25**	2.31

Notes: Settings as in Table 5.2. Number of observations: N = 500. Averages over 100 repetitions.

5.C Supplementary

5.C.1 Correlated Covariates

We show how we generate the covariates and their correlation. We assume correlation through a uniform distribution of the covariance matrix, which is then transformed to a correlation matrix. Correlated chararcteristics are more common in real datasets and helps to investigate the performance of ML algorithms, especially the regularization bias, in a more realistic manner. Figure 5.7 shows the correlation matrix for 10 randomly selected covariates to give an example of the correlation.

Algorithm 13: Generation of Covariates

```
1 Generate random positive definite covariance matrix $\Sigma$ based on a uniform
    distribution over the space $k \times k$ of the correlation matrix.
2 Scale covariance matrix This equals the correlation matrix and can be seen as the
    covariance matrix of the standardized random variables $\Sigma = \frac{X}{\sigma(X)}$.
3 Generate random normal distributed variables $X_{N \times k}$ with mean = 0 and variance =
    $\Sigma$.
```

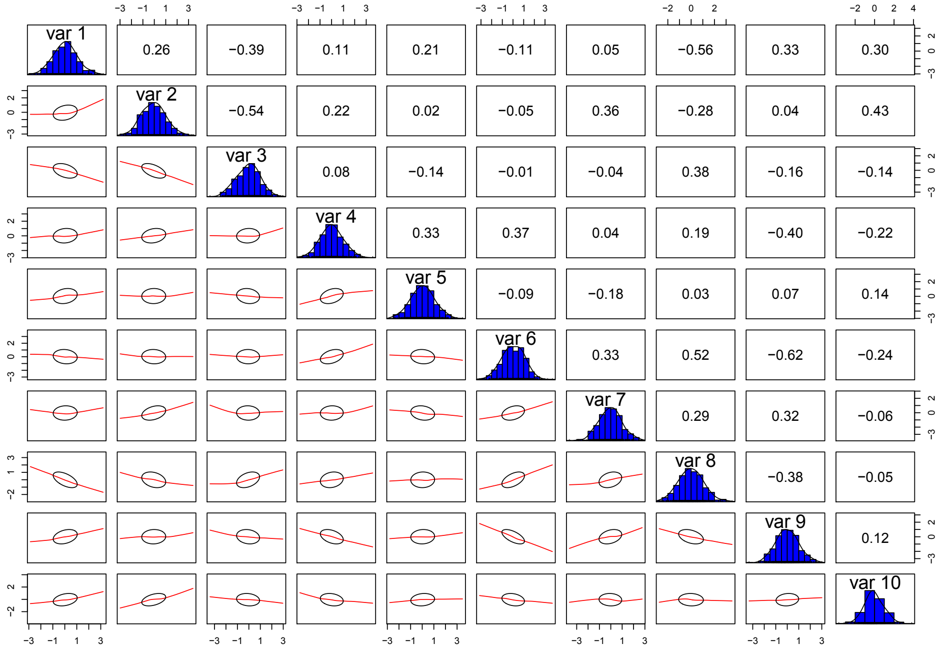


Figure 5.7: Correlation Matrix of Covariates. Correlation metric is Bravais-Pearson.

5.C.2 Additional Simulations

Based on the results from Knaus et al. (2020) we considered the doubly-robust estimator as a benchmark model for our simulation. Using the doubly-robust estimator enables us to use different machine learning algorithms which is an advantage in terms of flexibility. Here we also show simulation results where we compare our DO GATES method with the generalized random forest (GRF) (Athey et al., 2019). We do keep the doubly-robust estimator to build the CATE function and estimate the group effects as mentioned above for the DO GATES estimator. To compare the results with the GRF we split the resulting CATE function from the GRF in five groups and take the mean over the observations within the groups. Since we use B iterations to create the final DO GATES groups we do the same with the CATE function from the GRF as we did with the doubly-robust estimator. This means taking the median for each observation over the B iterations resulting from the GRF.

For computational reasons we again limit the simulation in the sense that we use a random forest algorithm to estimate the functions for the CATE from the doubly-robust estimator as well as for the nuisance functions we use for orthogonalization. Different from the simulation in the main text we use different tuning parameter for all functions and select the final model via cross-validation. We do this since the GRF algorithm also provides the option to tune parameters in order to select the final model. The DGP is the same as described in section 5.3.1 except that we do not standardize the treatment effect functions. This might make the estimation of the CATE a little easier since we neglect one transformation.

We show the results for N = 500 observations and limit the repetition from the Monte Carlo simulation to 10 due to computational reasons.

Table 5.5: Settings and Monte Carlo averages for N = 2000.

Scenarios	A/G	B/H	C/I	D/J	E/K	F/L
N	500	500	500	500	500	500
$\mathbb{R}^p$	20	20	20	20	20	20
$\mathrm{P}(D = 1)$	0.5	0.2 (imbalanced)	linear	interaction	non-linear	linear
Misspecification	No	No	No	No	No	Yes
MAE GRF	0.71/0.67	0.88/0.81	0.75/0.84	0.93/0.81	0.88/0.66	**1.03/1.06**
MAE DO GATES	**0.32/0.37**	**0.50/0.56**	**0.58/0.70**	**0.70/0.76**	**0.58/0.58**	1.20/1.10
$Bias^2$ GRF	0.73/0.68	1.10/0.94	0.84/1.04	1.27/0.94	1.12/0.65	1.49/1.56
$Bias^2$ DO GATES	**0.10/0.15**	**0.26/0.34**	**0.29/0.50**	**0.68/0.59**	**0.41/0.34**	**1.47/1.20**

Notes: Mean absolute error (MAE) of the average treatment effect (estimated from the quantile estimates) over S = 10 Monte Carlo simulations. Setting G to L consists of a non-linear treatment effect function.

Table 5.5 shows a smaller MAE and bias for the DO GATES method compared to the GRF. Only for the setting with misspecification we find a slightly smaller MAE for the GRF while the bias is always smaller for our proposed method. When looking at the specific groups we find that especially group 1 and 5 are the main driver for the smaller MAE while group 3 and 4 behave competitively to the GRF. We plot the MAE for the single groups and each DGP repetition in Figure 5.8. If we introduce misspecification group 1 still shows a lower MAE while group 5 now shows a slightly bigger MAE compared to the GRF. We also notice that the estimates from the GRF might have a smaller variance among the 10 Monte Carlo repetitions in some settings. However, there is no real structure among the DGP settings or the groups. We leave this investigation for further research.

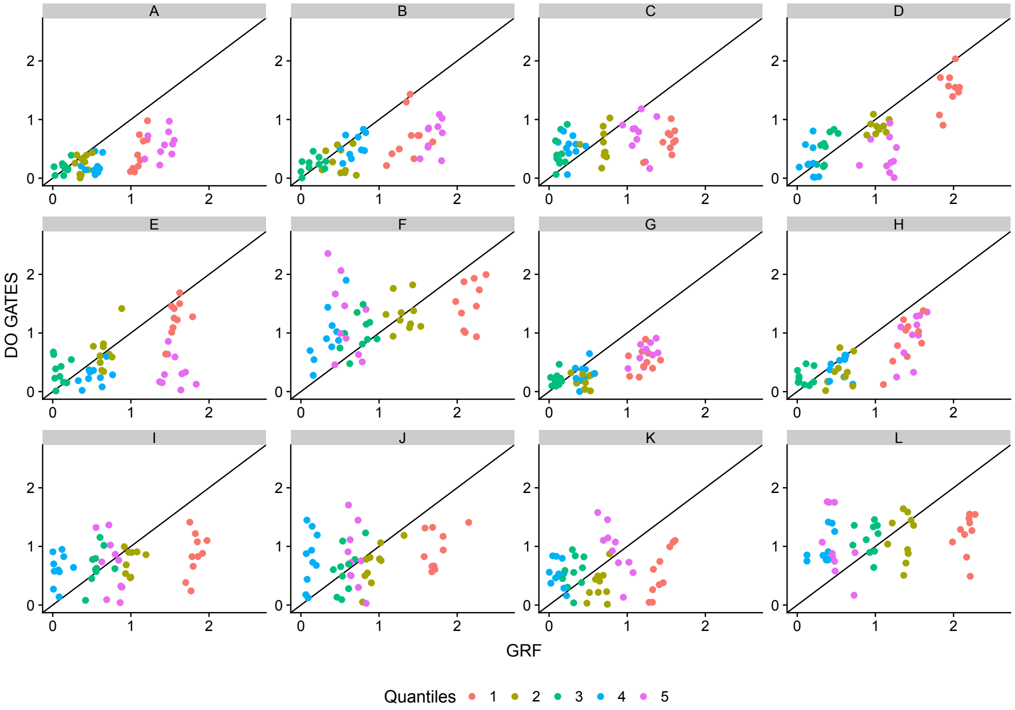


Figure 5.8: Group estimate from the GRF function and DO GATES. Groups are coloured by quantiles (1 = least affected, 5 = most affected). Axes show the absolute error between estimates and true quantile treatment effects. The 45-degree line indicates the equality of both methods. Th number of observations equals 500.

Chapter 6

Supervised Randomization in Controlled Experiments

Publication

Haupt, J., Jacob, D., Gubela, R., & Lessmann, S. (2019). Affordable Uplift: Supervised Randomization in Controlled Experiments. *Proceedings of the 40th International Conference on Information Systems (ICIS).*

Abstract

Customer scoring models are the core of scalable direct marketing. Uplift models provide an estimate of the incremental benefit from a treatment that is used for operational decision-making. Training and monitoring of uplift models require experimental data. However, the collection of data under randomized treatment assignment is costly, since random targeting deviates from an established targeting policy. To increase the cost-efficiency of experimentation and facilitate frequent data collection and model training, we introduce *supervised randomization*. It is a novel approach that integrates existing scoring models into randomized trials to target relevant customers, while ensuring consistent estimates of treatment effects through correction for active sample selection. An empirical Monte Carlo study shows that data collection under supervised randomization is cost-efficient, while downstream uplift models perform competitively.

JEL Classification: C14, C21, C52, M31

Replication code is available on Quantlet.

6.1 Introduction

Direct marketing plays a key role in consumer markets. The continuous growth of e-commerce, accounting for 1.8 trillion Euros globally in 2019 (Statista, 2019), is accompanied by simultaneous growth in online and email advertising. Spending on traditional print advertising like catalog marketing has shown a similar growth (Statista, 2017). At the core of scalable direct marketing, campaign analysts employ models to predict future customer behavior and target responsive clients (Olson et al., 2012).

For example, a decision tree could be trained to predict the probability for a customer to purchase in the next week based on known characteristics. The expected behavior of the customer could then be used to inform operational decision-making in that customers with a probability below average are targeted with an incentive. However, the predictive model is agnostic to the marketing policy, the overall effectiveness of the marketing action and the effect of the marketing action on individual customers. Outcome models provide an estimate of customer behavior, but fail to provide an estimate of the potential change in customer behavior, which is the goal of marketing intervention.

A growing research stream advocates that the decision which customers to target should be addressed directly through causal inference in the form of uplift models (Devriendt et al., 2018; Gubela et al., 2017). Instead of predicting customer behavior, uplift models estimate the causal effect of a marketing action on an individual customer given their characteristics. In the above example, an uplift tree could be trained to predict the increase in probability for a customer to conduct a purchase in the next week if a catalog was sent. Uplift models thus provide an estimate of the incremental benefit from the marketing treatment, which can explicitly be used as a direct criterion for operational decisions by comparing it to the incremental cost. Conceptually, uplift models align with the actual decision problem of choosing the action with the highest incremental gain for each customer.

Uplift models are trained on experimental data and estimate the treatment effect by comparing the observed behavior of a group of individuals who have received the treatment, the treatment group, and a distinct group of individuals who have not received the treatment, the control group. Similarly, experimental data is required to evaluate the performance of uplift models (Radcliffe, 2007). In contrast, non-causal models of customer behavior are trained and evaluated on customers of which all or none have received the treatment. Collecting experimental data in randomized experiments is well established in practice in the form of A/B tests. Although used to evaluate the gross benefit of campaigns, A/B tests are not commonly used for uplift modeling to estimate individualized treatment effects (Ascarza, 2018).

During experiments, random assignment of individuals to either the treatment or control group is crucial to train unbiased uplift models. However, data collection through randomized experiments is costly, since random targeting withholds marketing spending on some customers that would be targeted under the established targeting policy and applies spending on customers that would otherwise not be targeted. The deployment of uplift models exacerbates data collection costs since decision support systems typically require continuous or frequent evaluation and occasional retraining on recent observations, which in turn require fresh experimental data.

We propose a novel approach for the collection of experimental data for uplift modeling based on the combination of cost-optimized randomization at the time of data collection and selection bias correction during model building, which we refer to as *supervised randomization*. In a nutshell, supervised randomization introduces a stochastic component to the existing targeting model and extends the standard experimental design with full randomization by considering customers that are rejected by the targeting policy.

Our contribution is two-fold. First, we show that supervised randomization can be used to integrate existing scoring models into randomized trials. The integration of existing scoring models into group assignment increases the cost-efficiency of experimentation and facilitates continuous data collection during regular business operation. Continuous data collection is critical for non-disruptive experimentation, monitoring the performance of uplift models under deployment and recurring model training. Facilitating model training and monitoring has the additional benefit to improve the acceptance of causal models by management and stakeholders.

Second, we introduce inverse probability weighting and doubly robust estimation as methods to control for biased treatment assignment to the uplift literature. Uplift models have so far relied on the assumption of data collected under full or imbalanced randomization in randomized controlled trials. We show that recent advances in the econometrics literature extend the applicability of uplift models to cases with non-standard treatment assignment. The bias-corrected uplift models are shown to perform competitively on simulated data.

6.2 Background

Consider a marketing action applied to an individual user i as a treatment intended to change an observed outcome Y_i. Let $D_i \in 0, 1$ be an indicator if the individual has been treated and denote the outcome with and without treatment as Y_i^1 and Y_i^0, respectively. Then the individual treatment effect is the incremental gain caused by a marketing action $Y_i^1 - Y_i^0$. Because a customer either does or does not receive the marketing action, the actual treatment effect is not observable. We can, however, estimate the treatment effect on the population or on the individual level. We denote the average campaign uplift as average treatment effect (ATE) and the customer-level uplift $\tau = \mathsf{E}[Y_i^1 - Y_i^0 | x = X_i]$ as individualized treatment effect (ITE) (Knaus et al., 2020; Powers et al., 2018), sometimes alternatively denoted conditional average treatment effect (CATE) in the econometrics literature. Furthermore, we refer to a model used to estimate the outcome Y_i as outcome model and a model used to estimate the treatment effect τ as a causal model, as a more general alternative to the term uplift model. The operational decision problem posed in uplift modeling is to decide if an individual customer should receive the marketing treatment. The decision is automated through the targeting policy, a mapping from the estimated ITE to the binary decision of whether to treat the individual.

Three assumptions are needed for causal inference following the potential outcome theorem (Rosenbaum & Rubin, 1983). First, the *Stable Unit Treatment Value Assumption (SUTVA)* guarantees that the potential outcome of a customer is unaffected by changes in the treatment assignment of other customers. This assumption may be violated when treatment effects propagate through the social network of customers (Ascarza et al., 2017). In settings of low value or low involvement products, research on treatment effects in marketing typically assumes that no interaction takes place (Hitsch & Misra, 2018) or that the indirect effect of treatment on other customers is at least substantially smaller than the direct effect of the treatment (Imbens & Wooldridge, 2009).

The second assumption is *conditional unconfoundedness*, i.e. the independence between the potential outcomes and the treatment assignment given the observed covariates (X) (Rosenbaum & Rubin, 1983).

$$\left(Y_i^1, Y_i^0\right) \perp\!\!\!\perp D_i | X_i = x. \tag{6.1}$$

The third assumption, called *overlap*, guarantees that for all $x \in supp(X_i)$ the probability to receive treatment $e(x) = \mathsf{P}(D = 1 | X_i = x)$ is bounded away from 0 and 1:

$$0 < e(x) < 1. \tag{6.2}$$

When the treatment assignment process is under the control of the experiments as in the customer targeting setting, conditional independence and overlap can be ensured by design through fully randomizing treatment assignment with treatment probability $e(x) = e \in (0; 1)$. Randomized experiments assign individuals at random to one of at least two conditions, where each condition entails a specific treatment. In controlled experiments, one condition is the control condition in which individuals receive no treatment. In combination, randomized controlled trials (RCT) are the gold standard of data collection for causal inference. We refer to uniform treatment assignment as *full randomization*. Supervised randomization provides a framework that preserves the advantage of the randomized experimental design but allows some control over the probability of treatment assignment on the individual level.

6.3 Literature Review

The unbiased training of causal models and targeting policies requires data that fulfills the assumptions of the potential outcome theorem. In addition, the unbiased evaluation of causal models and policies also requires experimental data and metrics developed for counterfactual prediction (Hitsch & Misra, 2018; Radcliffe, 2007). Violation of the unconfoundedness (Eq. 6.1) and overlap assumptions (Eq. 6.2) in observational studies cannot be substituted by collecting more data in the form of more covariates or more observations (Gordon et al., 2019). Randomized experiments are thus considered a prerequisite to uplift modeling. However, the design and costs of RCT are often not discussed in the literature. We aim to fill this gap by proposing a more efficient design for randomized experiments. We first summarize recent developments in causal machine learning, related research on efficient experimental design and methods to correct for treatment assignment in observational studies.

Causal machine learning methods can be divided into direct and indirect approaches. Direct estimation algorithms construct a feasible loss to estimate a model for the ITE. Indirect approaches model the expected customer response conditional on the treatment group and estimate the ITE as the difference between expected responses. This study employs a robust, indirect two-model logistic regression and a state-of-the-art, direct causal forest for the empirical comparison and provides a discussion of these models below. For an in-depth discussion and benchmark of recent methods for ITE estimation see (Knaus et al., 2020; Künzel et al., 2019; Powers et al., 2018).

Indirect approaches estimate the treatment effect via estimating the response with and without treatment using common statistical learners. The two-model approach (Radcliffe, 2007), or k-model approach in settings with more than one treatment, estimates a separate model for the outcome in the treatment group and control group data and estimates the ITE as the difference between the predicted outcomes. The two-model approach is flexible with regard to the underlying outcome models. K-nearest neighbors learners (Gubela et al., 2019) and deep neural networks (Farrell et al., 2021) have demonstrated promising model performance in the two-model framework. While recent research advocates discretizing the outcome variable to use classification models in continuous settings (Gubela et al., 2017), the two-model approach extends naturally to both categorical and continuous outcomes. This facilitates the use of classification and regression models to forecast, for example, purchase completion or customer spending, respectively.

A number of well-known machine learning algorithms have been extended to estimate the ITE directly without the need to model the customer response (Lo, 2002; Zaniewicz & Jaroszewicz, 2013). Note that the average treatment effect within a subgroup provides a useful estimate of the treatment effect for individuals within that subgroup. Hence, algorithms that split the data into groups to calculate estimates on the subset are inherently applicable to causal modeling and modifications of the k-nearest neighbor estimator (Hitsch & Misra, 2018) and tree-based models (Athey & Imbens, 2016; Rzepakowski & Jaroszewicz, 2012) have been applied to estimate individualized treatment effects. Causal tree models modify the Classification and Regression Tree by a splitting criterion maximizing the expected variance in treatment effects between leaves (Athey & Imbens, 2016). Within each terminal node conditional on the covariate splitting, the conditional average treatment effect can be estimated and provides an ITE for the observations falling into that node.
Causal trees can be combined into ensembles through bagging or boosting. (Powers et al., 2018) propose a gradient-boosted ensemble of causal trees and an algorithm using multivariate adaptive regression splines. Causal forests are similarly flexible models and have been shown to be consistent and asymptotically normal for a fixed covariate space (Athey et al., 2019).

Both direct and indirect approaches to ITE estimation share the need for experimental data. The collection of experimental data has not been explicitly explored in the uplift literature. However, concerns over the organizational difficulty and the opportunity cost of running randomized controlled trials have led to research on the optimal use of available data and efficient experimental design in related fields.
A popular strategy for the evaluation of multiple targeting policies is to avoid experimentation for each candidate policy and instead to estimate each policy's performance using one existing, fully randomized experiment. The cost-efficient evaluation is possible through extrapolation from observations where the policy recommendation matches the observed random treatment assignment, weighted to match the actual population (Athey & Wager, 2017; Hitsch & Misra, 2018; Swaminathan & Joachims, 2015). This evaluation strategy requires existing experimental data, while our goal is to decrease the cost of collecting experimental data through efficient randomization. Approaches to efficient evaluation and efficient randomization are therefore complementary.

The experimental design of previous studies indicates awareness of data collection costs. Table 6.1 shows the marketing goal, data accessibility, number of observations and the imbalance between treatment and control group sizes of experimental campaigns in customer targeting applications. The large number of observations in recent studies is unsurprising since common technologies in e-commerce settings (e.g., web cookies) facilitate the collection of large data volumes of customer interactions in online shops Diemert et al. (2018). However, large-scale experimentation imposes substantial costs by randomly withholding profitable treatment for a sizeable control group. We reason that large experiment sizes indicate that companies perceive potential gains from causal modeling and are willing to collect data on a sufficient scale. Since the costs of experimentation are a result of the randomization of treatment assignment, we propose that supervised randomization can lead to cost reductions that are economically relevant in practice given the scale of experimentation. The savings potential increases with the targeting cost and will thus be most effective for catalog or telephone marketing, where resource-intensive treatments drive cost, and in customer churn management, where targeting customers may increase awareness of contract expiration and induce churn in otherwise passive customers.

We further observe that 10 of 11 datasets show a substantial difference in size between the treatment and control group, which we denote *imbalanced full randomization.* The imbalance implies that companies assign customers to the treatment group with probabilities 2–17 times higher than assignment to the control group. The observed imbalanced experimental design is more efficient than equal assignment to treatment and control group when the marketing action is expected to be profitable on average and treatment is the dominant targeting strategy. Companies are thus conducting active cost management of random experiments based on an assessment of overall treatment effectiveness. Our approach follows the same motivation, but extends cost management to the individual level based on an assessment of the individual treatment effectiveness.

Table 6.1: Randomized treatment data in marketing

Application	Source	Obs. (in 1000)	T:R Ratio*
Direct mail in office supplies	Kane et al. (2014)	460	17:1
Mail promotion	B. J. Hansotia and Rukstales (2002)	550	10:1
Cross-selling mail in insurance	Guelman (2014)	34,370	9:1
MSN subscription	Chickering and Heckerman (2000)	110	9:1
Criteo online advertising campaign	Diemert et al. (2018)	29,106	7:1
Direct mail in financial services	Kane et al. (2014)	1,144	5:1
Simulation study	Lo (2002)	100,000	4:1
Customer retention mail in insurance	Guelman et al. (2015)	12	2:1
Catalog marketing	Hitsch and Misra (2018)	441	2:1
E-mail promotion in merchandising	Hillstrom (2008)	64	2:1
E-mail promotion in holiday marketing	B. Hansotia and Rukstales (2002)	282	1:1

*Treatment:Control Ratio

The design of randomization on the individual level is more thoroughly discussed in the medical literature (Schulz & Grimes, 2002). On the one hand, administering a pharmaceutical to a random patient can induce severe health issues, so randomized trials pose a risk for patient health. On the other hand, new treatment may prove to be a substantial improvement over comparative options, so that withholding treatment can be seen as suboptimal care. The latter concern for optimal treatment of patients has motivated research on adaptive randomization procedures, where patients are more likely to be assigned to treatment for which positive outcomes have been previously observed over the course of the study (Lachin et al., 1988; Rosenberger & Lachin, 1993). Response-adaptive randomization in medical trials differs from our approach in that we use a scoring model to adjust treatment probability conditional on customer characteristics rather than observing treatment outcomes during the trial.

The trade-off between collecting more data to improve the scoring model and applying an existing treatment policy deterministically corresponds to the exploration-exploitation problem in reinforcement learning and multi-armed bandit approaches. Supervised randomization is related to the ε-greedy algorithm (Schwartz et al., 2017), extended by heterogeneous exploration probabilities $\varepsilon_i = 1 - e(x)$. In comparison to upper confidence bound sampling or Thompson sampling (Schwartz et al., 2017) which favor exploration of uncertain predictions, supervised randomization favors exploration close to the decision boundary of the policy and facilitates straightforward logging of the true treatment probability.

This study considers supervised randomization for continuous evaluation and periodical updating of treatment effect models. We do not adapt the scoring model and conditional treatment probabilities during the duration of the experiment as opposed to online learning of the treatment effect model under reinforcement learning. We leave a more in-depth comparison for future research.

Supervised randomization introduces dependency between the covariates and treatment assignment in the data as a side effect of adjusting treatment probabilities on the individual level. This violates the conditional independence assumption and without correction would lead to biased treatment estimates known as selection bias. An intuitive interpretation is that selection bias is due to the covariates being non-identically distributed between the treatment and control group because group assignment is itself based on the observed covariates. Statistical analysis of the average or individualized treatment effect on data that violates the unconfoundedness assumption thus requires correction for the effect of the covariates on the individual probability to receive treatment. To the best of our knowledge, methods to systematically correct for selection bias have not yet been studied in the uplift community. Instead, research on uplift modeling assumes the feasibility of randomized control trials where there is no such selection bias by design, but ignores the associated costs of data collection in practice.

However, selection bias corrections are well-understood and common for research using observational data, where random treatment assignment is not ethical or feasible. The most common technique for selection bias correction are the inverse probability weighting estimator (Horvitz & Thompson, 1952) (IPW) and its extension to the doubly robust (DR) estimator (Robins et al., 1994). Selection bias correction has recently been integrated into popular ITE estimators, which are applied in the observational studies prevalent in economics research (Athey et al., 2019; Künzel et al., 2019).

Within the field of information systems, the IPW correction is used in observational studies (Marco Caliendo et al., 2012) and research on recommender systems and reinforcement learning. In recommendation settings, explicit and some forms of implicit feedback can be understood as the outcome of a non-randomized experiment, where users evaluate a subset of items that they select based on their preferences. The customer feedback can be corrected by weighting the feedback by the probability of a customer to interact with an item before rating it (Schnabel et al., 2016).
Reinforcement learning research makes regular use of existing log data, which is cost-efficient to collect and available at scale, but not fully randomized. Under some conditions, unbiased training or evaluation of a learning algorithm is possible using IPW to correct for the treatment policy at the time of data collection (Swaminathan & Joachims, 2015). Interestingly, if treatment under the existing policy is stochastic with a known probability and the treatment probability is logged, the resulting process can be seen as an online version of the supervised randomization process.

For observational data, the treatment probability used to correct for the selection bias is unknown and IPW thus follows a two-step process. In the first step, the treatment probabilities used for IPW correction are estimated from the observed data. In the second step, the treatment probability estimates are used to correct subsequent estimates of treatment effects. The approach proposed here is substantially different from the common applications of IPW in the first step. Under supervised randomization, the true treatment probability for each observation is actively controlled and thus known, eliminating the need for estimation and the potential for estimation error through unobserved variable bias or model misspecification.

6.4 Efficiently Randomized Experimental Design

Within this study, we take a holistic view of the causal modeling process emphasizing the interaction between data collection and model building. Data collection through a randomized controlled trial is a necessary part of the causal modeling process. To collect RCT data, the established targeting policy is temporarily replaced by randomized treatment assignment. However, the replacement of an efficient targeting strategy by a random assignment has negative side effects in practice, even when restricted to a subset of customers. First, randomized treatment assignment carries opportunity costs resulting from targeting the wrong customers. Compared to an existing effective targeting strategy, profitable customers are less likely to be targeted while less profitable customers are more likely to be targeted. Second, customers may misconceive data collection periods as a decrease in service or advertising quality. Since customers are not informed about the temporary replacement of the targeting model, they will attribute the random treatment assignment to the targeting efforts of the company.

Instead of replacing the established targeting policy with randomized treatment allocation, we propose to introduce a stochastic component to the existing targeting model as *supervised randomization*. Under supervised randomization, treatment assignment is largely driven by the effective targeting model but sufficiently randomized to allow the estimation and evaluation of causal models. Embedding the existing targeting model into the experimental design has three merits.

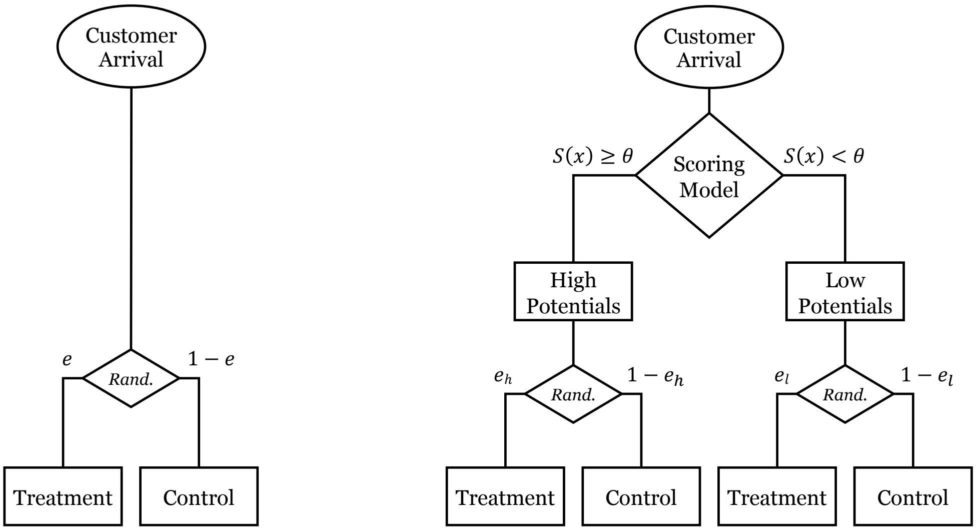
First, supervised randomization increases the return on treatment during experimentation when compared to full randomization. Supervised randomization allows us to actively decrease the cost of running randomization experiments by treating profitable customers identified by the targeting model with a higher probability than customers identified as less profitable by the scoring model.

Second, supervised randomization with conservative propensity mapping could facilitate continuous experimentation. Since both training and evaluation of causal models require experimental data, regular repetition of experimental data collection is necessary when causal models are deployed. Continuous experimentation could reduce variability in service from the perspective of the customer and streamline data collection for the company by avoiding interference with operation to run experiments and reducing the need to justify and approve the need of data collection at intervals. These goals are shared with reinforcement or bandit learning with the important difference that our approach facilitates model estimation through standard machine learning or uplift methodology.

6.4.1 Supervised Randomization

Supervised randomization introduces heterogeneity in treatment probabilities into a randomized controlled experiment. We discuss the proposed approach as an extension to A/B testing through the introduction of a targeting model in several steps. We describe the process for one treatment and one control group in the online context where customers arrive in sequence, but note that the same process extends to more than one treatment group and other static settings.

A/B testing for treatment evaluation is an instance of randomized controlled experiments with a single treatment. Each arriving customer is randomly assigned to the treatment or control group. The probability to receive treatment is identical for all customers $e(x) = e$ with the probability for assignment to the control group $1 - e$. The probability of treatment assignment can be equal $e = 1 - e = 0.5$ or imbalanced towards the preferred strategy for $e \in (0; 1)$. As discussed in the literature review, imbalanced probabilities are used to control



(a) Full randomization (A/B test) showing targeting policy

(b) Supervised randomization (two groups)

Figure 6.1: Experimental design of full randomization (left) and supervised randomization (right). Note the heterogeneity in treatment probability for supervised randomization. *Rand.* indicates random assignment.

the costs of the experiment in practice. For the case of multiple treatments, a different probability can be assigned to each treatment.

During regular business operation, the existing scoring model assigns a score $S(x)$ to each customer, where $S(x)$ could be an estimate of the conversion probability or the ITE. The model score $S(x)$ is compared to a threshold θ to classify customers into groups, where the group *high potentials* consists of the customers with the higher score, e.g., the highest probability to respond positively to the treatment. The *high potential* group would be targeted during regular operation, while the *low potential* group would receive no treatment. Figure 6.1a visualizes a scoring model during A/B testing. For the purpose of experimental data collection, the classification and deterministic targeting is replaced by random targeting. Independent of group assignment, each customer has an equal probability to receive treatment $e(x) = e$.

The proposed process of supervised randomization (Algorithm 14) integrates the scoring model into the randomized treatment assignment. As an intermediate step, let the treatment probability be dependent on the classification by the targeting strategy as depicted in Figure 6.1b. Different to the A/B test described in Figure 6.1a, where the targeting policy does not affect the treatment assignment, we now treat high potential customers with probability e_h and low potential customers with probability e_l, where $e_h \neq e_l$ and $e_l, e_h \in (0; 1)$. Note that e_h and e_l do not need to sum up to 1. We increase the treatment probability in the high potential group relative to the low potential group by choosing e_h and e_l so that $e_h > e_l$. Thereby, more high potential customers than low potential customers are treated, in accordance with the scoring model and approximating the regular targeting policy. Simultaneously, we preserve a degree of randomization in the treatment/control assignment, since each customer has some probability to be assigned to the treatment or control group, respectively. The randomization is required to fulfill the overlap assumption (Eq. 6.2) and should be large enough in practice to ensure coverage over the range of customer characteristics in both the treatment and control group. By violating the overlap assumption and setting $e_h = 1$ and $e_l = 0$, we recover the deterministic targeting policy of the classification model, where only customers in the *high potential* group are treated. Note that if we choose a constant treatment probability $e_h = e_l$, the process simplifies to an A/B test on the whole population as shown in Figure 6.1a.

We can further approximate individualized targeting by introducing more groups, each with a unique treatment probability e_k. Define a set of thresholds $[\theta_1, \theta_2, \ldots, \theta_K]$ and corresponding treatment probabilities $[e_1, e_2, \ldots, e_K]$ to target customer i with probability e_k for which $\theta_{k-1} < S(x_i) \leq \theta_k$. As above, we require $e_k \in (0; 1)$ and $\sum_k e_k = 1$. By increasing the number of thresholds K, we approximate a continuous mapping $M : S(x) \rightarrow e$, where each customer is assigned an individual treatment probability e_i proportional to the individual model score.
The specific mapping from model scores to treatment probability should follow the requirements of the application. We propose to determine the mapping by defining a set of k equal-sized intervals on the range of the model score in the training data $[\min(S(X_{train}))$, $\max(S(X_{train}))]$ and assigning a linearly increasing treatment probability e_k to each interval, while setting the lowest treatment probability at $e_1 = 0.05$ and the highest at $e_K = 0.95$. Note that asymmetric mappings result in a controlled shift of average treatment probability. The design of the mapping thus allows the straightforward extension to imbalanced supervised randomization. We reiterate that supervised randomization (Algorithm 14) randomly assigns each customer to the treatment or control group, but adjusts the probability of this assignment based on the output of a scoring model, so that customers with higher score are treated with higher probability. The assigned individual treatment probabilities are logged and used in the subsequent analyses.

6.4.2 Inverse Probability Weighting

Using the targeting model to adjust the individual probability to receive treatment introduces a sampling bias into the experiment. The sampling bias is a direct result from the violation of independence between treatment probability and the individual characteristics via the scoring model. This type of selection bias commonly occurs in observational studies, where customers self-select into the treatment group, or in natural experiments. In both situations, the sample shows measurable distributional differences between the control and treatment group. Subsequent evaluation or model building need to correct for the selection

Algorithm 14: Supervised Randomization for a Controlled Experiment with K Treatments

```
Input: Scoring model $S(\cdot)$; Treatment probability mapping $M(\cdot)$
Output: Treatment probability $e_{i,k}$; Treatment assignment $D_i \in \{0, 1, \dots, K\}$; Outcome $Y_i$
for $i$ = 1, ..., $N$ do
    Observe customer $X_i$
    Calculate customer score $s_{i,k} = S(X_i)$
    Set treatment probability $e_{i,k} = M(s_{i,k})$
    Draw treatment $D_i \sim \text{Categorical}(e_{i,k})$
    if $D_i == 0$ then
        Do not treat individual $i$
        Observe outcome $Y_i(0)$
    else
        for $k$ in $1, \dots, K$ do
            if $D_i == k$ then
                Treat individual $i$ with treatment $k$
                Observe outcome $Y_i(k)$
            end
        end
    end
end
```

bias to ensure unbiased estimates of the treatment effect. We will discuss IPW as a method that is easily integrated into model building and evaluation and discuss the doubly robust estimator as a recent extension. For a comprehensive overview of approaches including IPW see (Knaus et al., 2020). The idea underlying all approaches is to weight each observation in the treatment or control group by the inverse of its respective probability to be assigned to the observed group.

In contrast to observational studies where the treatment probability is estimated, the true probability at which customers receive the treatment is assigned actively based on a scoring model and a set of observed variables and is consequently known exactly under supervised randomization. Without the need to estimate the treatment probability from the data, we avoid confoundedness due to unobserved variables or misspecification of the propensity model by design.

IPW restores the hypothetical distribution as it would look like in a fully randomized experiments by weighting every customer with regard to the individual treatment probability. Intuitively, customers who were assigned by chance to the treatment group, even though their characteristics result in a low treatment probability, are underrepresented in the treatment group. IPW assigns these customers a higher weight. For example, if the probability of being in the treatment group for a customer is $e(x) = 0.2$ then the observed outcome if this customer received treatment is multiplied by $1/e(x) = 1/0.2 = 5$. Vice versa, if the same customer was assigned to the control group, which happened with a probability of $1 - e(x)) = 0.8$, the customer's outcome in the control group is weighted by $1/0.8 = 1.25$. The IPW corrected ATE can then be estimated as:

$$\widehat{ATE}_{IPW} = \frac{1}{N}\left(\sum_{i=1}^{N}\frac{D_iY_i}{e(X_i)} - \sum_{i=1}^{N}\frac{(1-D_i)Y_i}{1-e(X_i)}\right). \tag{6.3}$$

In observational studies, the propensity scores are unknown and need to be estimated from observed covariates. The *doubly robust* (DR) estimator is consistent and unbiased if only one of the models, the regression or the propensity score, is correctly specified (Lunceford & Davidian, 2004):

$$\widehat{ATE}_{DR} = \frac{1}{N}\sum_{i=1}^{n}\frac{D_iY_i - (D_i - e(X_i))\, g_1(X_i)}{e(X_i)} - \frac{1}{N}\sum_{i=1}^{n}\frac{(1-D_i)\,Y_i + (D_i - e(X_i))\, g_0(X_i)}{1-e(X_i)}. \tag{6.4}$$

Here $g_D(X_i) = \mathsf{E}(Y|D, X_i = x)$ are models of the outcome variable on x, estimated separately for $D \in \{0, 1\}$.

Adjusting for the propensity score under full randomization has no effect on the point estimate for the average treatment effect. However, there is some evidence that even in fully randomized experiments the large-sample variance of the estimate can be reduced by using estimated propensity scores to control for random imbalance in covariates as well as orthogonalization with the mean as in the doubly robust (Williamson et al., 2014).

6.5 Empirical Evaluation

We evaluate the proposed randomization procedures through a simulation study designed to represent a direct marketing setting, which is a common application of uplift modeling in marketing (see Devriendt et al., 2018; Radcliffe, 2007).[1]

Since IPW correction with the true propensity score is feasible, ATE and ITE estimates are consistent under supervised randomization. The goal of the empirical study is to compare the increased conversion rate and cost savings due to supervised randomization with the loss in efficiency due to a less balanced sample. The efficiency of each randomization procedure has two dimensions, that is, 1) monetary cost of the experiment and 2) the quality of models trained on the data collected during the experiment measured on downstream tasks.
First, the campaign profit during the experiment provides a metric on which to compare the opportunity cost of different experimental designs. We compare the campaign profit under

[1] The R code for the empirical evaluation is available at https://github.com/Humboldt-WI/supervised_randomization.

supervised randomization to the baseline of full randomization, which provides optimal data quality, and expect opportunity costs to be lower under the proposed supervised randomization.
Second, we evaluate the data generated from the experiment by comparing the predictive performance of ITE estimators trained on data under supervised randomization to the same estimators trained on data under full randomization. Our metrics of model performance are the mean absolute error to the true treatment effect (MAE), which is known in this simulation study but unknown in real-world settings, and the Qini coefficient, which is a standard metric in the uplift literature. The Qini coefficient is a rank metric similar to model lift based on the group-wise difference in conversion rates for customers ranked by their estimated treatment effect (Radcliffe, 2007).

6.5.1 Simulation Design

We compare the ATE and ITE estimates on experimental data collected under full and supervised randomization. An online evaluation of randomization procedures is challenging since it requires running a randomized experiment for each experimental design. We therefore evaluate the supervised randomization design in an offline study and leave online testing for future research. Our empirical Monte Carlo study uses real data to the extent possible to ensure a realistic setting in which we simulate the treatment effect and have full control over the treatment assignment (Knaus et al., 2020; Nie & Wager, 2020). The UCI Bank Marketing dataset (Moro et al., 2014) provides data on 45,211 customers of a Portuguese bank through 17 continuous or categorical variables covering individual socio-demographic and financial information, campaign details and macroeconomic indicators. All customers were subject to a phone marketing campaign promoting a term deposit and the target variable indicates if a customer has agreed to a deposit following the campaign.

Based on the available data, we simulate the individual treatment effect and hypothetical outcomes following the procedure of Nie and Wager (2020). The treatment effect in real data can be a complex, non-linear function of a subset of observed variables and unobserved variables (Farrell et al., 2021). Therefore, we simulate the treatment effect as a combination of the twelve variables containing personal or macroeconomic information. The treatment effect as a non-linear combination of covariates is then modelled by a neural network of one hidden layer with the number of nodes equal to the number of input variables and sigmoid activation, initialized with random weights drawn from a standard Gaussian. To simulate the existence of unobserved covariates, e.g. due to privacy concerns, we remove variables with personal information on the customers' age and marital status from the subsequent analysis.
In marketing settings, we further expect the ATE to be positive but small and the ITE to be mostly non-negative as marketing theory suggests a direct marketing campaign to increase

overall conversion, with potentially zero but rarely negative impact on customers (Hitsch & Misra, 2018). We center the simulated ITE distribution at an ATE of 5% and scale the standard deviation to 0.04 for 89% of simulated ITE to be positive. For our application, an ATE of 5% implies that the telephone campaign will convince an additional 5% of randomly targeted customer to register a term deposit. Because all customers in the observed data have received the marketing treatment, we simulate the potential outcome without treatment by flipping outcome labels for observations chosen randomly in proportion to their treatment effect as in Nie and Wager (2020).

Supervised randomization integrates an existing customer scoring model into the experimental design. A more accurate estimate from the existing model increases the extent to which potential cost savings are realized during experimentation. Noisier estimates of the scoring model lead to treatment assignment that is less profitable but has less influence on downstream tasks. In particular, supervised randomization with a noisy scoring model samples more evenly in the covariate space, thus mitigating the efficiency loss in downstream tasks, assuming a stochastic estimation error. We control the quality of the existing targeting model by simulating a noisy causal model with predictions $\hat{\tau}_i = \tau_i + \varepsilon$ and $\varepsilon \sim \mathcal{N}(0, \sigma)$. We report results for σ=0.025, such that customers with ITE equal to the ATE have a 95% chance to receive a prediction in the range [0,0.1] between the true and predicted treatment effect to provide a conservative estimate of the cost savings from supervised randomization. We split the data into four folds for cross validation and randomly assign treatment to each observation in the training data according to full or supervised randomization. For ITE estimation, we then estimate the ITE model on the training data and evaluate its prediction on the holdout fold. Since the random treatment assignment introduces additional randomness into the evaluation, we repeat the treatment assignment 50 times for each holdout fold and report the average over a total of 200 repetitions.

6.5.2 Statistical Model Performance Analysis

We first establish the effectiveness of supervised randomization independent of any application-specific cost setting. We evaluate the cost efficiency during experimentation through a comparison of conversion rates, ATE estimates based on their variance and ITE estimates based on uplift-specific performance metrics.

For cost efficiency, Table 6.2 reports the mean target fraction and conversion rate for full randomization at equal probability, full but imbalanced randomization with treatment probability 0.66 and the proposed supervised randomization procedure. We provide statistics on targeting no or all customers for context. However, targeting no or all customers and other non-randomized targeting strategies do not allow experimental data collection. In other words, settings *None* and *All* are inapplicable in practice for targeting policy evaluation or treatment effect estimation.

Table 6.2: Ratio of targeted customers and corresponding conversion rate under each randomization procedure. *None*/*All* denote targeting no/all customers for reference

	None	Full	*Supervised*	Full (Imb.)	All
Targeted Fraction of Customers	0.000	0.500	*0.500*	0.666	1.000
Conversion Rate	0.109	0.135	*0.143*	0.143	0.160

The target fractions for full randomization is 0.5 by definition and for imbalanced full randomization 0.66 by design. Small deviations from the target fraction are possible since treatment assignment is randomized on the individual level. The direction and ratio of the imbalance between the size of treatment to control group are in practice set by the experimenter to match the expected average treatment effect or marketing requirements, e.g. campaign budget. We chose a ratio of 2:1 in favor of targeting a larger group of customers following the most common design observed for customer targeting data in related research (see Table 6.1).

The conversion rate under each randomization provides an indirect measure of the campaign success with a higher conversion rate as an indicator of monetary returns. The increase in conversion rate from targeting no customers at 10.9% to targeting all customers at 16% reflects the simulated positive average treatment effect, specifically that customers are on average 5 percentage points more likely to convert after receiving the marketing treatment. During a fully randomized experiment, we observe an increase in the overall conversion rate by 2.6 percentage points to 13.5%. At the same fraction of customers targeted, the proposed supervised randomization increases the conversion rate by another 0.8 points to 14.3%. The improvement due to supervised randomization is the direct result of adjusting each customer's probability to be treated based on the targeting model and targeting customers with a high predicted treatment effect.

The benchmark strategy, imbalanced full randomization, increases treatment probability for all customers indiscriminately. The increase in individual treatment probability results in a conversion rate increase by 0.8 percentage points, identical to the increase under full randomization, but at a higher fraction of customers targeted. The managerial implication is that supervised randomization achieves the same conversion rate as current best practice, while reducing the targeting rate with its associated costs by 24%.

The higher conversion rate from targeting randomization towards relevant customers comes at the downside of collecting less data for customer groups with very high or very low treatment probability. While we can use the logged treatment probabilities to optimally correct for the sampling bias that is introduced by supervised randomization, estimates of the treatment effect will exhibit higher uncertainty through higher variance. Figure 6.2 shows the estimated ATE under each randomization procedure. We see that 1) deviations from full randomization in the form of imbalanced full randomization and supervised randomization return unbiased estimates and 2) the overall variance from the true value and the number of extreme deviations increases when moving from full randomization to supervised randomization.

A Kruskal-Wallis test verifies that there is no significant difference in the mean point estimate among the four settings (*df*=3, $\chi^2 = 1.00$). We are thus able to verify the theoretical exposition and to show that the selection bias introduced by supervised randomization can be corrected for by applying either IPW or DR as described above. The additional uncertainty due to supervised randomization is less pronounced when using DR to correct for heterogeneous treatment probabilities instead of IPW. DR estimates exhibit a significantly lower variance when compared to IPW estimates, based on a Levene-test for homogeneity of variance (*df*=1, F=10.29).

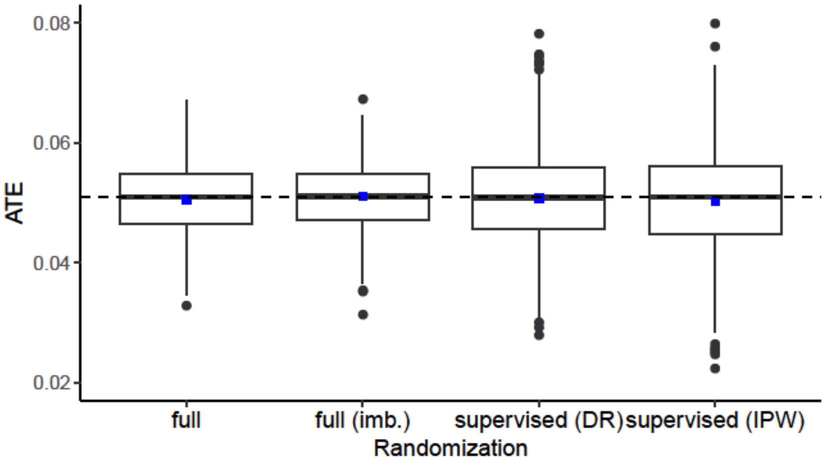


Figure 6.2: Estimated average treatment effect averaged over 200 iterations for each randomization procedure. The dashed horizontal line denotes the simulated true average treatment effect, dots within each boxplot denote the mean estimated ATE

As uplift applications are concerned with the estimation of individualized treatment effects for customer scoring, we proceed to evaluate the model performance of two causal models on the data collected under each randomization procedure. We select the two-model approach using logistic regression and the causal forest and report the performance of using the ATE as a constant prediction for reference. Since our focus is on the comparison of the randomization procedures rather than a comparison of ITE estimators, we manually set the parameters for the causal forest as follows: number of variables tried at each split (mtry) = 7, number of trees = 500, minimum node size = 20 and sample fraction for honest tree building = 0.5. Model predictions are evaluated using the mean absolute error to the true ITE and the Qini score on holdout data.

Table 6.3: Average profit-agnostic performance of causal models for each randomization procedure. We evaluate the models using the MAE to the (simulated) true treatment effect (lower is better) and the Qini coefficient (higher is better)

	ATE		Two-Model (LR)		Causal Forest	
	MAE	Qini	MAE	Qini	MAE	Qini
Full	0.0324	-	0.0353	0.0045	0.0276	0.0056
Full (imb.)	0.0324	-	0.0357	0.0045	0.0275	0.0057
Supervised	*0.0325*	-	*0.0383*	*0.0041*	*0.0295*	*0.0047*

We identify two takeaways in Table 6.3. First, the causal forest outperforms the two-learner approach on both MAE and Qini. The performance difference is consistent over all randomization procedures with the causal forest resulting in a MAE lower by about 0.008 points and a Qini higher by 0.001 points. Second, we observe that deviating from full randomization to supervised randomization leads to the expected decrease in model performance. Under supervised randomization, the difference to full randomization for the two-model approach is 0.003 points MAE and 0.0004 points Qini and for the causal forest 0.002 points MAE and 0.001 points Qini. Imbalanced full randomization at $e = 0.66$ shows no substantial performance decrease compared to balanced full randomization, although additional experiments indicate lower performance at higher levels of imbalance. The subsequent profit-based analysis aims to provide a comprehensible evaluation of the observed differences in a business context.

6.5.3 Profit Analysis

We proceed to empirically show the extent to which supervised randomization can reduce the cost of running a randomized experiment and the size of the expected trade-off measured by the performance of models trained on the collected data. The profit setting for telephone marketing is described by the gross profit resulting from a conversion and the variable contact cost of making a call to the customer. If we assume a constant interest margin for the bank, the gross profit from a one-year term deposit Ω is equivalent to the net interest margin m and the deposit amount A, $\Omega_i = mA_i$.

The incremental gross profit due to a marketing campaign is defined as change in the conversion probability, the treatment effect, to earn the gross profit on conversion minus the contact cost c, i.e. $\Delta\Omega_i = \tau_i mA_i - c$.

Given an accurate estimate of the treatment effect τ_i, the decision to target a specific customer is profitable when the predicted incremental gross profit for the customer is positive, i.e. $\hat{\tau}_i mA_i - c > 0$.

To simplify interpretation, we consider cost ratios in the range of $[5, 10, \ldots, 50]$ to 1. Evaluation over a range of cost settings ensures the robustness of our results and allows generalization to a variety of profit and cost scenarios that may arise across banks or industries, e.g. for catalog marketing. We can empirically confirm the plausibility of the range of cost ratios by analyzing the ratio of customers which are targeted under each cost setting. For cost ratios below 10:1 and above 50:1, individual targeting policies are dominated by indiscriminate targeting of no or all customers, respectively. The cost ratio corresponds to different values of the interest margin m and deposit amount A at standardized contact cost. Assuming a constant amount of the term deposit $\overline{A}$ for each customer, the cost ratio can be interpreted as the ratio between the gross profit over a range of interest margins m standardized to contact costs of $c = 1$ per contact.

We evaluate the cost-saving potential of using supervised randomization during experimentation based on the campaign profit resulting from a randomized experiment for each randomization procedure. We report the campaign profit per prospective customer and the difference in campaign profit relative to full randomization in Table 6.4. As above, we include targeting no customers and targeting all customers for reference, but stress that non-randomized targeting strategies do not allow experimental data collection, making them inapplicable for causal modeling in practice.

Table 6.4: Campaign profit (per customer) for randomized experiments under each randomization procedure and across purchase margins. *None*/*All* denote targeting no/all customers for reference. *Full (Imb.)* denotes full randomization with a treatment probability of 66%

Conversion	Campaign profit per customer (€)				
Value (€)	None	Full	*Supervised*	Full (Imb.)	All
10	1.09	0.85	*0.93*	0.76	0.60
15	1.64	1.52	*1.65*	1.48	1.40
20	2.18	2.19	*2.37*	2.19	2.20
25	2.73	2.86	*3.09*	2.91	3.00
30	3.27	3.54	*3.80*	3.62	3.80
35	3.82	4.21	*4.52*	4.34	4.60
40	4.36	4.88	*5.24*	5.05	5.40
45	4.91	5.56	*5.96*	5.77	6.20
50	5.45	6.23	*6.67*	6.48	7.00

The empirical results in Table 6.4 support the proposition that supervised randomization increases the campaign profit during experimentation relative to full randomization for the full range of conversion values we consider in this study. In relative terms, supervised randomization increases the experimental campaign profit by 7.1–9.4% compared to full randomization and by 2.9–8.2% compared to imbalanced randomization.

For a conversion value of € 10, we observe a marginal profit of € 0.85 per customer under full randomization and a marginal profit of € 0.93 under supervised randomization. The absolute increase in campaign profit is more pronounced when the cost ratio is higher. A value of € 50 corresponds to a marginal profit per customer of € 6.23 under full randomization compared to € 6.67 under the proposed supervised randomization. Cost savings per customer compared to full randomization amount to € 0.08 and € 0.44, respectively. We translate the per customer savings to an experimental campaign of 40,000 prospective customers, who are randomly targeted. This is the size of the observed telephone marketing campaign and, with less observations than 9 of the 11 experimental marketing campaigns summarized in Table 6.1, may provide a conservative estimate. The total cost savings per experiment when replacing full randomization with supervised randomization translate to € 3,200 for a marginal profit of € 10, € 10,400 for a marginal profit of € 30 and € 17,600 for a marginal profit of € 50. Experiment costs and the related savings arise whenever data is collected for policy evaluation or (re-)estimation of the customer scoring model.

For conversion values greater or equal € 20, targeting all customers is more profitable than not targeting any customer. The imbalanced full randomization, which we identify as standard in practice, is more profitable than full randomization only at values above € 20. At these values, imbalanced randomization achieves savings of 0 to € 0.25 per customer compared to full randomization for conversion values between € 20 and € 50, respectively. Compared to imbalanced full randomization, the proposed supervised randomization generates additional cost savings per customer of about € 0.18 for all values between € 20 and € 50. Again translated to an experiment campaign of 40,000 prospective customers, the total cost savings per experiment of supervised randomization when compared to the industry-standard range from € 7,200 for a marginal profit of € 20 to € 7,600 for a marginal profit of € 50. Note that it is possible to combine supervised randomization with imbalanced targeting. Increasing the average treatment probability through a custom treatment probability mapping may further increase campaign profit in settings where treatment is highly profitable.

Having discussed the expected cost savings during experimentation, we next discuss the opportunity costs on downstream tasks associated with the increase in model uncertainty under supervised randomization. We first report the campaign profit per customer when customers are targeted by the two-model approach or causal forest and each model is trained on experimental data collected under the different randomization procedures.

Table 6.5: Campaign profit using targeting models trained on data collected under each randomization procedure. We evaluate the campaign profit per customer over a range of cost ratios

Conversion	Two-Model (Logit)			Causal Forest		
Value (€)	Simple	Simple (Imb.)	*Supervised*	Simple	Simple (Imb.)	*Supervised*
10	1.06	1.06	*1.06*	1.09	1.09	*1.09*
15	1.65	1.65	*1.65*	1.66	1.67	*1.65*
20	2.33	2.33	*2.32*	2.36	2.36	*2.33*
25	3.07	3.06	*3.05*	3.11	3.12	*3.08*
30	3.83	3.82	*3.80*	3.89	3.89	*3.84*
35	4.60	4.60	*4.57*	4.67	4.67	*4.62*
40	5.38	5.38	*5.35*	5.45	5.45	*5.41*
45	6.16	6.16	*6.13*	6.24	6.24	*6.20*
50	6.95	6.95	*6.91*	7.03	7.03	*7.00*

Table 6.5 shows that the expected decrease in profit for scoring models trained on data collected under supervised randomization is small but observable in the order of 1% of the absolute campaign profit per customer. For a basket margin of € 30, the two-model logistic regressions achieve a campaign profit of € 3.83 per customer under full randomization and a campaign profit of € 3.80 under supervised randomization, a decrease of 0.8%. The causal forest achieves a campaign profit of € 3.89 when trained on data from experiments under full randomization with a decrease by 1.3% to € 3.84 under supervised randomization. Compared over all values, supervised randomization induces a decrease in per customer profit between € 0 and € 0.04 for the two-model approach and € 0 and € 0.05 for the causal forest compared to full randomization.

6.6 Conclusion

Customer targeting is a continuously growing and widely studied application of scoring models. While research has focused on the prediction of future customer behavior to inform decision-making, a growing research stream has established uplift models to estimate the causal effect of a marketing action on each customer based on observed customer characteristics. The training and evaluation of causal models require data collected through experiments, in which customers are randomly assigned to treatments. However, experimental data collection incurs high costs by temporarily replacing an established targeting policy with random targeting.

We propose supervised randomization as a solution to reduce the cost of experimentation by integrating an existing scoring model into the experimental design. By mapping model scores to individual treatment propensities, we are able to target more profitable customers while maintaining stochastic treatment assignment. An empirical Monte Carlo study on telemarketing shows that supervised targeting can reduce the cost of an experimental campaign on 40,000 prospective customers by 7.1–9.4% compared to full randomization and 2.9–8.2% compared to imbalanced randomization, depending on the specific profit-cost ratio.

Active management of treatment assignment during experimentation leads to an overrepresentation of profitable customers in the treatment group, which causes selection bias when standard estimators are applied to estimate treatment effects. We consequently summarize inverse probability weighting and doubly robust estimation as well-studied methods to correct for selection bias when estimating average and individualized treatment effects. We show that the estimated treatment effects are unbiased and provide indicators of the increase in uncertainty related to supervised randomization. Empirical evaluation indicates that higher uncertainty of the scoring model may lead to a decrease in campaign profit by 0.8–1.3% depending on the specific profit-cost ratio. Further evaluation in real-world experiments is necessary to establish net cost savings in practice.

Overall, we argue that the methodology developed in the medical and econometric literature has not yet been fully studied and applied in the uplift setting. Doubly robust estimation serves as one example of a wider set of tools to correct for selection issues in the data. We further identify experimental data collection as a fundamental part of causal modeling. We expect that supervised randomization provides a first step towards a wider analysis of practical experimental design.

Bibliography

Ascarza, E. (2018). Retention futility: Targeting high risk customers might be ineffective. *Journal of Marketing Research*, *55*(1). https://doi.org/10.1509/jmr.16.0163

Ascarza, E., Ebbes, P., Netzer, O., & Danielson, M. (2017). Beyond the target customer: Social effects of customer relationship management campaigns. *Journal of Marketing Research*, *54*(3), 347–363. https://doi.org/10.1509/jmr.15.0442

Athey, S., & Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, *113*(27), 7353–7360. https://doi.org/10.1073/pnas.1510489113

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *Annals of Statistics*, *47*(2), 1148–1178. https://doi.org/10.1214/18-AOS1709

Athey, S., & Wager, S. (2017). Efficient policy learning. *arXiv preprint arXiv:1702.02896.*

Chickering, M., & Heckerman, D. A Decision Theoretic Approach to Targeted Advertising. In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI'00).* Morgan Kaufmann, 2000, 82–88.

Devriendt, F., Moldovan, D., & Verbeke, W. (2018). A literature survey and experimental evaluation of the state-of-the-art in uplift modeling: A stepping stone toward the development of prescriptive analytics. *Big Data*, *6*(1), 13–41. https://doi.org/10.1089/big.2017.0104

Diemert, E., Betlei, A., Renaudin, C., & Amini, M.-R. A Large Scale Benchmark for Uplift Modeling. In: *Proceedings of the AdKDD and TargetAd Workshop, KDD.* London, United Kingdom: ACM, 2018.

Farrell, M. H., Liang, T., & Misra, S. (2021). Deep neural networks for estimation and inference. *Econometrica*, *89*(1), 181–213. https://doi.org/10.3982/ecta16901

Gordon, B. R., Zettelmeyer, F., Bhargava, N., & Chapsky, D. (2019). A comparison of approaches to advertising measurement: Evidence from big field experiments at Facebook. *Marketing Science*, *38*(2), 193–364. https://doi.org/10.1287/mksc.2018.1135

Gubela, R. M., Bequé, A., Gebert, F., & Lessmann, S. (2019). Conversion uplift in e-commerce: A systematic benchmark of modeling strategies. *International Journal of Information Technology & Decision Making*, *18*(3), 747–791. https://doi.org/10.1142/s0219622019500172

Gubela, R. M., Lessmann, S., Haupt, J., Baumann, A., Radmer, T., & Gebert, F. Revenue Uplift Modeling. In: *Proceedings of the 38th International Conference on Information Systems (ICIS).* AIS, 2017.

Guelman, L. (2014). *Optimal Personalized Treatment Learning Models with Insurance Applications* (Doctoral Thesis). Universitat de Barcelona.

Guelman, L., Guillén, M., & Pérez-Marín, A. M. (2015). Uplift Random Forests. *Cybernetics and Systems*, *46*(3-4), 230–248. https://doi.org/10.1080/01969722.2015.1012892

Hansotia, B. J., & Rukstales, B. (2002). Direct marketing for multichannel retailers: Issues, challenges and solutions. *Journal of Database Marketing & Customer Strategy Management*, *9*(3), 259–266. https://doi.org/10.1057/palgrave.jdm.3240007

Hansotia, B., & Rukstales, B. (2002). Incremental value modeling. *Journal of Interactive Marketing*, *16*(3), 35–46. https://doi.org/10.1002/dir.10035

Hillstrom, K. (2008). *The MineThatData E-Mail Analytics and Data Mining Challenge.*

Hitsch, G. J., & Misra, S. (2018). Heterogeneous treatment effects and optimal targeting policy evaluation. *Available at SSRN 3111957*. https://doi.org/10.2139/ssrn.3111957

Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, *47*(260), 663–685. https://doi.org/10.1080/01621459.1952.10483446

Imbens, G. W., & Wooldridge, J. M. (2009). Recent developments in the econometrics of program evaluation. *Journal of Economic Literature*, *47*(1), 5–86. https://doi.org/10.1257/jel.47.1.5

Kane, K., Lo, V. S. Y., & Zheng, J. (2014). Mining for the truly responsive customers and prospects using true-lift modeling: Comparison of new and existing methods. *Journal of Marketing Analytics*, *2*(4), 218–238. https://doi.org/10.1057/jma.2014.18

Knaus, M. C., Lechner, M., & Strittmatter, A. (2020). Machine Learning Estimation of Heterogeneous Causal Effects: Empirical Monte Carlo Evidence. *The Econometrics Journal*. https://doi.org/10.1093/ectj/utaa014

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, *116*(10), 4156–4165. https://doi.org/10.1073/pnas.1804597116

Lachin, J. M., Matts, J. P., & Wei, L. J. (1988). Randomization in clinical trials: Conclusions and recommendations. *Controlled Clinical Trials*, *9*(4), 365–374. https://doi.org/10.1016/0197-2456(88)90049-9

Lo, V. S. Y. (2002). The true lift model: A novel data mining approach to response modeling in database marketing. *ACM SIGKDD Explorations Newsletter*, *4*(2), 78–86. https://doi.org/10.1145/772862.772872

Lunceford, J. K., & Davidian, M. (2004). Stratification and weighting via the propensity score in estimation of causal treatment effects: A comparative study. *Statistics in Medicine*, *23*(19), 2937–2960. https://doi.org/10.1002/sim.1903

Marco Caliendo, Michel Clement, Dominik Papies, & Sabine Scheel-Kopeinig. (2012). The cost impact of spam filters: Measuring the effect of information system technologies in organizations. *Information Systems Research*, *23*(3), 1068–1080. https://doi.org/10.1287/isre.1110.0396

Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, *62*, 22–31. https://doi.org/10.1016/j.dss.2014.03.001

Nie, X., & Wager, S. (2020). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*. https://doi.org/10.1093/biomet/asaa076

Olson, D. L., Delen, D., & Meng, Y. (2012). Comparative analysis of data mining methods for bankruptcy prediction. *Decision Support Systems*, *52*(2), 464–473. https://doi.org/10.1016/j.dss.2011.10.007

Powers, S., Qian, J., Jung, K., Schuler, A., Shah, N. H., Hastie, T., & Tibshirani, R. (2018). Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in Medicine*, *37*(11), 1767–1787. https://doi.org/10.1002/sim.7623

Radcliffe, N. J. (2007). Using control groups to target on predicted lift: Building and assessing uplift models. *Direct Marketing Analytics Journal*, 14–21.

Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1994). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, *89*(427), 846–866. https://doi.org/10.1080/01621459.1994.10476818

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*(1), 41–55. https://doi.org/10.1093/biomet/70.1.41

Rosenberger, W. F., & Lachin, J. M. (1993). The use of response-adaptive designs in clinical trials. *Controlled Clinical Trials*, *14*(6), 471–484. https://doi.org/10.1016/0197-2456(93)90028-C

Rzepakowski, P., & Jaroszewicz, S. (2012). Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems*, *32*(2), 303–327. https://doi.org/10.1007/s10115-011-0434-0

Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., & Joachims, T. Recommendations as Treatments: Debiasing Learning and Evaluation. In: *Proceedings of the 33rd International Conference on Machine Learning*. *48*. 2016, 1670–1679.

Schulz, K. F., & Grimes, D. A. (2002). Generation of allocation sequences in randomised trials: Chance, not choice. *The Lancet*, *359*(9305), 515–519. https://doi.org/10.1016/S0140-6736(02)07683-3

Schwartz, E. M., Bradlow, E. T., & Fader, P. S. (2017). Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, *36*(4), 500–522. https://doi.org/10.1287/mksc.2016.1023

Statista. (2017). *Advertising Spending in the Catalog, Mail-order Houses Industry in the United States* (tech. rep.).

Statista. (2019). *eCommerce* (tech. rep.).

Swaminathan, A., & Joachims, T. (2015). Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, *16*, 1731–1755.

Williamson, E. J., Forbes, A., & White, I. R. (2014). Variance reduction in randomised trials by inverse probability weighting using the propensity score. *Statistics in Medicine*, *33*(5), 721–737. https://doi.org/10.1002/sim.5991

Zaniewicz, L., & Jaroszewicz, S. Support Vector Machines for Uplift Modeling. In: *13th International Conference on Data Mining Workshops*. IEEE, 2013, 131–138. https://doi.org/10.1109/icdmw.2013.23.

Selbständigkeitserklärung

Ich bezeuge durch meine Unterschrift, dass meine Angaben über die bei der Abfassung meiner Dissertation benutzten Hilfsmittel, über die mir zuteil gewordene Hilfe sowie über frühere Begutachtungen meiner Dissertation in jeder Hinsicht der Wahrheit entsprechen.

Daniel Jacob, Berlin 06.08.2021