

**Causality Concepts in Machine Learning:
Heterogeneous Treatment Effect Estimation
with Machine Learning & Model Interpretation
with Counterfactual and Semi-factual Explanations**

Susanne Dandl

München 2023



**Causality Concepts in Machine Learning:
Heterogeneous Treatment Effect Estimation
with Machine Learning & Model Interpretation
with Counterfactual and Semi-factual Explanations**

Susanne Dandl

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

eingereicht von
Susanne Dandl
am 18.09.2023

Erster Berichterstatter: Prof. Dr. Bernd Bischl
Zweiter Berichterstatter: Prof. Dr. Torsten Hothorn
Dritter Berichterstatter: Prof. Dr. Marvin N. Wright

Tag der Disputation: 06.12.2023

Acknowledgments

I would like to express my sincere thanks to all those who have supported and advised me throughout this incredible journey of pursuing my Ph.D. In particular, I am deeply grateful to ...

- ... Prof. Dr. Bernd Bischl for arousing my interest in machine learning during my Master's program and for providing invaluable supervision, advice, and support throughout my Ph.D.*
- ... Prof. Dr. Torsten Hothorn for his insights and guidance, which opened up new perspectives and greatly shaped my research path.*
- ... Prof. Dr. Marvin N. Wright for his willingness to be the third reviewer of my Ph.D. thesis.*
- ... Prof. Dr. Frauke Kreuter and Prof. Dr. Christian Heumann for their availability to be part of the examination committee.*
- ... Prof. Dr. Achim Zeileis, Prof. Dr. Stefan Wager, and Dr. Erik Sverdrup for the fruitful discussions and valuable contributions to our joint projects despite the physical distance between our universities.*
- ... Giuseppe Casalicchio, Ludwig Bothmann, and Andreas Bender for their guidance throughout the research projects and their supervision of the research subgroups I was part of, which heavily inspired me.*
- ... Heidi Seibold, Cornelia Fütterer, Christoph Molnar, and Martin Binder for their encouragement to pursue this academic path.*
- ... all my current and former colleagues at the chair of Statistical Learning and Data Science for being a source of inspiration and for their contributions to joint projects.*
- ... all research fellows at the Department of Statistics for creating a supportive environment for exchanging ideas in the form of mensa visits, tea talks, summer retreats, ...*
- ... my friends and my family for their unwavering support throughout this journey.*

Summary

Over decades, machine learning and causality were two separate research fields that developed independently of each other. It was not until recently that the exchange between the two intensified. This thesis comprises seven articles that contribute novel insights into the utilization of causality concepts in machine learning and highlights how both fields can benefit from one another.

One part of this thesis focuses on adapting machine learning algorithms for estimating heterogeneous treatment effects. Specifically, random forest-based methods have demonstrated to be a powerful approach to heterogeneous treatment effect estimation; however, understanding the key elements responsible for that remains an open question. To provide answers, one contribution analyzed which elements of two popular forest-based heterogeneous treatment effect estimators – causal forests and model-based forests – are beneficial in case of real-valued outcomes. A simulation study reveals that model-based forests’ simultaneous split selection based on prognostic and predictive effects is effective for randomized controlled trials, while causal forests’ orthogonalization strategy is advantageous for observational data under confounding. Another contribution shows that combining these elements yields a versatile model framework applicable to a wide range of application cases: observational data with diverse outcome types, potentially under different forms of censoring.

Another part focuses on two methods that leverage causality concepts to interpret machine learning models: counterfactual explanations and semi-factual explanations. Counterfactual explanations describe *minimal* changes in a few features required for changing a prediction, while semi-factual explanations describe *maximal* changes in a few features required for *not* changing a prediction. These insights are valuable because they reveal which features do or do not affect a prediction, and they can help to object against or justify a prediction. The existence of multiple equally good counterfactual explanations and semi-factual explanations for a given instance is often overlooked in the existing literature. This is also pointed out in the first contribution of the second part, which deals with possible pitfalls of interpretation methods, potential solutions, and open issues. To address the multiplicity of counterfactual explanations and semi-factual explanations, two contributions propose methods to generate multiple explanations: The underlying optimization problem was formalized multi-objectively for counterfactual explanations and as a hyperbox search for semi-factual explanations. Both approaches can be easily adapted to other use cases, with another contribution demonstrating how the multi-objective approach can be applied to assess counterfactual fairness. Despite the multitude of counterfactual methods proposed in recent years, the availability of methods for users of the programming language R remains extremely limited. Therefore, another contribution introduces a modular R package that facilitates the application and comparison of multiple counterfactual explanation methods.

Zusammenfassung

Über Jahrzehnte waren maschinelles Lernen und Kausalität zwei getrennte Forschungsbereiche, die sich unabhängig voneinander entwickelten. Erst in jüngster Zeit hat sich der Austausch zwischen den beiden Bereichen intensiviert. Diese Arbeit umfasst sieben Artikel, die neue Einblicke in die Nutzung von Kausalitätskonzepten im maschinellen Lernen geben, und zeigt, wie beide Bereiche voneinander profitieren können.

Ein Teil dieser Arbeit befasst sich mit der Anpassung von Algorithmen des maschinellen Lernens zur Schätzung heterogener Behandlungseffekte. Insbesondere Random-Forest-Methoden haben sich als leistungsfähiger Ansatz für die Behandlungseffekt-Schätzung erwiesen; das Verständnis der Schlüsselemente, die dafür verantwortlich sind, bleibt jedoch eine offene Frage. Um Antworten zu finden, wurde in einem Beitrag analysiert, welche Elemente von zwei beliebigen Random-Forest-Schätzern - Causal Forests und Model-based Forests - im Fall von reellwertigen Zielvariablen von Vorteil sind. Eine Simulationsstudie zeigt, dass die gleichzeitige Split-Auswahl von Model-based Forests auf der Grundlage von prognostischen und prädiktiven Effekten für randomisierte kontrollierte Studien effektiv ist, während die Orthogonalisierungsstrategie der Causal Forests für Beobachtungsdaten mit Confoundern von Vorteil ist. Ein weiterer Beitrag zeigt, dass die Kombination dieser Elemente ein vielseitiges Framework für Modelle ergibt, welches auf viele verschiedene Fälle anwendbar ist: Beobachtungsdaten mit verschiedenen Arten von Zielvariablen, möglicherweise unter verschiedenen Formen von Zensierung.

Ein weiterer Teil dieser Arbeit konzentriert sich auf zwei Methoden, die Kausalitätskonzepte zur Interpretation von Modellen des maschinellen Lernens nutzen: Counterfactual Explanations (kontrafaktische Erklärungen) und Semi-factual Explanations (semi-faktische Erklärungen). Counterfactual Explanations beschreiben *minimale* Änderungen in einigen wenigen Merkmalen, die für die Änderung einer Vorhersage erforderlich sind, während Semi-factual Explanations *maximale* Änderungen in einigen wenigen Merkmalen beschreiben, die zu *keiner* Änderung der Vorhersage führen. Diese Erkenntnisse sind wertvoll, weil sie zeigen, welche Merkmale eine Vorhersage beeinflussen und welche nicht, und sie können helfen, eine Vorhersage zu widerlegen oder zu rechtfertigen. Die Existenz mehrerer gleich guter Counterfactual Explanations und Semi-factual Explanations für einen Datenpunkt wird in der bestehenden Literatur oft übersehen. Darauf weist auch der erste Beitrag des zweiten Teils hin, der sich mit möglichen Fallstricken von Interpretationsmethoden, möglichen Lösungen und offenen Fragen befasst. Um der Vielzahl von Counterfactual Explanations und Semi-factual Explanations zu begegnen, werden in zwei Beiträgen Methoden zur Generierung multipler Erklärungen vorgeschlagen: Das zugrundeliegende Optimierungsproblem wurde für Counterfactual Explanations multi-objektiv und für Semi-factual Explanations als Hyperbox-Suche formalisiert. Beide Ansätze können leicht an andere Anwendungsfälle angepasst werden, wobei ein weiterer Beitrag zeigt, wie der multi-objektive Ansatz zur Bewertung der Modellfairness im kontrafaktischen Sinne angewendet werden kann. Trotz der Vielzahl von Counterfactual Explanations Methoden, die in den letzten Jahren vorgeschlagen wurden, ist die Verfügbarkeit von Methoden für Nutzer der Programmiersprache R äußerst begrenzt. Daher wird in einem weiteren Beitrag ein modulares R-Paket vorgestellt, das die Anwendung und den Vergleich mehrerer Counterfactual Explanations Methoden erleichtert.

Contents

I	Introduction and Background	1
1	Overview	3
2	Introduction to Machine Learning	5
3	Heterogeneous Treatment Effect Estimation with Machine Learning	9
3.1	Causality Concept: Potential Outcomes Framework	10
3.1.1	Causal Estimand	10
3.1.2	Statistical Estimand	11
3.2	Estimation via Machine Learning Approaches	13
3.2.1	Model-Agnostic Estimators	14
3.2.2	Model-Specific Estimators	15
3.3	Beyond Continuous Outcomes	18
4	Model Interpretation with Counterfactual and Semi-factual Explanations	19
4.1	Causality Concept: Counterfactuals	20
4.2	Counterfactual Explanations	21
4.2.1	Desired Properties	21
4.2.2	Generation Methods	22
4.2.3	Connection to Counterfactual Fairness	25
4.3	Semi-factual Explanations	25
4.3.1	Desired Properties	26
4.3.2	Generation Methods	27
II	Contributions	29
5	What Makes Forest-Based Heterogeneous Treatment Effect Estimators Work?	31
6	Heterogeneous Treatment Effect Estimation for Observational Data using Model-based Forests	67
7	General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models	119
8	Multi-Objective Counterfactual Explanations	151
9	Multi-Objective Counterfactual Fairness	175
10	counterfactuals: An R Package for Counterfactual Explanation Methods	183
11	Interpretable Regional Descriptors: Hyperbox-Based Local Explanations	233
III	Conclusion and Outlook	269
12	Conclusion and Outlook	271
	References	275

Contributing Articles

- Chapter 5** Dandl S, Haslinger C, Hothorn T, Seibold H, Sverdrup E, Wager S, Zeileis A (2023b). “What Makes Forest-Based Heterogeneous Treatment Effect Estimators Work?” *arXiv 2206.10323 v2*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2206.10323. To appear in *The Annals of Applied Statistics*
- Chapter 6** Dandl S, Bender A, Hothorn T (2022a). “Heterogeneous Treatment Effect Estimation for Observational Data using Model-based Forests.” *arXiv 2210.02836*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2210.02836. To appear in *Statistical Methods in Medical Research*
- Chapter 7** Molnar C, König G, Herbinger J, Freiesleben T, Dandl S, Scholbeck CA, Casalicchio G, Grosse-Wentrup M, Bischl B (2022). “General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models.” In A Holzinger, R Goebel, R Fong, T Moon, KR Müller, W Samek (eds.), *xxAI - Beyond Explainable AI*, volume 13200 of *Lecture Notes in Artificial Intelligence*, pp. 39–68. Springer, Cham. doi:10.1007/978-3-031-04083-2_4
- Chapter 8** Dandl S, Molnar C, Binder M, Bischl B (2020). “Multi-Objective Counterfactual Explanations.” In T Bäck, M Preuss, A Deutz, H Wang, C Doerr, M Emmerich, H Trautmann (eds.), *Parallel Problem Solving from Nature – PPSN XVI*, pp. 448–469. Springer International Publishing, Cham. doi:10.1007/978-3-030-58112-1_31
- Chapter 9** Dandl S, Pfisterer F, Bischl B (2022b). “Multi-Objective Counterfactual Fairness.” In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’22, p. 328–331. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3520304.3528779
- Chapter 10** Dandl S, Hofheinz A, Binder M, Bischl B, Casalicchio G (2023c). “**counterfactuals**: An R Package for Counterfactual Explanation Methods.” *arXiv 2304.06569 v2*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2304.06569
- Chapter 11** Dandl S, Casalicchio G, Bischl B, Bothmann L (2023a). “Interpretable Regional Descriptors: Hyperbox-Based Local Explanations.” In D Koutra, C Plant, M Gomez Rodriguez, E Baralis, F Bonchi (eds.), *Machine Learning and Knowledge Discovery in Databases: Research Track (ECML PKDD 2023)*, pp. 479–495. Springer Nature Switzerland, Cham. doi:10.1007/978-3-031-43418-1_29

List of Acronyms

- BI** bivariate imputation
CATE conditional average treatment effect
CF causal forest
CFE counterfactual explanation
DR doubly robust
FE factual explanation
HTE heterogeneous treatment effect
ML machine learning
MOB model-based forest
MOC multi-objective counterfactual
PA protected attribute
POF potential outcomes framework
RF random forest
SCM structural causal model
SFE semi-factual explanation

Part I

Introduction and Background

1 Overview

Motivation Supervised machine learning (ML) is increasingly applied to various domains, encompassing medicine, ecology, and finance (MacEachern and Forkert, 2021; Humphries *et al.*, 2018; Warin and Stojkov, 2021). The success of ML was enabled by improved technological prerequisites and methodological achievements since the late 1950s, when Rosenblatt (1957) developed the first ML algorithm – the perceptron, a predecessor of neural networks. Over the decades, more and more ML algorithms were developed, for example, support vector machines and classification and regression trees in the 80s, boosting models in the 90s, random forests in the 00s, generative adversarial networks in the 2010s, and nowadays, transformer neural network architectures for large language models like ChatGPT (Vapnik, 1982; Breiman *et al.*, 1984; Schapire, 1990; Breiman, 2001a; Goodfellow *et al.*, 2014; OpenAI, 2023). The complexity of these models necessitates an advanced model analysis: performance assessments based on unseen test data to mitigate the risk of overfitting and the application of model interpretation methods that help to inspect how predictions are obtained. Such analyses are particularly crucial when ML models aid the decision-making process of highly sensitive tasks such as evaluating credit risk, screening job applicants, or diagnosing diseases.

Research in causality emerged a few decades before ML. The field provides a deeper understanding of causal relations beyond mere associations. Wright (1921) was the first to formalize causal effects mathematically and to visualize them in graphs. Splawa-Neyman *et al.* (1923) introduced a different notation of causes in the form of potential outcomes to randomized trials. Rubin extended the framework to observational data by stating identifying assumptions (Rubin, 1974, 1980), thus taking a statistical viewpoint on causality. Pearl (1995) developed a different framework based on structural causal models and their graphical representation as causal graphs. Both frameworks differ in their representation, but Pearl (2022) considers them “logically equivalent”. Nowadays, both of them are frequently used, but often within different communities (Pearl, 2022).

The short excerpts on the history reveal that research on ML and causality developed independently for many decades. It was not until recently that the exchange between the two fields intensified. The research can be distinguished into two areas: The first area inspects how ML algorithms can help in causality, e.g., with the estimation of heterogeneous treatment effects (Curth and van der Schaar, 2021) or with causal structure learning (Vowels *et al.*, 2022). The second area inspects how causality concepts can help to improve ML models, e.g., w.r.t. their robustness and generalizability (Schölkopf *et al.*, 2021), interpretability (Wachter *et al.*, 2018; Karimi *et al.*, 2021) or fairness (Kusner *et al.*, 2017).

This thesis comprises seven contributing articles that focus on two subareas (one from each of the areas above): (1) heterogeneous treatment effect (HTE) estimation using ML and (2) model interpretation with counterfactual explanations (CFEs) and semi-factual explanations (SFEs). Both topics are approached from an ML viewpoint and are seen as embedding causality concepts into the general ML workflow, which is presented in Chapter 2.

Heterogeneous Treatment Effect Estimation using Machine Learning HTEs reflect that a causal effect of a treatment is not constant over a population, but differs between individuals or subgroups. ML methods allow for HTE estimations in a flexible, non-parametric way. The causality concept underlying this is the potential outcomes framework by Rubin (1974), which is presented in Chapter 3, alongside different strategies for HTE estimation with ML algorithms. This chapter also introduces model-based forests (Seibold *et al.*, 2018) and causal forests (Athey *et al.*, 2019) – two random forest-based estimators. The contribution in Chapter 5 provides theoretical and empirical insights on what elements of these approaches are beneficial for HTE estimation and how they can be blended into a novel method that combines the best of model-based forests and causal forests. While the investigations were restricted to continuous outcomes, the contributing article in Chapter 6 discusses extensions of this blended method to diverse outcome types, forming a versatile model framework applicable to a wide range of use cases.

Model Interpretation with Counterfactual & Semi-factual Explanations CFEs and SFEs provide insights into a prediction by presenting alternative data points with a different or the same prediction, respectively. The causality concept underlying this approach are counterfactuals. Counterfactuals were considered by Rubin under the potential outcomes framework, as well as by Pearl using structural causal models. Both viewpoints are presented in Chapter 4 alongside an introduction to CFEs and SFEs – their purposes, properties, and generation methods.

Many of these generation methods only return a single explanation and, thus, ignore that multiple equally good CFEs and SFEs can exist. This is one of the many pitfalls of interpretation methods stated in the contributing article of Chapter 7. This thesis offers two solutions to address multiplicity: For CFEs, the contributing article of Chapter 8 formalizes the optimization problem underlying the generation of CFEs multi-objectively such that a diverse Pareto-set of CFEs is returned. The approach can be flexibly adapted to other use cases, as the contribution of Chapter 9 shows for counterfactual fairness (an introduction to counterfactual fairness provides Section 4.2.3). For SFEs, the contributing article of Chapter 11 formalizes the search as a hyperbox search. The returned hyperbox reflects a set of SFEs.

Both proposed generation methods are implemented in R (R Core Team, 2022), which is in sharp contrast to other methods that are predominantly available in Python (Van Rossum and Drake Jr, 1995). To facilitate the implementation of more CFE methods in R, the contribution of Chapter 10 introduces a modular, user-friendly R package that currently offers three CFE methods as well as multiple evaluation and visualization methods.

2 Introduction to Machine Learning

Machine learning encompasses three core areas: supervised and unsupervised machine learning, as well as reinforcement learning. Supervised machine learning aims to find a model that can approximate the functional relationship between inputs and an outcome such that the model accurately predicts on new unseen data. The name “supervised” originates from the knowledge of true outcome values that “guide the learning process” (Hastie *et al.*, 2009). In contrast, unsupervised machine learning aims to detect patterns in a set of features in the absence of an outcome of interest, and reinforcement learning seeks to find optimal actions by maximizing a reward function (Sutton and Barto, 2018). In the following, supervised machine learning is abbreviated as ML since unsupervised machine learning and reinforcement learning are not considered further throughout this thesis. In addition, the thesis focuses solely on tabular data and not on image or text data.

The following sections present the main steps of the (supervised) ML workflow: model training, prediction, and analysis. Figure 2.1 visualizes these steps.¹ The final paragraph of this chapter presents examples of two ML algorithms: a regression tree and a random forest. They play a crucial role in the contributions of Chapters 5 and 6.

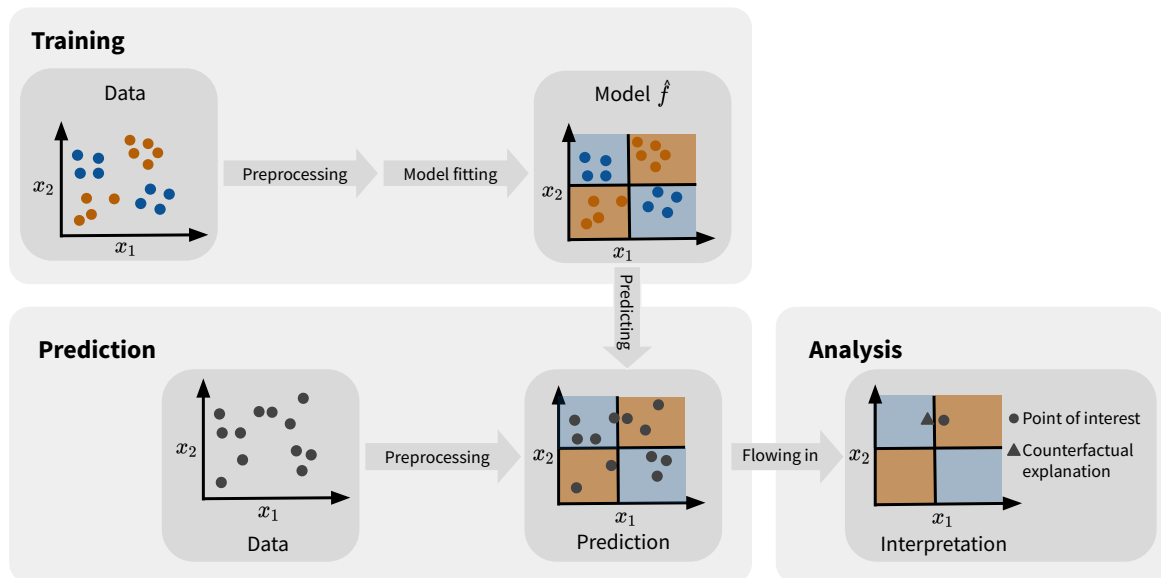


Figure 2.1: Main steps of the machine learning workflow. A two-dimensional classification data set illustrates the steps. A model is fitted (Training), applied to new data (Prediction) and interpreted by counterfactual explanations (Analysis).

¹This representation is simplified. Tuning and post-processing steps are omitted since they are not a matter of this thesis.

Training In the training step, a model is fitted to a given (potentially preprocessed) data set using a learning algorithm. In ML, the data set $\mathcal{D} = (\mathbf{x}^{(i)}, y^{(i)})_{i=1}^n$ consists of n independent and identically distributed observations. The p -dimensional vector $\mathbf{x}^{(i)} = (x_1^{(i)}, \dots, x_p^{(i)})^\top$ comprises realizations of the random variables $\mathbf{X} = (X_1, \dots, X_p)^\top$, which are called features, covariates or variables². They originate from the feature space $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_p$. Realizations of the outcome (or target) variable Y are denoted as $y^{(i)}, i \in \{1, \dots, p\}$ in \mathcal{D} . They originate from the target space \mathcal{Y} . $\mathbb{P}_{\mathbf{X}, Y}$ defines the joint probability distribution on $\mathcal{X} \times \mathcal{Y}$. Before fitting a model, the data might be preprocessed by selecting, extracting, or transforming features.

The goal of ML is to approximate the functional relationship between \mathcal{X} and \mathcal{Y} by a model $f : \mathcal{X} \rightarrow \mathbb{R}^g$ that maps $\mathbf{x} \in \mathcal{X}$ to predictions in \mathbb{R}^g with $g \in \mathbb{N}^+$. If $\mathcal{Y} = \mathbb{R}$, then $g = 1$ and f is called a regression model; if $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, +1\}$, then $g = 1$ and we search for a binary classification model which either returns hard labels, probabilities or scores; if $\mathcal{Y} = \{1, \dots, g\}$, we search for a multi-class classification model (Hastie *et al.*, 2009).

The functional family from which f originates needs to be restricted to a specific model class (e.g., to regression trees or neural networks). Otherwise, finding a best model among all potential model classes would be impossible in finite time (Mitchell, 1997). The hypothesis space \mathcal{H} denotes the set of functions that define a model class. Parameters $\boldsymbol{\theta} \in \Theta$ parameterize the models in \mathcal{H} , such that finding an optimal model is equal to finding an optimal set of parameter values $\boldsymbol{\theta}$. This optimal set is found by a learning algorithm, short learner, $\mathcal{I} : \mathbb{D} \times \Lambda \rightarrow \Theta$, with \mathbb{D} as the space of data sets and Λ as the hyperparameter space comprising the control parameters for \mathcal{I} .³ Most learning algorithms find the best $\boldsymbol{\theta}$ by minimizing an empirical risk function $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \sum_{i=1}^n L(y^{(i)}, f(\mathbf{x}^{(i)} | \boldsymbol{\theta}))$ given a loss function $L : \mathcal{Y} \times \mathbb{R}^g \rightarrow \mathbb{R}_0^+$ and the data set \mathcal{D} . The best $\boldsymbol{\theta}$ found by the learner based on \mathcal{D} defines the trained model $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}^g$. As an example, the last paragraph of this chapter presents two machine learning algorithms: regression trees and random forests. These will be revisited in Chapter 3.

Prediction From \hat{f} , predictions for (potentially new) data points can be obtained. These data points need to be preprocessed in the same manner as the training data before predictions can be obtained. The data points can originate from the training data set \mathcal{D} , from a test data set (that was not used for training but for which the true outcomes are known) originating from $\mathbb{P}_{\mathbf{X}, Y}$, or from \mathcal{X} for which the true outcomes are unknown. Which data to use depends on what insights should be gained from the model analysis step.

Analysis The analysis step can serve different purposes. In the following, two of them are discussed: performance assessment and interpretation. Performance or quality assessment of \hat{f} requires a data set for which the true outcome values are known, such that the true and predicted values can be compared using a performance measure (e.g., the mean squared error). When using training data, we are only concerned with the quality of fit of the model. Good performance on the training data does not necessarily mean that the model also accurately predicts on data points that were not used for training. Therefore, an unseen test data set should be used to assess the predictive performance (see, e.g., Japkowicz and Shah, 2011, for an overview).

²In the ML literature, “feature” is predominantly used, but in the statistical and causal literature, “variable” or “covariate” are the standard. That is why this thesis uses the three terms interchangeably.

³Tuning methods can help to find a suitable vector of hyperparameters $\boldsymbol{\lambda} \in \Lambda$ for a given data set.

Interpretation methods can give further insights into a model. They help to identify which features are most important for deriving predictions or how features affect a given prediction. Model interpretation is important, especially in highly sensitive tasks like credit lending or selecting job candidates, where predictions can affect a human’s life. Interpretation methods can help to explain predictions and to audit a model. Compared to the performance assessment, which only returns a scalar, the output of interpretation methods and, therefore, the insights into a model can be diverse and do not follow a uniform format. For example, CFE methods – presented in Chapter 4 – return (a potential set of) close neighbors of a data point with a different prediction. In contrast, feature importance methods return an importance score per feature (Breiman, 2001a; Fisher *et al.*, 2019). Deriving these insights is often based on a given data set. The interpretation method determines whether the outcome must be known or not. For example, most CFE methods do not require knowledge of Y but only access to predictions obtained from \hat{f} .

The analysis stage can lead to adaptations of the model by restarting the training process, e.g., to improve the performance or to avert adverse or implausible predictions that were detected by interpretation methods.

Example: Regression Tree & Random Forest Tree algorithms divide the feature space into disjoint rectangular regions. The first algorithm was proposed by Belson (1959), with the classification and regression tree algorithm by Breiman *et al.* (1984) being one of the most popular variants. The following focuses on the regression tree algorithm by Breiman *et al.* (1984) and the random forest algorithm by Breiman (2001a) for $Y \in \mathbb{R}$. They are chosen because they can be adapted for HTE estimation, as shown in Chapters 3, 5 and 6. The following notation is based on Hastie *et al.* (2009).

Regression trees recursively partition a region \mathcal{N} into two disjoint regions \mathcal{N}_1 and \mathcal{N}_2 based on a split feature X_j . For a numeric split feature $X_j, j \in \{1, \dots, p\}$, a split point $t \in \mathcal{X}_j$ splits the data into two nodes $\mathcal{N}_1 = \{(\mathbf{x}, y) \in \mathcal{N} : x_j \leq t\}$ and $\mathcal{N}_2 = \{(\mathbf{x}, y) \in \mathcal{N} : x_j > t\}$. For a categorical split feature X_j , a split t divides the set of possible classes K_j into two subsets $\mathcal{N}_1 = \{(\mathbf{x}, y) \in \mathcal{N} : x_j \in k \subset K_j\}$ and $\mathcal{N}_2 = \{(\mathbf{x}, y) \in \mathcal{N} : x_j \in K_j \setminus k\}$. The best split variable and point are found based on a splitting criterion evaluated on training samples in \mathcal{D} . For regression trees, the optimal split minimizes the empirical risk function $\mathcal{R}(\mathcal{N}, j, t) = \mathcal{R}(\mathcal{N}_1) + \mathcal{R}(\mathcal{N}_2)$. A common choice for the risk’s loss function is the L_2 loss, such that

$$\mathcal{R}(\mathcal{N}) = \sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{N}} \left(y^{(i)} - \bar{y}_{\mathcal{N}}\right)^2, \quad (2.1)$$

where $\bar{y}_{\mathcal{N}}$ is the average outcome of observations in node \mathcal{N} . Splits are conducted until a stopping criterion is reached, for example, the minimum number of observations in a node or the maximum depth of a tree. The stopping criterion is one of the hyperparameters $\boldsymbol{\lambda}$ of a regression tree learning algorithm. Nodes that are not further split are called terminal nodes and are denoted as $Q_m, m \in \{1, \dots, M\}$, in the following. Predictions for a new observation \mathbf{x} are then obtained from the final model

$$f(\mathbf{x}) = \sum_{i=1}^M c_m \mathbb{I}(\mathbf{x} \in Q_m),$$

where c_m is the average Y of all training observations in Q_m . Q_m and c_m with $m \in \{1, \dots, M\}$ form the set of parameters $\boldsymbol{\theta}$ that parameterizes the hypothesis space \mathcal{H} of regression trees.

One disadvantage of regression trees is their high variance: small changes in the underlying data set can result in a very different structure. However, since they are also approximately unbiased (if grown sufficiently deep), they are also a suitable base learner for bootstrap aggregation or short bagging. Bagging helps to reduce the variance of a base learner by applying the base learner to multiple bootstrap samples (n observations that are randomly drawn from \mathcal{D} with replacement) of the training data. Predictions are obtained by averaging the obtained predictions of the base learners. Under the assumption that the bootstrap samples are identically distributed, the bias of the ensemble is similar to that of single base learners (Breiman, 1996; Hastie *et al.*, 2009).

Breiman (2001a) proposed a version of bagging with trees as a base learner, called random forest (RF). The used trees slightly differ from conventional trees: First, before each split, not all but only a given number of variables ($< p$) are considered for splitting, which should “decorrelate” the predictions of the trees such that they do not make the same errors; Second, the trees are grown relatively deep for approximate unbiasedness.

Random forests are harder to interpret than regression trees due to their complex structure. To address this challenge, Breiman (2001a) proposed a feature importance method that quantifies the importance of a feature as the increase in the model’s prediction error when the feature values are permuted. The generalization of this principle to arbitrary ML models is called permutation feature importance and is nowadays one of the most popular model interpretation methods (Fisher *et al.*, 2019; Molnar *et al.*, 2020).

3 Heterogeneous Treatment Effect Estimation with Machine Learning

In ML, the core interest lies in accurately approximating the relation between \mathcal{X} and \mathcal{Y} in the model f . These relations do not have to be causal. For example, to predict a disease, we can use symptoms as features in the model, but symptoms are not causes of a disease but effects of it, so the estimated effects of symptoms on the disease are not causal. Causal effects are of interest in many applications; for example, in medicine, causal effects help to assess whether and to what extent a treatment affects the progress of a disease. In recent years, the focus shifted from average to heterogeneous treatment effects (HTEs). HTEs reflect that a treatment’s effect direction and magnitude on Y can differ depending on other variables, such as a patient’s characteristics. An overview of how machine learning algorithms can be used to estimate HTEs is presented in this chapter. The potential outcomes framework provides the basis for HTE estimation, which is introduced in Section 3.1. Section 3.2 categorizes the ML-based approaches into four classes, while Section 3.3 inspects approaches beyond continuous outcomes.

Before diving into the framework and estimation approaches, the following example briefly illustrates the difference between causal and non-causal associations and highlights when HTEs are of importance. The example is based on the use case in the contributing article of Chapter 5.

Illustrative Example Large postpartum blood losses are a major cause of maternal morbidity, with increasing prevalence worldwide (MacDorman *et al.*, 2016). Mode of delivery W – vaginal delivery ($W = 0$) or cesarean section ($W = 1$) – might have a causal effect on the postpartum measured blood loss Y , but this was not adequately investigated so far (see Section 1.1 in the contribution of Chapter 5). For simplification, it is assumed that $Y|W$ is normally distributed (although this assumption is wrong as highlighted in Haslinger *et al.* (2020)), and a linear model is fitted $f(w) = \mathbb{E}(Y | W = w) = \mu_0 + \tau_w w$. To derive recommendations of actions regarding W it is tempting to interpret the estimate $\hat{\tau}_w$ as a causal effect and base all future decisions on that.

Whether $\hat{\tau}_w$ reflects a causal effect is doubtful, especially since the mode of delivery W is not randomly chosen but is chosen in agreement with the doctor and patient. There might exist risk factors that have a causal effect on both the blood loss Y and mode of delivery W . These variables are called confounders (Section 3.1 gives a formal introduction to confounders). They can introduce a spurious non-causal association between W and Y . Multifetal pregnancies can be a confounder: Chapter 5’s contribution showed that it increases the blood loss Y , and Loscul *et al.* (2019) showed that the rate of cesarean sections is higher for multifetal pregnancies than for singleton pregnancies. If the group with cesarean section contains more multifetal births with increased blood loss Y than the group with vaginal delivery, $\hat{\tau}_w$ contains not only the causal effect of cesarean section on blood loss Y but also some spurious correlation through the risk factor multifetal birth. We can account for the effect of multifetal birth, denoted as X , by adding X to

f (resulting in $f(w, x) = \mu_0 + \mu_x x + \tau_w w$).⁴ If we can assume that we accounted for all confounders, there are no variable measurement errors, and the model assumptions are correct, $\hat{\tau}_w$ reflects the causal effect of W on Y (McNamee, 2005).

The estimate $\hat{\tau}_w$ only provides an average for the population, but there might be heterogeneous effects where one group may benefit or be harmed more than others. To allow for heterogeneity in $\hat{\tau}_w$ based on some X , an interaction term for W and X needs to be added

$$\begin{aligned} f(w, x) &= \mu_0 + \mu_x x + \tau_w w + \tau_{xw} x w \\ &= \underbrace{\mu_0 + \mu_x x}_{:=\mu(x)} + \underbrace{(\tau_w + \tau_{xw} x)}_{:=\tau(x)} w := \mu(x) + \tau(x)w. \end{aligned} \tag{3.1}$$

Eq. (3.1) also motivates the usage of ML models for HTE estimation: Compared to the parametric linear model, ML models allow for more flexible, non-linear functions τ and μ .

3.1 Causality Concept: Potential Outcomes Framework

The potential outcomes framework (POF) is a statistical approach to causal inference. The framework was introduced by Splawa-Neyman *et al.* (1923) and was later extended and popularized by Rubin (1974). As in Chapter 2, Y denotes the outcome and \mathbf{X} are variables, more specifically pre-treatment variables that are observed before a treatment is administered (e.g., a patient’s characteristics like age, sex, or disease status). W denotes a treatment variable whose causal effect on the outcome is of interest. This thesis focuses primarily on a binary $W = \{0, 1\}$, where $W = 0$ corresponds to the control treatment (no/placebo/standard treatment) and $W = 1$ corresponds to (a potentially new) treatment. Section 2.3 in the contribution of Chapter 5 discusses extensions to multiple treatments.

The POF assumes that each unit has two potential outcomes $Y(w), w \in \{0, 1\}$ under each treatment arm. The POF was introduced for $Y \in \mathbb{R}$ and we focus on this case throughout Sections 3.1 and 3.2. Extensions to other types of outcomes are discussed in Section 3.3.

3.1.1 Causal Estimand

For $Y \in \mathbb{R}$, an individual treatment effect τ for an observation \mathbf{x} can be defined as the difference between its two potential outcomes $\tau := Y(1) - Y(0)$. Unfortunately, it is, in most cases, not possible to observe both potential outcomes for an individual but only one.⁵ This problem is called the fundamental problem of causal inference (Holland, 1986). If a data set $\mathcal{D} = (\mathbf{x}^{(i)}, w^{(i)}, y^{(i)})_{i=1}^n$ is available, we might be able to approximate the individual treatment effects by averaging the outcomes of instances i that are similar to \mathbf{x} . This causal estimand is then the conditional average treatment effect (CATE)

$$\tau(\mathbf{x}) := \mathbb{E}(Y(1) - Y(0) | \mathbf{X} = \mathbf{x}). \tag{3.2}$$

⁴Other strategies are matching methods or inverse propensity score weighting (see Hernán and Robins, 2020).

⁵Observing both outcomes is only possible under strong invariance assumptions, e.g., that $Y(w)$ measured at an earlier time point is the same as the value $Y(w)$ measured at a later time point, for $\forall w \in \{0, 1\}$ (Holland, 1986).

3.1.2 Statistical Estimand

For mapping the causal estimand of Eq. (3.2), which still contains both potential outcomes, to statistical quantities, four identifying assumptions must hold.

Identifying Assumptions

The following assumptions are based on early work by Rubin and Rosenbaum (Rubin, 1974, 1980; Rosenbaum and Rubin, 1983). A detailed summary is given in Hernán and Robins (2020).

Assumption 1. *Conditional Exchangeability/Unconfoundedness*

*The treatment assignment is independent of the **potential** outcomes given \mathbf{X} , such that*

$$Y(1), Y(0) \perp\!\!\!\perp W \mid \mathbf{X}.$$

This means that, within levels of \mathbf{X} , the group receiving the treatment and the group receiving the control do not differ in the characteristics that affect the *potential* outcomes. (The minimum set of) variables \mathbf{X} required for the fulfillment of Assumption 1 are called confounders (VanderWeele and Shpitser, 2013). If not all confounders are observed, Assumption 1 is not fulfilled. Figure 3.1a provides an illustration of why conditioning on confounders is required based on causal graphs. Causal graphs consist of nodes or vertices reflecting variables and arrows that connect them. Arrows from one node to another reflect a direct causal effect of the former to the latter. In Figure 3.1a, W has a causal effect on Y but also a non-causal effect resulting from an open “backdoor-path” over the confounder X . By conditioning on X , we can “block” this path such that there is only causal association.

Assumption 2. *Positivity*

It holds for all values $\mathbf{X} = \mathbf{x}$ with $\mathbb{P}(\mathbf{X} = \mathbf{x}) > 0$ in the population of interest that

$$0 < \pi(\mathbf{x}) := \mathbb{P}(W = 1 \mid \mathbf{X} = \mathbf{x}) < 1,$$

with $\pi(\mathbf{x})$ as the propensity score. This means that assignment to one of the treatment groups is never deterministic.

Assumption 3. *No Interference*

The potential outcome $Y^{(i)}$ of one observation i does not depend on other individuals’ treatment, i.e., $Y^{(i)}(w^{(1)}, \dots, w^{(i)}, \dots, w^{(n)}) = Y^{(i)}(w^{(i)})$.

Assumption 4. *Consistency*

If, for a given observation \mathbf{x} , the treatment is w , then the observed Y is equal to the potential outcome under treatment, such that $Y = Y(w)$.

Assumption 4 assumes that there are not multiple hidden versions of the treatment $W = 1$ and “no matter how unit \mathbf{x} received treatment 1, the outcome that would be observed would be $Y(1)$ ” (Rubin, 2005, p. 323). Many research papers, including the contributions of Chapters 5 and 6, do not explicitly state Assumptions 3 and 4 under the argument that the definition of potential outcomes presupposes them (VanderWeele and Hernán, 2013). The following subsection discusses the plausibility of Assumptions 1 and 2, also assuming that Assumptions 3 and 4 are fulfilled.

Randomized Trial vs. Observational Study

Whether Assumptions 1 and 2 are plausible for a given use case depends on the data collection process or study type.

In randomized trials, the assignment process to one of the treatment arms is randomized and, therefore, W is binomially distributed with constant propensity scores $\pi := \mathbb{P}(W = 1)$, such that $W \sim B(\pi)$. Since $W \perp \{X, Y(0), Y(1)\}$, Assumption 1 is fulfilled. If $0 < \pi < 1$, also Assumption 2 is naturally fulfilled, since $\mathbb{P}(W = 1 \mid \mathbf{X} = \mathbf{x}) = \pi \in (0, 1)$. Although randomized trials are seen as the gold standard to answer causal questions (Hariton and Locascio, 2018), it is not always possible to conduct them due to ethical, time, or monetary budget issues. Furthermore, they can have limitations; for example, the trial sample might not represent the target population because of the limited sample size and the recruitment process (Deaton and Cartwright, 2018). In addition, the trial is conducted in a controlled setting (Cook and Thigpen, 2019). Overall, the question remains whether conclusions from the trial can be transferred to the real world.

Observational studies, on the other hand, infer information about a population from a sample in which the treatment group assignment is not under the control of the researcher. The advantages are that data is readily available, with lower costs, and in larger quantities, such that the target population might be better represented (Colnet *et al.*, 2023). The disadvantage is that there is the risk that Assumption 1 is not fulfilled. Since the treatment assignment is not necessarily randomized, confounders can exist. If we can assume that all confounders were measured, Assumption 1 would be fulfilled. However, the absence of unmeasured confounders cannot be guaranteed or proven (Rubin, 1974). To diminish the risk of unmeasured confounders, more variables might be included in the analysis, but then Assumption 2 might not be satisfied anymore, due to the high dimensionality of \mathbf{X} and the related curse of dimensionality (D’Amour *et al.*, 2021).⁶

Identification

If we can assume that the above assumptions hold, we can reduce the causal estimand of Eq. (3.2) to statistical quantities

$$\begin{aligned}
 \tau(\mathbf{x}) &= \mathbb{E}(Y(1) - Y(0) \mid \mathbf{X} = \mathbf{x}) = \mathbb{E}(Y(1) \mid \mathbf{X} = \mathbf{x}) - \mathbb{E}(Y(0) \mid \mathbf{X} = \mathbf{x}) \\
 &\stackrel{\text{A.1\&2}}{=} \mathbb{E}(Y(1) \mid \mathbf{X} = \mathbf{x}, W = 1) - \mathbb{E}(Y(0) \mid \mathbf{X} = \mathbf{x}, W = 0) \\
 &\stackrel{\text{A.4}}{=} \underbrace{\mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}, W = 1)}_{:=\eta_1(\mathbf{x})} - \underbrace{\mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}, W = 0)}_{:=\eta_0(\mathbf{x})} = \eta_1(\mathbf{x}) - \eta_0(\mathbf{x}).
 \end{aligned} \tag{3.3}$$

The next section discusses ML approaches to derive a function $\tau : \mathcal{X} \rightarrow \mathbb{R}$, which estimates the CATE $\tau(\mathbf{x})$ for observations \mathbf{x} .

⁶Caution is also required to not include variables that are not confounders but mediators or colliders. See Cinelli *et al.* (2022) for an introduction to the topic.

3.2 Estimation via Machine Learning Approaches

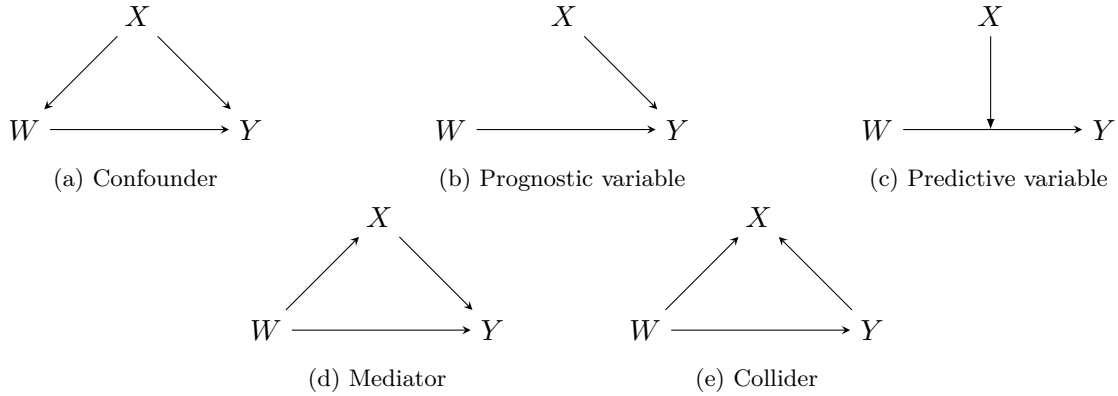


Figure 3.1: Different roles of a variable/feature X depending on the causal structure.

3.2 Estimation via Machine Learning Approaches

In previous years, multiple ML approaches have been proposed for estimating HTEs (Knaus *et al.*, 2020; Künzel *et al.*, 2019). Compared to classical statistical approaches (such as the linear model of Eq. (3.1)), ML approaches are based on weaker assumptions. This allows for a (close to) non-parametric estimation of the relationship between \mathbf{X} , Y , and W , and, therefore, more flexible structures for deriving HTEs. Another advantage is that many ML algorithms automatically identify (higher-order) interaction effects between features. Since heterogeneity in the treatment effect arises from the interaction between the treatment W and variables \mathbf{X} (as illustrated in the initial example of this chapter), this property is beneficial for HTE estimation.

Before an overview of the approaches is presented, ML (as introduced in Chapter 2) and causal inference based on the POF (as introduced in Section 3.1) are set into relation.

1. Target: Instead of deriving a model $f : \mathcal{X} \rightarrow \mathbb{R}^g$, we are now interested in a model $\tau : \mathcal{X} \rightarrow \mathbb{R}$ to accurately predict the causal effect of W on Y .
2. Role of W : Due to the focus on the treatment effect, treatment variable W has a special role compared to the other variables \mathbf{X} .
3. Roles of \mathbf{X} : \mathbf{X} are not simply features but can have different roles depending on their causal relations. Figures 3.1a to 3.1c provide visual examples using causal graphs. Confounders were already defined in Section 3.1.2. Features \mathbf{X} that affect the treatment effect are called predictive variables, and features \mathbf{X} that affect Y are called prognostic. For HTE estimation, the features \mathbf{X} are *pre-treatment* variables that are not affected by W . Figures 3.1d and 3.1e show a mediator and collider as counterexamples, which are influenced by W or by W and Y , respectively.
4. Assumptions: In order to estimate causal effects, strong (and mostly untestable) assumptions are required (Section 3.1.2), which is not the case for ordinary ML tasks.
5. Ground truth: Due to the fundamental problem of causal inference (Section 3.1.1), true treatment effects are not observable for real-world use cases, while outcomes Y can be observed. Treatment effects are only observable if we know the data-generating process (e.g., in simulation studies) or under very strong invariance assumptions (see Footnote 5).

To aim for HTEs, the model training step (step 1 of the ML workflow in Chapter 2) must be adapted. Over the past years, multiple ML-based estimators have been proposed. They can be divided into model-agnostic (Section 3.2.1) and model-specific approaches (Section 3.2.2) (Curth and van der Schaar, 2021).

3.2.1 Model-Agnostic Estimators

Model-agnostic estimators use ML algorithms off-the-shelf without any adaptations such that the ML algorithm can be easily replaced. Curth and van der Schaar (2021) and Crabbé *et al.* (2022) further divide model-agnostic estimators into two subclasses: indirect and direct estimators.

Model-agnostic Indirect Estimators

Model-agnostic indirect estimators are inspired by Eq. (3.3). First, ML algorithms learn the expected outcome functions $\eta_1(\mathbf{x}) = \mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}, W = 1)$ and $\eta_0(\mathbf{x}) = \mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}, W = 0)$. Then, the treatment effect of a new data point \mathbf{x} is equal to $\hat{\tau}(\mathbf{x}) = \hat{\eta}_1(\mathbf{x}) - \hat{\eta}_0(\mathbf{x})$. Two popular members of this class are the T-learner and S-learner proposed by Künzel *et al.* (2019).

For the T-learner, *two* ML algorithms learn η_1 and η_0 separately. For η_1 , $\mathbb{E}(Y \mid \mathbf{X} = \mathbf{x})$ is estimated by using only the treated individuals in the training data \mathcal{D} , for η_0 , $\mathbb{E}(Y \mid \mathbf{X} = \mathbf{x})$ is estimated by using only the individuals in \mathcal{D} of the control group. Since the two ML models do not share any information, the T-learner is especially suitable if no common patterns appear in η_0 and η_1 (Künzel *et al.*, 2019).

For the S-learner, only a *single* ML model is fitted. The expected outcome function $\eta(\mathbf{x}, w) := \mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}, W = w)$ is estimated by treating W as another feature in addition to \mathbf{X} . By defining $\hat{\eta}_w(\mathbf{x}) := \hat{\eta}(\mathbf{x}, w)$, the S-learner estimates $\tau(\mathbf{x})$ as defined in Eq. (3.3). If algorithms conduct feature selection like RFs, the treatment assignment can also be ignored, which is beneficial if the CATE is 0 (Künzel *et al.*, 2019).

Prominent choices for ML algorithms for S- and T-learners are neural networks (Curth and van der Schaar, 2021), RFs (Nie and Wager, 2020; Künzel *et al.*, 2019; Foster *et al.*, 2011), or Bayesian additive regression trees (Künzel *et al.*, 2019).

Model-agnostic Direct Estimators

Model-agnostic direct estimators are approaches that use ML algorithms off-the-shelf to estimate treatment effects $\tau(\mathbf{x})$ directly. Since knowledge of the true treatment effect is not available, these approaches transform the outcomes to pseudo-outcomes \tilde{Y} for which $\mathbb{E}(\tilde{Y} \mid \mathbf{X} = \mathbf{x}) = \tau(\mathbf{x})$ holds (Curth and van der Schaar, 2021). The derivation of pseudo-outcomes can be seen as a preprocessing step within the training step of the ML workflow (Chapter 2). Different transformation approaches exist (Curth and van der Schaar, 2021). As an example, the doubly robust (DR-) learner by Kennedy (2022) is briefly presented.

The DR-learner of Kennedy (2022) is based on the doubly robust augmented inverse propensity weighting estimator by Robins and Rotnitzky (1995). First, the propensity score and expected outcome functions π , η_1 , and η_0 are estimated from the training data (e.g., by an ML algorithm).

3.2 Estimation via Machine Learning Approaches

The outcomes $y^{(i)}$ of the training observations $i \in \{1, \dots, n\}$ are then transformed to reflect treatment effects given $\mathbf{x}^{(i)}$ and $w^{(i)}$

$$\tilde{y}^{(i)} = \frac{w^{(i)} - \hat{\pi}(\mathbf{x}^{(i)})}{\hat{\pi}(\mathbf{x}^{(i)})(1 - \hat{\pi}(\mathbf{x}^{(i)}))} \left(Y - \hat{\eta}_{w^{(i)}}(\mathbf{x}^{(i)}) \right) + \hat{\eta}_1(\mathbf{x}^{(i)}) - \hat{\eta}_0(\mathbf{x}^{(i)}).$$

An ML model $f(\mathbf{x}) = \mathbb{E}(\tilde{Y} \mid \mathbf{X} = \mathbf{x})$ is then fitted to the transformed data. The method is called doubly robust because it requires the correct specification of either the propensity score function π or the expected outcome functions η_1 and η_0 to be unbiased w.r.t. τ (Kennedy, 2022).

3.2.2 Model-Specific Estimators

Model-specific estimators rely on a specific, potentially adapted ML algorithm to derive treatment effects $\tau(\mathbf{x})$. Replacing the ML algorithm is not easily possible compared to model-agnostic approaches. The following subsections focus on adaptations to RFs, which were introduced in Chapter 2. They play a crucial role in the contributing articles of Chapters 5 and 6. Adaptions to other ML approaches have also been proposed, e.g., to neural networks (Shalit *et al.*, 2017) or boosting models (Powers *et al.*, 2018). Like model-agnostic approaches, RF-based approaches can be distinguished into indirect and direct estimators.

Model-specific Indirect Estimators

Model-specific indirect estimators apply specific ML algorithms to estimate $\eta_1(\mathbf{x})$ and $\eta_0(\mathbf{x})$. The difference between $\eta_1(\mathbf{x})$ and $\eta_0(\mathbf{x})$ defines the treatment effect $\tau(\mathbf{x})$. As an example, the bivariate imputation (BI) approach by Lu *et al.* (2018) is presented.

The BI approach assumes the existence of bivariate outcomes (Y_1, Y_0) , one for each treatment arm. Due to the fundamental problem of causal inference, only one of the outcomes $y_w^{(i)}, w \in \{0, 1\}$ can be observed for each observation i in \mathcal{D} . The other is treated as missing and needs to be imputed. In the first iteration, a bivariate RF is grown given only the observed outcomes. Compared to ordinary RFs, bivariate RFs consider both outcomes (under $W = 0$ and under $W = 1$) for splitting. The risk function is updated from Eq. (2.1) to

$$\mathcal{R}(\mathcal{N}) = \sum_{w=0}^1 \left\{ \sum_{(\mathbf{x}^{(i)}, w^{(i)}, y_1^{(i)}, y_0^{(i)}) \in \mathcal{N}} \mathbb{I}_{w^{(i)}=w} \left(y_w^{(i)} - \bar{y}_{w,\mathcal{N}} \right)^2 \right\}, \quad (3.4)$$

with $\bar{y}_{w,\mathcal{N}}$ as the average outcome under $W = w$ of observations in node \mathcal{N} . After fitting the forest, the mean terminal node values of Y_1 and Y_0 replace the missing $y_w^{(i)}, w \in \{0, 1\}$. The complete data set is then the input to another bivariate RF, which again updates the missing outcomes ($\mathbb{I}_{w^{(i)}=w}$ is removed from Eq. (3.4)). This process is repeated a fixed number of times. In the simulation study by Lu *et al.* (2018), the BI approach did not perform better than the model-agnostic approaches with RFs.

Model-specific Direct Estimators

Model-specific direct estimators adapt specific ML algorithms to focus on the *direct* estimation of $\tau(\mathbf{x})$. The following paragraphs present how Seibold *et al.* (2018) and Athey *et al.* (2019) adapted the RF algorithm for model-based forests (MOBs) and causal forests (CFs), respectively. Both approaches derive the HTEs in a model-driven way based on the additive interaction model

$$(Y \mid \mathbf{X} = \mathbf{x}) = \mu(\mathbf{x}) + \tau(\mathbf{x})W + \sigma Z, \quad (3.5)$$

where σZ is the error term with $\mathbb{E}(Z \mid \mathbf{X}, W) = 0$ and standard deviation $\sigma > 0$. Besides the treatment effect $\tau(\mathbf{x})$, the equation includes $\mu(\mathbf{x})$, the effect of prognostic variables \mathbf{X} on Y . We already saw a similar model in Eq. (3.1) but with a linear $\mu(\mathbf{x})$ and $\tau(\mathbf{x})$.

Model-based Forest MOBs are based on the model-based recursive partitioning algorithm by Hothorn *et al.* (2006) and Zeileis *et al.* (2008) – a general framework combining parametric models with an (unbiased) tree algorithm. Seibold *et al.* (2016, 2018) applied the general framework to estimate HTEs. The following paragraph focuses on MOBs differences to regression trees and RFs within MOBs’ application as HTE estimators.

First, MOBs attach parametric models to the nodes of a tree instead of constant estimates. In each node \mathcal{N} , the following base model is fitted based on Eq. (3.5)

$$\mathbb{E}(Y \mid W = w) = \mu + \tau w \quad (3.6)$$

using ordinary least squares, i.e., by minimizing the negative log-likelihood/ L_2 loss

$$(\hat{\mu}, \hat{\tau})^T = \arg \min_{\mu, \tau} \sum_{(\mathbf{x}^{(i)}, w^{(i)}, y^{(i)}) \in \mathcal{N}} \underbrace{\frac{1}{2} (y^{(i)} - \mu - \tau w^{(i)})^2}_{:=l_i(\mu, \tau)}.$$

Second, the splitting criterion detects parameter instabilities instead of outcome instabilities by focusing on the model scores (partial derivatives of the log-likelihood) $s(\hat{\mu}, \hat{\tau}) = (Y - \hat{\mu} - \hat{\tau}w)(1, w)^T$, given $\hat{\mu}$ and $\hat{\tau}$ which were estimated in node \mathcal{N} .

Third, the best split variable and best split point are selected in two separate steps. This averts a potential variable selection bias due to variables with many split points (Zeileis *et al.*, 2008). The split variable is the variable $X_j, j \in \{1, \dots, p\}$ with the lowest p-value for a permutation test that tests for independence between the model scores $s(\hat{\mu}, \hat{\tau})$ and X_j . The split point is the value that results in the largest discrepancy between the score functions (see Appendix 2 of Seibold *et al.*, 2018, for details).

Fourth, predictions $\tau(\mathbf{x})$ for a new \mathbf{x} are not obtained by averaging but by local maximum likelihood aggregation. The aggregation requires weights for each training sample $\mathbf{x}^{(i)}$ that reflect how similar $\mathbf{x}^{(i)}$ is to \mathbf{x} w.r.t. to τ . These weights $\alpha_i(\mathbf{x})$ are derived from the MOB by measuring how often a sample $\mathbf{x}^{(i)}$ falls in the same leaf as \mathbf{x} . The reweighted training samples are the basis for estimating $\mu(\mathbf{x})$ and $\tau(\mathbf{x})$ by solving

$$(\hat{\mu}(\mathbf{x}), \hat{\tau}(\mathbf{x}))^T = \arg \min_{\mu, \tau} \sum_{i=1}^n \alpha_i(\mathbf{x}) l_i(\mu, \tau).$$

3.2 Estimation via Machine Learning Approaches

Causal Forest Athey *et al.* (2019) proposed CFs as a special case of their framework on generalized RFs, which estimates any quantity of interest that can be identified via a local moment equation. The local moment equation for HTEs is derived from the additive interaction model of Eq. (3.5). The fact that this is also the basis for MOBs was the starting point for an in-depth theoretical and empirical comparison of MOBs and CFs summarized in the contributing article of Chapter 5. The following introduces CFs by briefly describing their differences to MOBs.

The *first* difference is that Athey *et al.* (2019) transform Eq. (3.5) based on the orthogonalization strategy of Robinson (1988). They artificially add a 0 ($m(\mathbf{x}) - m(\mathbf{x})$) such that

$$\begin{aligned} (Y \mid \mathbf{X} = \mathbf{x}) &= m(\mathbf{x}) - m(\mathbf{x}) + \mu(\mathbf{x}) + \tau(\mathbf{x})W + \sigma Z \\ &= m(\mathbf{x}) + \tau(\mathbf{x})(W - \pi(\mathbf{x})) + \sigma Z \end{aligned}$$

using the conditional mean function $m(\mathbf{x}) := \mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}) = \mu(\mathbf{x}) + \tau(\mathbf{x})\pi(\mathbf{x})$. This reformulation motivates a two-step approach: First, the nuisance parameters $\pi(\mathbf{x})$ and $m(\mathbf{x})$ are estimated, then, $\tau(\mathbf{x})$ is estimated using CFs with $\mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}, W = w) = \hat{m}(\mathbf{x}) + \tau(w - \hat{\pi}(\mathbf{x}))$ as the base model in each node \mathcal{N} . The corresponding minimization problem is then

$$\hat{\tau} = \arg \min_{\mu, \tau} \sum_{(\mathbf{x}^{(i)}, w^{(i)}, y^{(i)}) \in \mathcal{N}} \underbrace{\frac{1}{2} \left(y^{(i)} - \hat{m}(\mathbf{x}^{(i)}) - \tau(w^{(i)} - \hat{\pi}(\mathbf{x}^{(i)})) \right)^2}_{:=l_i(\tau)}.$$

The idea behind orthogonalization is that the effects of \mathbf{X} on W and Y are “regressed out”. Athey *et al.* (2019) show that this leads to better performance in case of confounders. Compared to MOBs, CFs only focus on identifying heterogeneity in $\tau(\mathbf{x})$ and not in $\mu(\mathbf{x})$.

The *second* difference to MOBs is the splitting procedure. Like RFs, CFs do not separate the split variable and split point selection but search for the best split point among all split points of all considered features. To reduce the computational burden, Athey *et al.* (2019) use an efficient splitting procedure based on Wright and Ziegler (2017) that makes the reestimation of $\hat{\tau}$ in each potential child node obsolete. Details are given in Appendix A of the contributing article of Chapter 5. Predictions are obtained by local maximum likelihood estimation similar to MOBs.

The above and (in more detail) the contribution of Chapter 5 show that MOBs and CFs share the same *theoretical* grounds for $Y \in \mathbb{R}$ for an additive model under the L_2 loss. This allows for constructing hybrid approaches that blend CFs and MOBs to inspect which *computational* elements of the two approaches are beneficial for HTE estimation. Based on a simulation study, the contribution in Chapter 5 identifies the orthogonalization of W in CFs and the splitting based on heterogeneity in $\tau(\mathbf{x})$ and $\mu(\mathbf{x})$ in MOBs as the main drivers for good performance, especially in case of confounders.

Overall, this section presented four different classes of ML-based HTE estimators. Table 3.1 provides a short summary.

Table 3.1: Overview of the four classes of ML-based HTE estimators. The distinction is based on whether $\tau(\mathbf{x})$ are estimated indirectly or directly and whether the underlying ML algorithms are interchangeable.

	Model-agnostic (Sec. 3.2.1)	Model-specific (Sec. 3.2.2)
Indirect	T-learner, S-learner	BI approach
Direct	DR-learner	MOB, CF

3.3 Beyond Continuous Outcomes

The last section focused on outcomes $Y \in \mathbb{R}$, but in many application fields more complex outcome types are present. The contributing articles of Chapters 5 and 6 present examples from the medical context:

1. Assessment of the mode of delivery on postpartum blood loss is not as simple as described in the introduction to this chapter. Extreme blood losses are rare (left-skewed), and the measurement process is potentially inaccurate (interval-censored) (Chapter 5).
2. Assessment of the effect of a drug on the course of amyotrophic lateral sclerosis is based on scores of ordinal ability tests or the survival times of patients (Chapter 6).

Research on ML algorithms for HTE estimation beyond continuous outcomes has primarily focused on binary and (right-censored) survival data. For binary outcomes $Y \in \{0, 1\}$, conditional average treatment effects can still be estimated with the above methods. Estimates $\hat{\tau}(\mathbf{x})$ are interpreted as absolute risk differences $\tau(\mathbf{x}) = \mathbb{E}(Y(1) - Y(0) \mid \mathbf{X} = \mathbf{x}) = \mathbb{P}(Y(1) \mid \mathbf{X} = \mathbf{x}) - \mathbb{P}(Y(0) \mid \mathbf{X} = \mathbf{x})$. For right-censored survival outcomes, Hu *et al.* (2021) inspected an extension of T-learners: First, ML algorithms for survival analysis (like random survival forests (Ishwaran *et al.*, 2008)) estimate survival or hazard functions independently for each treatment group. The difference in the median survival time defines the HTE. Hu *et al.* (2021) compared this model-agnostic approach to a model-specific approach – the adapted BART algorithm by Henderson *et al.* (2018) – and found that the latter produces more reliable estimates. Cui *et al.* (2023) extended the CF algorithm of Athey *et al.* (2019) to right-censored survival outcomes by adapting the underlying loss function to focus on the difference in restricted mean survival times.

Because MOBs combine the parametric modeling framework with RFs, they offer the flexibility to estimate HTEs for various outcome types. The only requirement is that the outcomes can be well described by parametric models. The loss function of Eq. (3.6) then changes to the negative log-likelihood. The contributing article of Chapter 6 presents a holistic view of this approach, covering generalized linear models and transformation models. Constructing the tree and obtaining predictions is in essence the same as for the MOBs described in Section 3.2.2, but the interpretation of the treatment effect is less straightforward. HTEs are expressed by statistical quantities, e.g., log-odds ratios in binary logistic regression models, multiplicative mean effects in a Poisson model, or log-hazard ratios for Weibull proportional hazards models. Complex models require a careful assessment, and several papers worked out the details for different outcome classes and models (Seibold *et al.*, 2016, 2018; Korepanova *et al.*, 2020; Buri and Hothorn, 2020; Fokkema *et al.*, 2018; Hothorn and Zeileis, 2021).

While these papers focused on estimating HTEs for randomized trials, the contributions of Chapters 5 and 6 investigated the performance of MOBs in the case of confounders. Simulation studies showed that confounders affect the estimation of HTEs based on MOBs without adaptations. In the manuscripts, new variants of MOBs are proposed based on the orthogonalization/two-step approach of CFs. They can improve the performance of MOBs in case of confounders, for different types of outcomes, as shown in simulation studies.

4 Model Interpretation with Counterfactual and Semi-factual Explanations

As seen in Chapter 2, interpretation methods are a valuable tool for model analysis – the last step of the ML workflow. They complement performance assessment by providing further insights into a model. The research field that addresses the interpretability of ML models is called interpretable machine learning. It comprises research on methods to interpret ML models post-hoc and research on inherently interpretable (high-performant) ML models (Carvalho *et al.*, 2019). This chapter focuses on the former and presents two post-hoc interpretation methods: Counterfactual explanations (CFEs) and semi-factual explanations (SFEs). CFEs and SFEs are *local* interpretation methods because they aim to explain only the model behavior for a single observation (and its close surroundings) (Doshi-Velez and Kim, 2017).⁷ CFEs and SFEs give insights into a prediction by presenting alternative data points. For CFEs, these points describe *minimal* changes in a few features required for changing a prediction, while semi-factual explanations describe *maximal* changes in a few features required for *not* changing a prediction.

A denied credit application serves as a motivating example. A possible CFE could be “If the applicant had applied for a credit of € 2000 instead of € 4000, the application would have been classified as being of low risk (instead of high risk)”, while an SFE could be “Even if the applicant had applied for a credit of € 3000, the application would still be classified as being of high risk”. Table 4.1 summarizes what insights can be obtained from CFEs and SFEs.

Table 4.1: Overview of the insights CFEs and SFEs can offer. The first column specifies the purpose, the last two columns provide more details and an example.

explain	CFE	Details: explain why the current and not a different prediction was reached	Example: “these feature changes would result in a different prediction, they affect the prediction”
	SFE	Details: justify why the current prediction was reached	Example: “these feature changes would <i>not</i> change the prediction, they do not affect the prediction”
audit	CFE	Details: detect adverse predictions that should <i>not</i> change	Example: “these feature changes should <i>not</i> make a difference in prediction”
	SFE	Details: detect adverse predictions that should change	Example: “these feature changes should make a difference in prediction”
advise	CFE	Details: identify actions to reach the desired prediction in the future	Example: “these feature changes help to change the prediction in the future”
	SFE	Details: identify actions that <i>do not</i> help to reach a different prediction in the future	Example: “these feature changes <i>do not</i> help to change the prediction in the future”

Since the insights into a model provided by CFEs and SFEs differ, CFEs and SFEs should not be applied in an either-or-manner but complementary (the lack of one-fits-all interpretability is also highlighted in Section 1 of the contributing article of Chapter 7). The following section introduces

⁷In contrast, global methods aim to explain the model behavior in general, considering the whole feature space.

the causal concept of counterfactuals underlying CFEs and SFEs. Sections 4.2 and 4.3 formalize CFEs and SFEs: their definitions, desired properties, and generation methods.

4.1 Causality Concept: Counterfactuals

In general, the core question when trying to find explanations for a situation is “why did it happen?”. For answers, humans try to identify the causes of it. Hume (1748) and later Lewis (1973) promoted to rephrase “W has caused Y” to “If W had not been the case, Y would not have occurred”, defined as counterfactual reasoning. Counterfactuals are, therefore, a central part of causality. A rejected credit application serves as an illustrative example. A counterfactual reason can be: “If you owned a house, your application would not have been rejected“. The statement tells us that property ownership influences whether a credit is granted or not. Reasoning based on counterfactuals is beneficial because it is intrinsically grounded in us humans. After all, “we think of a cause as something that makes a difference, and the difference it makes must be a difference from what would have happened without it” (Lewis, 1973, p. 557).

We can define counterfactuals under the potential outcomes framework (Section 3.1). If we observe Y under $W = w$, the counterfactual is $Y(W = w')$, i.e., the outcome Y under a different value w' . We already saw in Section 3.1 that, in general, we cannot observe $Y(W = w')$ and must rely on strong assumptions to estimate it. Pearl *et al.* (2016) define counterfactuals slightly different as the expected Y under $W = w'$, given $W = w$ and $Y = y$

$$\mathbb{E}(Y(W = w') \mid W = w, Y = y). \quad (4.1)$$

Conditioning on the observed values of W and Y is required because, from these values, we can obtain unobserved background information. Pearl *et al.* (2016) present a three-steps approach to estimate Eq. (4.1). This approach relies on the knowledge of a structural causal model (SCM), a set of equations that represents the causal relationship between variables. SCMs induce or can be translated into causal graphs. The following presents an SCM M for the causal graph in Figure 4.1. W is the variable of interest, Y the outcome, and W and X are causes of Y .



Figure 4.1: Causal graph

$\mathbf{U} := (U_W, U_X, U_Y)$ in the SCM denotes a set of exogenous, unobserved random variables that define noise or background conditions of the variables. The three-step approach by Pearl only requires the knowledge of f_Y . Given an observation (y, w, x) , we can compute counterfactual outcomes $Y(W = w')$ by

1. Abduction: Use (y, w, x) to determine the value of U_Y .⁸

⁸If U_Y cannot be determined, it is possible to base the computation on the knowledge of probabilities $P(U_Y = u)$ (see Section 4.2.4 in Pearl *et al.*, 2016).

4.2 Counterfactual Explanations

2. Action: Modify the model M by replacing the structural equations for W with $W = w'$.
3. Prediction: Use the derived U_Y from step 1 and the modified model from step 2 to compute the counterfactual outcome $Y(W = w')$.

The approach requires a *parametric* model f_Y because only then the value of U_Y can be derived. For further details, readers are referred to Section 4 of Pearl *et al.* (2016).

4.2 Counterfactual Explanations

Wachter *et al.* (2018) introduced counterfactuals as a method for ML model interpretation, called counterfactual *explanations* (CFEs). They define CFEs as statements of the form (p. 848): “Score p was returned because variables V had values (v_1, v_2, \dots) associated with them. If V instead had values (v'_1, v'_2, \dots) , and all other variables had remained constant, score p' would have been returned.” The following formalizes the definition of Wachter *et al.*’s CFEs and embeds it in the ML terminology of Chapter 2. In accordance with Wachter *et al.* (2018) and the contributions in Chapters 8 and 10, the definition only considers models $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$.⁹

Definition 1 (Counterfactual explanation). *Given the prediction function $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$, an observation of interest \mathbf{x}^* and a set or interval of desired predictions $Y' \subset \mathbb{R}$ with $\hat{f}(\mathbf{x}^*) \notin Y'$, a point $\mathbf{x} \in \mathcal{X}$ is a CFE for \mathbf{x}^* , if it is most similar to \mathbf{x}^* while $\hat{f}(\mathbf{x}) \in Y'$.*

Wachter *et al.* (2018) note that “[their] version of CFEs perhaps most resembles a structural equations approach in execution by identifying alterations to variables” (p. 848) – the notion of causal counterfactuals given in Pearl *et al.* (2016). Causal counterfactuals and CFEs reason about similar, alternative worlds (in which a few features changed).¹⁰ They also differ in many aspects: While causal counterfactuals aim to inspect the data-generating process by investigating whether a predefined change in a feature results in a change in Y (denoting a causal effect), CFEs aim to inspect the model by investigating what minimal feature changes are required for Y to change to a predefined Y' . Another difference is that CFEs do not necessarily require causal knowledge (Wachter *et al.*, 2018) (however, a few methods utilize it to derive more realistic CFEs; see the next subsections). Furthermore, Rubin and Pearl introduced their methods to derive counterfactual outcomes with a single feature – often under the consideration that this feature is binary. CFEs are not restricted to single feature changes; multiple $X_j, j \in \{1, \dots, p\}$ can be changed simultaneously. To restrict the number of potential feature changes, desired properties of CFEs should be formalized based on their anticipated purposes in Table 4.1.

4.2.1 Desired Properties

In the following, six desired properties are presented, where the first three were already part of Definition 1. They reflect that CFEs should have predictions equal to the desired prediction Y' and that CFEs should be similar to the instance of interest \mathbf{x}^* .

⁹This naturally covers regression models. For classification models, it is assumed that the score or probability for a predefined class of interest is returned by \hat{f} .

¹⁰Contrasting two alternative worlds is essential to human cognition (Byrne, 2002). Therefore, CFEs are often referred to as explanations for laypersons, which can assist in the implementation of the GDPR’s “right to explanation” (European Parliament and Council of the European Union, 2016; Wachter *et al.*, 2018).

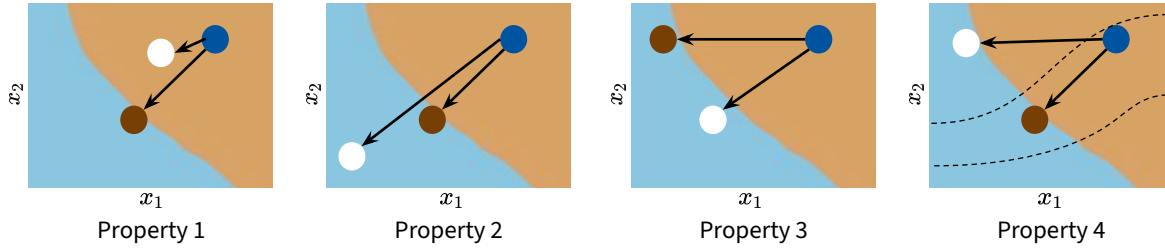


Figure 4.2: Illustration of Properties 1 to 4 for a binary classification data set with two features. The background color reflects the two classes (blue vs. brown). The observation to explain is the blue dot. In all four subfigures, the brown dot is preferred over the white dot based on the respective property. For the fourth property, the area between the two dashed lines reflects the data manifold.

Property 1 (Validity). \mathbf{x} should have a prediction $\hat{f}(\mathbf{x}) \in Y'$.

Property 2 (Proximity). \mathbf{x} should be close to \mathbf{x}^* .

Property 3 (Sparsity). \mathbf{x} should only differ from \mathbf{x}^* in a few features.

The next three properties are based on Verma *et al.* (2022) and Definition 1 of the contributing articles of Chapters 8 and 10. They reflect that CFEs should be realistic and consider feature dependencies, causal dependencies, or actionability constraints. This is particularly relevant if CFEs should recommend actions for changing the prediction in the future, denoted in the literature as algorithmic recourse (Karimi *et al.*, 2021).

Property 4 (Plausibility). \mathbf{x} should be realistic, *i.e.*, close to the data manifold, such that feature dependencies are taken into account.

Property 5 (Causality). \mathbf{x} reflects the underlying causal structure and considers causal relations of features.

Property 6 (Actionability). \mathbf{x} should not alter immutable features (*e.g.*, country of birth).

Figure 4.2 illustrates the first four properties in a simple example for a binary classification data set with two features. The last two properties are omitted because they are based on user input and domain knowledge.

4.2.2 Generation Methods

Over the past years, a multitude of CFE methods have been proposed. While it is beyond the scope of this thesis to describe the generation methods in detail, in the following, some of the distinguishing properties between the methods are addressed using a review of 50 methods by Guidotti (2022) and of 56 methods by Verma *et al.* (2022) for tabular classification data sets. These reviews also include the multi-objective CFE method (hereinafter abbreviated as MOC), which is introduced in the contributing article of Chapter 8.

4.2 Counterfactual Explanations

Regression or Classification Most CFE methods focus on classification models \hat{f} and only a few methods consider regression models (Spooner *et al.*, 2021; Hada and Carreira-Perpiñán, 2021) – including MOC. MOC can be applied to prediction functions $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$, which naturally covers regression models. For classification models, it is assumed that the score or probability for a predefined class of interest is returned by \hat{f} . MOC also poses no restrictions on \mathcal{X} and covers all feature types. In contrast, 16 of the 50 methods considered by Guidotti (2022) can only handle numeric features.

Model-agnostic or Model-specific Model-agnostic interpretation methods do not rely on the internals of a trained ML model \hat{f} , so the methods can be applied to any \hat{f} . Model-specific methods are tailored to a specific ML algorithm, for example, differentiable models (neural network or linear model) or tree-based models. In the review papers of Guidotti (2022) and Verma *et al.* (2022), 50 % of the methods were model-agnostic and 50 % model-specific. MOC is part of the former.

Targeted Properties Almost all methods in the review papers consider the first three properties (validity, proximity, and sparsity). Plausibility can be guaranteed if CFEs are equal or highly similar to observations in a given data set – an approach that only 7 of the 50 methods considered by Guidotti (2022) follow. In MOC, the plausibility of CFEs is enhanced by adding the distance to observed data points as another objective in the underlying optimization task. Furthermore, the user can generate new points based on conditional distribution functions estimated by transformation trees (Hothorn and Zeileis, 2021). The actionability and causality properties require user input: a list of immutable features and a (partially known) causal graph. Less than half of the methods in Guidotti (2022) and Verma *et al.* (2022) consider Property 6 (actionability) and only 15 % consider Property 5 (causality); a causal graph requires some domain knowledge and is often based on untestable assumptions, reflecting a large burden for their application. MOC considers immutable features but not (yet) causality.

Strategy Guidotti (2022) differentiates between four strategies to generate CFEs, which are presented in the following.

The *first* strategy is based on instances: A CFE is derived as the most similar point to \mathbf{x}^* with a prediction in Y' in a given data set. This approach was first proposed for binary classification models by Wexler *et al.* (2019).

The *second* one is optimization: First, a loss function is derived based on the desired properties. This loss function is then optimized by an optimization method to generate CFEs. An example is the method by Wachter *et al.* (2018) for binary classification models. The method combines an objective for validity $o_{\text{valid}}(\mathbf{x})$ and an objective for proximity $o_{\text{prox}}(\mathbf{x})$ into a single loss function weighted by $\lambda \in \mathbb{R}^+$

$$o(\mathbf{x}) = \lambda \cdot o_{\text{valid}}(\mathbf{x}) + o_{\text{prox}}(\mathbf{x}).$$

\mathbf{x} is found by iteratively minimizing $o(\mathbf{x})$ while increasing λ . Choosing a balancing parameter λ and its factor of iterative increase is difficult and depends on a user’s preference and the given use case. Furthermore, the method only returns a single CFE without discussing the inherent trade-off between validity and proximity; if a CFE is close to the original data point, it also tends to have a similar prediction.

The *third* strategy is heuristic-based: These methods use local heuristics to minimize a given cost function. Also MOC follows this strategy by formalizing the task of generating CFEs multi-objectively. The four properties validity (o_{valid}), proximity (o_{prox}), sparsity (o_{sparse}) and plausibility (o_{plaus}) are considered simultaneously in the objective

$$\mathbf{o}(\mathbf{x}) := \left(o_{\text{valid}}(\hat{f}(\mathbf{x}), Y'), o_{\text{prox}}(\mathbf{x}, \mathbf{x}^*), o_{\text{sparse}}(\mathbf{x}, \mathbf{x}^*), o_{\text{plaus}}(\mathbf{x}, \mathcal{D}) \right). \quad (4.2)$$

Validity is measured by the L_1 -norm, proximity to \mathbf{x}^* by the Gower distance (Gower, 1971), sparsity by the L_0 -norm to \mathbf{x}^* , and plausibility by the weighted Gower distance to the closest points in a given data set \mathcal{D} (details are given in Chapter 8). Compared to the method by Wachter *et al.* (2018), Eq. (4.2) does not require a priori balancing of the objectives. A genetic algorithm optimizes the objective, a modified version of the non-dominated sorting genetic algorithm of Deb *et al.* (2002). Given an (initial) set of candidates, the algorithm pairwise recombines the best ones (according to Eq. 4.2), slightly mutates the values of the resulting candidates, and selects the best and most diverse ones for the next iteration. This guides the search toward a diverse set of Pareto-optimal CFEs such that trade-offs among the different objectives can be explored.

The *fourth* strategy is based on decision trees: First, a decision tree is trained on a given data set with the predictions of \hat{f} as the outcome variable. Approximating the behavior of a black box model with an interpretable model is another interpretation method called surrogate models or model distillation (Ribeiro *et al.*, 2016; Frosst and Hinton, 2017). Afterward, the tree structure is exploited to generate CFEs, for example, by following the leaves, leading to predictions different from \mathbf{x}^* . One disadvantage of this method is that it requires the tree to accurately approximate the behavior of \hat{f} , which is especially difficult to guarantee on the entire feature space. One approach is to build a *local* surrogate model that only focuses on the neighborhood of \mathbf{x}^* and the closest decision boundary, for example, by giving data points close by a higher weight for training the tree (Guidotti, 2022).

Number of CFEs Around 62 % of the methods considered by Guidotti (2022) and Verma *et al.* (2022) return only one CFE, although a set of CFEs is preferable because multiple, equally good counterfactuals with the desired prediction can exist (referred to as the Rashomon effect (Breiman, 2001b)). This is one pitfall often overlooked in research, as discussed in Section 8 of the contributing article in Chapter 7. Furthermore, a set is more likely to encompass a CFE that aligns with a user’s latent preferences. This is why, for MOC, the generation of CFEs was formalized as a multi-objective problem; the method returns a Pareto-set of equally good CFEs. The underlying genetic algorithm was also adapted to improve the diversity of CFEs in terms of their feature values.

Software Of the 50 considered papers in Guidotti (2022), only 32 offer an implementation for their methods. 30 of them are implemented in Python (Van Rossum and Drake Jr, 1995), one in Julia (Bezanson *et al.*, 2017) and one (MOC) in R (R Core Team, 2022). Therefore, R and Julia users face limited access to CFE methods and limited comparability due to the lack of a common interface. The **counterfactuals** package introduced in the contributing article of Chapter 10 offers the first user-friendly and unified interface for CFE methods in R. The package currently offers three methods as well as some optional enhancements for generalization and comparability, with an emphasis on the generation of a set of counterfactuals. Unified evaluation and visualization

4.3 Semi-factual Explanations

methods for all implemented CFE methods help to compare them to each other. The modularity of the package allows for adding new CFE methods in the future.

4.2.3 Connection to Counterfactual Fairness

As noted in Table 4.1, CFEs can help to detect adverse predictions of a model. This is the case if a CFE (that at least fulfills validity and proximity) proposes a change in a feature that, from a normative perspective, should not lead to a change in prediction. These features are called protected attributes (PA). Examples are gender, religion, or sexual orientation. CFEs that propose a change in a PA indicate discriminative behavior of the underlying prediction function \hat{f} . The opposite is not necessarily true: A discriminatory \hat{f} does not necessarily result in CFEs with changes in the PA; likewise, a CFE that does not change the PA is not an indicator for a non-discriminatory \hat{f} .

Kusner *et al.* (2017) introduced a causal fairness notion for binary classification models based on the definition of counterfactuals by Pearl *et al.* (2016), given in Eq. (4.1). It defines a predictor \hat{Y} as counterfactually fair if the distribution of the predictions remains unchanged when a PA A is changed from one value to any other value $a' \in A$, i.e.,

$$\mathbb{P}(\hat{Y}(A = a) = y \mid \mathbf{Z} = \mathbf{z}, A = a) = \mathbb{P}(\hat{Y}(A = a') = y \mid \mathbf{Z} = \mathbf{z}, A = a),$$

with $X := (\mathbf{Z}, A)$ such that \mathbf{Z} is the set of features excluding A . Compared to CFEs for model interpretation, this definition does not rely on counterfactuals that lead to a different model prediction but on realistic counterfactuals that adhere to causal knowledge. The authors also propose a method to compute $\hat{Y}(A = a)$ for $\forall a \in A$ similar to the three-step approach by Pearl *et al.* (2016) (see Section 4.1), which requires (at least) access to the underlying causal graph.

The contributing article of Chapter 9 presents a fairness notion for binary classification models for scenarios without knowledge of the causal graph. It relies on MOC, where the first objective (o_{valid}) is adapted. Instead of aiming for a counterfactual with a prediction equal to the desired prediction, the objective aims for a counterfactual with a high likelihood of belonging to a different protected group instead of the current one. The genetic algorithm returns a Pareto-optimal set that represents a distribution over counterfactuals, accounting for potential stochasticity in the data-generating process. Based on this set of counterfactuals, the manuscript also presents fairness evaluation criteria for trained models.

4.3 Semi-factual Explanations

As seen in Section 4.1, counterfactuals in causality are not generated to change a prediction but to adhere to causal knowledge (such that a *potential* change in the prediction can be defined as a causal effect). Thus, they provide not only the basis for CFEs but also for SFEs, where for the former, feature changes should lead to a prediction change and for the latter not. Relative to CFEs, SFEs are less explored in the literature, although their philosophical and psychological implications have been studied for many decades already (Goodman, 1947; Bennett, 1982; McCloy and Byrne, 2002). Searches for the terms “semi-factual explanations” and “semifactual explanations” on Web of Science on 15.08.23 returned two published articles, compared to 168 for “counterfactual

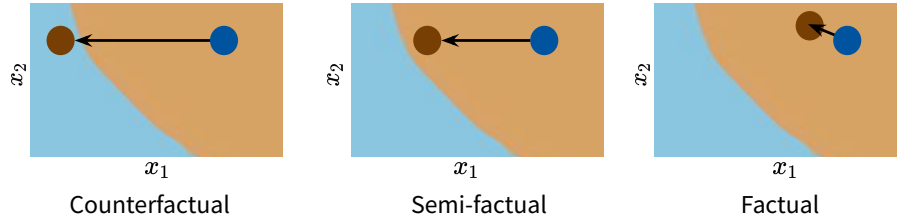


Figure 4.3: Illustration of counter-, semi-, and factual explanations for a binary classification data set with two features. The background color reflects the two classes (blue vs. brown). The observation to explain is the blue dot and the respective explanation is the brown dot.

explanations” (Clarivate, 2023). The following definition formalizes SFEs. As in Definition 1, only models $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$ are considered.

Definition 2 (Semi-factual explanation). *Given the prediction function $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$, an observation of interest \mathbf{x}^* and a set or interval of desired predictions $Y' \subset \mathbb{R}$ with $\hat{f}(\mathbf{x}^*) \in Y'$, a point $\mathbf{x} \in \mathcal{X}$ is an SFE for \mathbf{x}^* if it differs to \mathbf{x}^* in a few features while $\hat{f}(\mathbf{x}) \in Y'$.*

SFEs give insights into a prediction model by highlighting feature changes to a point of interest for which the prediction does not change. SFEs, therefore, follow the notion of *a fortiori* arguments that express justification of a prediction by an example with “less convincing” feature values that has the same prediction (Nugent *et al.*, 2009).

SFEs differ from factual explanations (FEs), because FEs follow the notion of *similia similibus* that similar inputs result in similar predictions, and becoming aware of such similarities leads to a greater comprehension of the model (Nugent *et al.*, 2009). An FE for the above credit example would be “your credit was of high risk because a customer with the same feature values, although one year older than you, was also classified as a high risk”. The difference is that SFEs are even more convincing if they lie close to the closest decision boundary of \mathbf{x}^* and not just close to \mathbf{x}^* . For example, the argument “even if you had applied for a credit of € 3900 instead of € 4000, your application would still be classified as a high risk” would be less convincing than a change to € 3000. Figure 4.3 visualizes the differences between CFEs, SFEs, and FEs for a binary classification model with two features. As with CFEs, desired properties of SFEs can be specified.

4.3.1 Desired Properties

The following list of desired properties for an SFE \mathbf{x} is based on Aryal and Keane (2023) and Artelt and Hammer (2022).

Property 7 (Validity). \mathbf{x} should differ to \mathbf{x}^* , i.e. $\exists j \in \{1, \dots, p\} : x_j \neq x_j^*$, while $\hat{f}(\mathbf{x}) \in Y'$.

For SFEs, a change in feature values is explicitly required, while for CFEs, it is implicitly included because only feature changes can lead to a different prediction than $\hat{f}(\mathbf{x}^*)$.

All other properties overlap with the properties of CFEs, namely sparsity (Property 3), plausibility (Property 4), causality (Property 5), and actionability (Property 6). The only difference is that

proximity is no longer a desired property – otherwise, we would generate FEs. Artelt and Hammer (2022) define a “distance” property: the distance between \mathbf{x} and \mathbf{x}^* should be “reasonably large“. This definition is rather vague and requires some further considerations of what it exactly means and how it could be operationalized (see the Outlook, Chapter 12).

4.3.2 Generation Methods

Compared to CFEs, only a few methods exist for generating SFEs for tabular data. Most of the methods were proposed for (binary) classification models, return a single SFE, and are instance-based, meaning that an SFE is chosen among the set of observed data points with the same prediction as \mathbf{x}^* .

Different criteria were proposed to select one instance as an SFE from this set. Doyle *et al.* (2004) base the selection on a user-defined utility function that reflects how convincing feature changes are to justify the status quo. Deriving an appropriate utility function depends on the use case and is knowledge-intensive, which makes the generation method difficult to use in practice. To overcome this problem, Nugent *et al.* (2009) fit a logistic regression model to the instances surrounding \mathbf{x}^* and its closest decision boundary, equal to a local surrogate model (Ribeiro *et al.*, 2016). The SFE is the instance with a probability closest to a defined threshold (e.g., 0.5) and is, therefore, closest to the decision boundary. Cummins and Bridge (2012) choose the instance as SFE that is closest to the nearest CFE of \mathbf{x}^* , while Aryal and Keane (2023) choose the instance that maximizes an objective that aims for a few but large features changes w.r.t. \mathbf{x}^* .

The method of Artelt and Hammer (2022) differs from the above in that it returns a set of diverse SFEs. Therefore, it takes into account that there can be multiple SFEs that differ in the proposed feature changes. The method iteratively generates SFEs by optimizing a single objective with the Nelder-Mead method (Nelder and Mead, 1965). The objective is a weighted sum of multiple objectives that promote validity (Property 7), sparsity (Property 3), distance to \mathbf{x}^* , and diversity w.r.t. to the SFEs that were already found. The number of iterations specifies how many SFEs are returned. The open questions are: How many SFEs are enough, and how to avoid users being overwhelmed by the number of SFEs?

The contributing article of Chapter 11, tries to answer these questions by summarizing a set of SFEs in an interpretable way: in the form of a hyperbox with p dimensions, intervals for real-valued features and a subset of the potential classes for categorical features. For the generation of hyperboxes, previous methods for generating hyperboxes were reviewed and modified to embed them in a general framework. A benchmark study compares the adapted methods based on a set of proposed quality measures. The observation that no method “rules them all” underlines the need for a unifying framework comprising multiple methods. Overall, these investigations formalize a new class of local interpretations called interpretable regional descriptors.

Part II

Contributions

5 What Makes Forest-Based Heterogeneous Treatment Effect Estimators Work?

Contributing Article

Dandl S, Haslinger C, Hothorn T, Seibold H, Sverdrup E, Wager S, Zeileis A (2023b). “What Makes Forest-Based Heterogeneous Treatment Effect Estimators Work?” *arXiv 2206.10323 v2*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2206.10323. To appear in *The Annals of Applied Statistics*

The article was accepted at *The Annals of Applied Statistics* shortly before this thesis was submitted. The following manuscript is the accepted version of the work available on arXiv.

Replication Code

The code for replicating the results in this manuscript is available as part of the R package **htesim** available at <https://github.com/dandls/htesim>.


Declaration of Contributions

Susanne Dandl implemented the orthogonalization approach, the real-world use case, and the infrastructure to conduct the simulation study in parallel in R. She implemented an R package to flexibly simulate from diverse data generating processes based on a first code base of Torsten Hothorn. She performed the experiment, and aggregated and interpreted the results. Susanne Dandl wrote major parts of the first draft of the paper and created all the included figures. She contributed substantially to the revision of the manuscript.

Contributions of Co-authors


All co-authors contributed to the formulation and evolution of overarching research goals and aims for the manuscript. Torsten Hothorn wrote a first draft of Sections 2 and 4 which were majorly extended and rewritten by Susanne Dandl. He also contributed the initial code base for the experimental design. All co-authors helped to revise the manuscript.

What Makes Forest-Based Heterogeneous Treatment Effect Estimators Work?

Susanne Dandl 
LMU Munich
MCML

Christian Haslinger 
Universitätsspital und
Universität Zürich

Torsten Hothorn 
Universität Zürich

Heidi Seibold 
IGDORE München

Erik Sverdrup 
Stanford University

Stefan Wager 
Stanford University

Achim Zeileis 
Universität Innsbruck

Abstract

Estimation of heterogeneous treatment effects (HTE) is of prime importance in many disciplines, from personalized medicine to economics among many others. Random forests have been shown to be a flexible and powerful approach to HTE estimation in both randomized trials and observational studies. In particular “causal forests”, introduced by Athey, Tibshirani, and Wager (2019), along with the R implementation in package `grf` were rapidly adopted. A related approach, called “model-based forests”, that is geared towards randomized trials and simultaneously captures effects of both prognostic and predictive variables, was introduced by Seibold, Zeileis, and Hothorn (2018) along with a modular implementation in the R package `model4you`.

Neither procedure is directly applicable to the estimation of individualized predictions of excess postpartum blood loss caused by a cesarean section in comparison to vaginal delivery. Clearly, randomization is hardly possible in this setup and thus model-based forests lack clinical trial data to address this question. On the other hand, the skewed and interval-censored postpartum blood loss observations violate assumptions made by causal forests. Here, we present a tailored model-based forest for skewed and interval-censored data to infer possible predictive prepartum characteristics and their impact on excess postpartum blood loss caused by a cesarean section.

As a methodological basis, we propose a unifying view on causal and model-based forests that goes beyond the *theoretical* motivations and investigates which *computational* elements make causal forests so successful and how these can be blended with the strengths of model-based forests. To do so, we show that both methods can be understood in terms of the same parameters and model assumptions for an additive model under L_2 loss. This theoretical insight allows us to implement several flavors of “model-based causal forests” and dissect their different elements *in silico*.

The original causal forests and model-based forests are compared with the new blended versions in a benchmark study exploring both randomized trials and observational settings. In the randomized setting, both approaches performed akin. If confounding was present in the data generating process, we found local centering of the treatment indicator with the corresponding propensities to be the main driver for good performance. Local centering of the outcome was less important, and might be replaced or enhanced by simultaneous split selection with respect to both prognostic and predictive effects. This lays the foundation for future research combining random forests for HTE estimation with other types of models.

Keywords: Causal forests, heterogeneous treatment effects, observational data, personalized medicine, postpartum hemorrhage, random forest.

arXiv:2206.10323v2 [stat.ME] 20 Dec 2023

1. Introduction

1.1. Challenges in treatment effect estimation for cesarean sections

Cesarean section is the most frequent surgical procedure performed in young and healthy women, with currently one out of three babies in the USA being born that way (Antoine and Young 2021). Short-term postpartum benefits and the perceived safety of the procedure explain the increase in popularity over the last 50 years, including the rise of electively performed cesarean sections. At the same time, maternal mortality and morbidity increased globally (WHO 2012; Say *et al.* 2014). More recently, adverse long-term effects, including gynecological and obstetrical complications in mothers as well as potential and controversially discussed immune disorders in their children, have gained attention (Antoine and Young 2021). Lack of clinical trial data directly comparing outcomes of natural births with those following cesarean sections render characterization and quantification of such effects challenging. Postpartum hemorrhage (PPH), defined as blood loss ≥ 500 mL within 24 hours after delivery by the WHO (2012), is a short-term complication associated with maternal morbidity and mortality worldwide. The prevalence of PPH is increasing in industrialized countries (for the USA, see MacDorman, Declercq, Cabral, and Morton 2016).

Management of PPH requires identification of at risk parturients and calls went out to the statistics, machine learning, and artificial intelligence communities to develop and evaluate prognostic models (Ende 2022). Typically, models for dichotomized PPH prognosis were created aiming at either women giving birth by vaginal delivery (Erickson and Carlson 2020; Akazawa, Hashimoto, Katsuhiko, and Kaname 2021) or at women scheduled for a cesarean section (Kawakita, Mokhtari, Huang, and Landy 2019). Models trained on data from both modes of delivery are rare, e.g., in Venkatesh *et al.* (2020) the mode of delivery was not taken into account as risk factor. Because of the often elective nature of the decision to undergo cesarean section, a quantification of the *additional* amount of hemorrhaging caused by surgery is relevant for the decision process, however, such information is hard to extract from stratified prognostic models. This is true even more considering the possibility of unplanned cesarean deliveries following attempted vaginal deliveries. From a statistical perspective, estimation of a heterogeneous cesarean section effect is non-trivial for a number of reasons. First, potential risk factors for PPH, such as age of the mother, estimated birth weight, gestational age, previous PPH, suspected placental disorders, or multifetal pregnancy might have an impact on both the decision to undergo a cesarean section (treatment) and postpartum blood loss (outcome). Randomization of mode of delivery is impossible and thus effects have to be estimated from observational data. Second, it is hard to obtain exact measurements of postpartum blood loss in the often hectic environment of a delivery ward, and thus imprecise assessments via interval-censored observations are only available. Third, one has to expect a high level of skewness and extreme values in blood loss measurements, rendering strong distributional assumptions questionable. Last, the association of prognostic factors and blood loss is expected to be complex, including nonlinear and interaction terms.

1.2. Heterogeneous treatment effect estimation and random forests

In the statistical literature, methods for the estimation of such heterogeneous treatment effects (HTEs) from randomized trials or observational studies has been receiving a lot of attention

during the past decade, triggered by an increasing demand from personalized medicine and the need for refined methods in causal inference. In particular, different variations of random forests (Breiman 2001) have been suggested for HTE estimation, and seem promising candidates for addressing the statistical challenges we are facing here. Random forest variants for HTE estimation can be roughly grouped in two classes.

The *first class* of methods employs random forests to estimate the expected outcomes given covariates separately in the treatment groups. The conditional average treatment effect (CATE) then corresponds to the difference in estimated mean factual and counterfactual outcomes. Notably, the virtual twins method (Foster, Taylor, and Ruberg 2011) has adopted this approach using random forests. Improvements can be obtained by additionally considering treatment-covariate-interactions or fitting separate (synthetic) forests for each treatment group (Foster *et al.* 2011; Dasgupta, Szymczak, Moore, Bailey-Wilson, and Malley 2014; Ishwaran and Malley 2014). Moreover, Lu, Sadiq, Feaster, and Ishwaran (2018) proposed a bivariate imputation approach which uses a bivariate splitting rule (Ishwaran, Kogalur, Blackstone, and Lauer 2008; Tang and Ishwaran 2017) that simultaneously considers the expected outcome under both treatments. In a more general setup, Künzel, Sekhon, Bickel, and Yu (2019) introduced X-learners, a class of meta-algorithms which build upon any supervised/regression algorithm including random forests, Bayesian regression trees (BART, Chipman, George, and McCulloch 2010; Hill 2011; Starling, Murray, Lohr, Aiken, Carvalho, and Scott 2021), or neural networks. Most forest methods were initially developed for randomized controlled trials and have later been adapted to be more robust to confounding. For example, the pollinated transformed outcome forests of Powers *et al.* (2018) build a single forest on propensity score weighted outcomes instead of the original outcomes to account for confounding.

The subject of this paper is the *second class* of random forest-type algorithms aiming at the *direct* estimation of HTEs in a model-driven way. Two such approaches, “causal forests” (Athey *et al.* 2019) and “model-based forests” (Seibold *et al.* 2018), have recently been proposed. “Causal forests” by Athey *et al.* (2019) implement a divide-and-conquer strategy, also referred to as “local centering” or “orthogonalization” for the direct estimation of HTEs from observational data. They first account for the dependence of both the marginal mean of the outcome and the treatment propensity on the available covariates. Subsequently, they exclusively focus on the estimation of the HTEs. In terms of distributional assumptions, causal forests have been developed for continuous outcomes and corresponding conditional means and the squared error loss plays an important role in the motivation of this algorithm. Cui, Kosorok, Sverdrup, Wager, and Ruoqing (2022) also applied causal forests to survival data and Mayer, Sverdrup, Gauss, Moyer, Wager, and Josse (2020) discussed strategies to handle missing values. We note that earlier causal tree and forest algorithms described in Imbens and Athey (2016) and Wager and Athey (2018) do not involve such a local centering step. In this paper, we use the term causal forests to describe the algorithm from Athey *et al.* (2019); see also Athey and Wager (2019). Causal forests are implemented in the R package `grf` (Tibshirani, Athey, Sverdrup, and Wager 2021).

“Model-based forests” by Seibold *et al.* (2018) simultaneously estimate prognostic effects and HTEs. They do so by leveraging model-based recursive partitioning (“MOB”, Zeileis, Hothorn, and Hornik 2008), a technique for learning model trees in which all relevant parameters are re-estimated in each subset of a tree. MOB is not a specific model but rather a general framework for model construction where the adaptation to different types of models

often still necessitates working out the details of parameter interpretation or model assessment, etc. Seibold, Zeileis, and Hothorn (2016) have adapted MOB to model-based trees for HTE, working out the details for Gaussian regression models as well as censored survival models (parametric Weibull model and semi-parametric Cox model). Subsequently, Seibold *et al.* (2018) have extended this work to model-based forests for HTEs, again working out the details of Gaussian regression and censored Weibull survival modeling. Other authors have adapted the general MOB idea to outcome variables on other scales and/or subject to censoring and truncation, e.g., as in survival data (Korepanova, Seibold, Steffen, and Hothorn 2020), ordinal data (Buri and Hothorn 2020), generalized mixed models (Fokkema, Smits, Zeileis, Hothorn, and Kelderman 2018), or transformation models (Hothorn and Zeileis 2021b). So far, model-based forests have only been developed for HTE estimation based on randomized trial data.

1.3. Model-based causal forests for postpartum blood loss

Neither of the random forest approaches from Section 1.2 is directly applicable to the estimation of heterogeneous cesarean section effects, described in Section 1.1. Our main contribution is therefore a novel random forest model that combines the strengths of the existing methods to tackle the challenges in the cesarean section data. We approach this problem by first studying the similarities and differences between causal forests and model-based forests theoretically and empirically. In a second step, we identify the key drivers for good HTE estimation performance in observational data on the one hand and for asymmetric and potentially interval-censored outcomes on the other hand. Lastly, we derive and apply the novel “blended” HTE random forest for PPH by combining the elements identified as being instrumental.

Given that both causal forests and model-based forests encompass additive models under L_2 loss, we adopt this modeling framework to investigate the specific elements that explain both the success of causal forests for observational studies and the flexibility of model-based forests for randomized trials. Specifically, the question of how the disparate strategies for handling the prognostic and confounding effects differ – or how they can be combined – is of both theoretical and practical interest. For obtaining some answers to this question, we employ the modular computational toolbox for tree induction and forest inference in the R package **model4you** (Seibold, Zeileis, and Hothorn 2019) which allows to “mix & match” the elements of both model-based and causal forests.

The results lay the foundation for future research that further expands potential synergies in HTE estimation using *model-based causal forests* by blending model-based and causal forests to leverage the strengths of both approaches. To demonstrate this in practice, we investigate the effect of cesarean section on postpartum blood loss in comparison to vaginal deliveries based on a prospective observational study from Switzerland. In this application, there is a need for a model-based approach that can deal with the skewed outcome distribution which is also interval-censored due to the lack of precise measurement techniques. Thus, we showcase a model-based transformation forest applicable to this observational setting. Our contributions here are three-fold: First, we provide a unified understanding of causal forests and model-based forests for HTE estimation in Section 2. Second, we evaluate why these methods work in different scenarios and what the key drivers for good HTE estimation performance in the observational setting are in Section 4. Last, based on the insights gained theoretically

and empirically, we discuss a novel “blended” random forest model in Section 3 specifically designed for blood loss prediction by pooling key components from causal and model-based forests (Section 5).

2. Models and forest algorithms

In this section, we first outline similarities and differences between causal forests and model-based forests theoretically, using the basic setup of regression for real-valued outcomes. Subsequently, two novel blended approaches are introduced that adapt HTE estimation with model-based forests to observational data.

2.1. The interaction model

We are interested in the conditional mean of a real-valued outcome $Y \in \mathbb{R}$, given covariates $\mathbf{X} \in \mathcal{X}$ under a specific binary treatment or intervention $W \in \{0, 1\}$, corresponding to control vs. treatment. Under the assumptions that a binomial model $W \mid \mathbf{X} = \mathbf{x} \sim \text{B}(1, \pi(\mathbf{x}))$ with propensities $\pi(\mathbf{x}) = \text{P}(W = 1 \mid \mathbf{X} = \mathbf{x}) = \mathbb{E}(W \mid \mathbf{X} = \mathbf{x})$ describes treatment assignment and residuals are given by an error term σZ with $\mathbb{E}(Z \mid \mathbf{X}, W) = 0$ and standard deviation $\sigma > 0$, the model reads

$$Y = \mu(\mathbf{X}) + \tau(\mathbf{X})W + \sigma Z \quad (1)$$

with conditional mean function

$$\mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}) = \mu(\mathbf{x}) + \tau(\mathbf{x})\pi(\mathbf{x}) =: m(\mathbf{x}).$$

Covariates \mathbf{x} with impact on the prognostic effect $\mu(\mathbf{x})$ are called *prognostic*, while covariates affecting the treatment effect $\tau(\mathbf{x})$ are called *predictive*. Treatment assignment is assumed to be non-deterministic, *i.e.* propensity scores have to be bounded away from zero and one

$$0 < \pi(\mathbf{x}) = \text{P}(W = 1 \mid \mathbf{X} = \mathbf{x}) = \mathbb{E}(W \mid \mathbf{X} = \mathbf{x}) < 1.$$

Personalized medicine and causal inference in general focus on the estimation of the heterogeneous treatment effect $\tau(\mathbf{x})$ and thus on the impact of predictive variables on treatment success; and accurate estimation of $\tau(\mathbf{x})$ is the main goal of all methods discussed in this paper.

As discussed in Nie and Wager (2021), the interaction model (1) is closely connected to a treatment model with potential outcomes (Imbens and Rubin 2015), where we posit potential outcomes $Y(0)$ and $Y(1)$ corresponding to the outcome a unit would have experienced without or with treatment respectively, and assume that we observe $Y = Y(W)$. Then under unconfoundedness (Rosenbaum and Rubin 1983)

$$(Y(0), Y(1)) \perp\!\!\!\perp W \mid \mathbf{X} = \mathbf{x},$$

we can define residuals σZ in (1) such that the interaction model is observationally equivalent to the specification using potential outcomes, and

$$\tau(\mathbf{x}) = \text{CATE}(\mathbf{x}) = \mathbb{E}(Y(1) - Y(0) \mid \mathbf{X} = \mathbf{x})$$

can be interpreted as the conditional average treatment effect. We note that in a uniformly randomized trial, we have $W \perp\!\!\!\perp \{\mathbf{X}, Y(0), Y(1)\}$ and so unconfoundedness is always satisfied, and the propensity scores $\pi(\mathbf{x}) \equiv \pi$ are constant by design.

2.2. Causal forests

For developing causal forests, [Athey et al. \(2019\)](#) rewrite Equation (1) as

$$\begin{aligned} (Y \mid \mathbf{X} = \mathbf{x}) &= m(\mathbf{x}) - m(\mathbf{x}) + \mu(\mathbf{x}) + \tau(\mathbf{x})W + \sigma Z \\ &= m(\mathbf{x}) + \tau(\mathbf{x})(W - \pi(\mathbf{x})) + \sigma Z \end{aligned} \quad (2)$$

which motivates their algorithmic approach of eliminating the marginal mean $m(\mathbf{x}) = \mathbb{E}(Y \mid \mathbf{X} = \mathbf{x})$ and propensities $\pi(\mathbf{x}) = \mathbb{E}(W \mid \mathbf{X} = \mathbf{x})$ first before estimating the heterogeneous treatment effect $\tau(\mathbf{x})$. This orthogonalization (introduced by [Robinson 1988](#)) is also called “local centering” because both outcome $Y - \hat{m}(\mathbf{x})$ and treatment indicator $W - \hat{\pi}(\mathbf{x})$ are centered before $\tau(\mathbf{x})$ is estimated. This approach leads to more robustness to confounding effects in case of observational data because it regresses out the effect of covariates \mathbf{X} on Y and W ([Nie and Wager 2021](#)). While in principle any non-parametric regression technique could be applied to estimate $m(\mathbf{x})$ and $\pi(\mathbf{x})$, [Athey et al. \(2019\)](#) chose regression forests.

In the second step of causal forests, treatment effects $\tau(\mathbf{x})$ in the model

$$(Y \mid \mathbf{X} = \mathbf{x}, W = w) = \hat{m}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x})) + \sigma Z$$

are then estimated by minimizing the L_2 loss

$$\ell_{\text{cf}}(\tau(\mathbf{x})) := 1/2 (Y - \hat{m}(\mathbf{x}) - \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x})))^2$$

w.r.t. τ , the only unknown quantity in this loss function.

Specifically, when splitting a (parent) node, cut-point estimation for causal trees relies first on estimating a constant treatment effect $\hat{\tau}$ in the parent node minimizing $\ell_{\text{cf}}(\tau)$ by solving the score equation

$$s_{\text{cf}}(\tau) = -\frac{\partial \ell_{\text{cf}}(\tau)}{\partial \tau} = (Y - \hat{m}(\mathbf{x}) - \tau(w - \hat{\pi}(\mathbf{x}))) (w - \hat{\pi}(\mathbf{x})) = 0 \quad (3)$$

and second on regressing the resulting score

$$s_{\text{cf}}(\hat{\tau}) = (Y - \hat{m}(\mathbf{x}) - \hat{\tau}(w - \hat{\pi}(\mathbf{x}))) (w - \hat{\pi}(\mathbf{x}))$$

on \mathbf{x} by means of a simple cut-point model. The classical simultaneous analysis-of-variance (ANOVA) selection of split variable and cut-point is implemented. Causal forests are robust to confounding because the score equation (3) is Neyman-orthogonal in the sense of [Chernozhukov et al. \(2018\)](#), thus enabling it to accurately target $\tau(\mathbf{x})$ even when estimators for the nuisance components $\pi(\mathbf{x})$ or $\mu(\mathbf{x})$ may be somewhat imprecise ([Nie and Wager 2021](#)). Of course, causal forests can be also applied to randomized data, in which case treatment should be centered by the true randomization probability π .

2.3. Model-based forests

In contrast to the marginal model (1) motivating local centering in causal forests, model-based forests (Seibold *et al.* 2018) for real-valued outcomes are based on a model which, in addition to \mathbf{x} , also conditions on treatment assignment $W = w$:

$$(Y \mid \mathbf{X} = \mathbf{x}, W = w) = \mu(\mathbf{x}) + \tau(\mathbf{x})w + \sigma Z. \quad (4)$$

The main difference between causal forests and model-based forests is that the latter aims to estimate both $\mu(\mathbf{x})$ and $\tau(\mathbf{x})$ simultaneously, whereas the former applies local centering in a two-step approach, that is, treating the prognostic effect $\mu(\mathbf{x})$ as a nuisance parameter. More specifically, by using model (4) instead of model (2), $(\mu(\mathbf{x}), \tau(\mathbf{x}))^\top$ is *simultaneously* estimated by minimizing the L_2 loss

$$\ell_{\text{mob}}(\mu(\mathbf{x}), \tau(\mathbf{x})) = 1/2 (Y - \mu(\mathbf{x}) - \tau(\mathbf{x})w)^2 \quad (5)$$

w.r.t. μ and τ , the two unknown quantities in this loss function.

Model-based forests separate split-variable and cut-point selection in a way inspired by unbiased recursive partitioning procedures. Specifically, in each node, constants $(\hat{\mu}, \hat{\tau})^\top$ are estimated by minimizing

$$\ell_{\text{mob}}(\mu, \tau) := 1/2 (Y - \mu - \tau w)^2$$

w.r.t both μ and τ . A split variable is selected by a bivariate permutation test relying on a quadratic test statistic for the null hypothesis that μ and τ are constant and independent of any split variable \mathbf{X} . For splitting, the variable is selected that has the lowest p -value. Afterwards, a cut-point is found by regressing the bivariate score

$$s_{\text{mob}}(\hat{\mu}, \hat{\tau}) := (Y - \hat{\mu} - \hat{\tau}w)(1, w)^\top \quad (6)$$

on covariates \mathbf{x} by a simple bivariate cut-point model. A cut-point is selected as the point that results in the largest discrepancy between the score functions in the two resulting subgroups (details are given in Appendix 2, Seibold *et al.* 2018). The core idea of this tree-induction method originates from unbiased recursive partitioning (Hothorn, Hornik, and Zeileis 2006) and the introduction of multiple model-based scores (Zeileis *et al.* 2008) in this framework. Section 1 in the Supplementary Material A provides a more detailed comparison of the cut-point selection of model-based forests with causal forests.

As a side-effect, heterogeneous treatment contrasts $\tau_{2-1}(\mathbf{x}), \tau_{3-1}(\mathbf{x}), \dots, \tau_{K-1}(\mathbf{x})$ of $K > 2$ treatment groups $W \mid \mathbf{X} = \mathbf{x} \sim \text{M}(K, \pi(\mathbf{x}))$ from a multinomial distribution can be estimated by model-based forests. In each node, the criterion

$$\frac{1}{2} \left(Y - \mu(\mathbf{x}) - \sum_{k=2}^K \tau_{k-1}(\mathbf{x})w_{k-1} \right)^2$$

is then minimized w.r.t. μ and all treatment contrasts τ_{k-1} for $k = 2, \dots, K$ simultaneously. This allows the comparison of the effects of different treatments or one treatment with various doses to a placebo (application examples could be found in Schnell, Tang, Müller, and Carlin 2017; Feng, Zhou, Zou, Fan, and Li 2012; Zanutto, Lu, and Hornik 2005).

2.4. Aggregation and honesty

Once multiple trees have been fitted to sub-samples of the data, causal forests and model-based forests apply the same local maximum likelihood aggregation scheme based on nearest

neighbor weights for the estimation of heterogeneous treatment effects $\tau(\mathbf{x})$ (Hothorn, Lausen, Benner, and Radespiel-Tröger 2004; Meinshausen 2006; Lin and Jeon 2006; Athey *et al.* 2019; Hothorn and Zeileis 2021b). First, nearest neighbor weights $\alpha_i(\mathbf{x})$ are derived from the B trees in a forest fitted to observations $(Y_i, \mathbf{x}_i, w_i), i = 1, \dots, N$. These weights measure the relevance of a training observation i for estimating $\tau(\mathbf{x})$. For a forest with B trees, $\alpha_i(\mathbf{x})$ for an observation \mathbf{x} is equal to the frequency with which the i -th training sample falls in the same leaf as \mathbf{x} over all B trees. In a second step, $\tau(\mathbf{x})$ is estimated using the reweighted training data by minimizing

$$\hat{\tau}(\mathbf{x}) = \arg \min_{\tau} \sum_{i=1}^n \alpha_i^{\text{cf}}(\mathbf{x}) \ell_{\text{cf},i}(\tau)$$

in causal forests and

$$(\hat{\mu}(\mathbf{x}), \hat{\tau}(\mathbf{x}))^{\top} = \arg \min_{\mu, \tau} \sum_{i=1}^n \alpha_i^{\text{mob}}(\mathbf{x}) \ell_{\text{mob},i}(\mu, \tau)$$

in model-based forests, where $\ell_{\text{cf},i}$ and $\ell_{\text{mob},i}$ denote the loss for the i -th observation and α_i^{cf} and α_i^{mob} are the weights obtained from a causal forest and a model-based forest, respectively.

Wager and Athey (2018) additionally recommend a sub-sample splitting technique called honesty: “a tree is honest if, for each training example i , it only uses the response Y_i to estimate the within-leaf treatment effect τ [...] or to decide where to place the splits, but not both”. They empirically and theoretically proved that honesty is necessary to accomplish valid statistical inference. This technique is independent of both tree-induction and forest aggregation and can be applied in both causal forests and model-based forests. In the following, we refer to the *adaptive version* of a tree fitting process, when no sample splitting is conducted, and we refer to the *honest version*, when honesty is performed.

2.5. Model generalizations

When heterogeneous treatment effects shall be estimated for an outcome variable Y that is not well described by model (1), adaptations to both causal forests and model-based forests are necessary. Causal forests rely on reformulations of the corresponding estimation problems such that the squared error loss can also be applied in other contexts, for example in survival analysis (Cui *et al.* 2022). For model-based forests, the loss function ℓ_{mob} (5) changes from squared error to the negative log-likelihood of some appropriate model (see Seibold *et al.* 2016, 2018; Korepanova *et al.* 2020; Buri and Hothorn 2020; Fokkema *et al.* 2018; Hothorn and Zeileis 2021b).

As a simple example, consider count observations $(Y | \mathbf{X} = \mathbf{x}, W = w) \sim \text{Po}(\exp(\mu(\mathbf{x}) + \tau(\mathbf{x})w))$ from a conditional Poisson distribution. A “Poisson forest” for HTE estimation can be implemented by replacing the squared error loss (5) with the corresponding Poisson negative log-likelihood

$$\ell_{\text{mob}}(\mu(\mathbf{x}), \tau(\mathbf{x})) = \exp(\mu(\mathbf{x}) + \tau(\mathbf{x})w) - (\mu(\mathbf{x}) + \tau(\mathbf{x})w)Y.$$

When it is appropriate to assume $Z \sim \text{N}(0, 1)$ with cumulative distribution function Φ , the conditional distribution $(Y | \mathbf{X} = \mathbf{x}, W = w) \sim \text{N}(\mu(\mathbf{x}) + \tau(\mathbf{x})w, \sigma^2)$ is also normal with cumulative distribution function

$$\text{P}(Y \leq y | \mathbf{X} = \mathbf{x}, W = w) = \Phi\left(\frac{y - \mu(\mathbf{x}) - \tau(\mathbf{x})w}{\sigma}\right).$$

For an observed interval $y < Y \leq \bar{y}$, model-based forests equipped with the negative log-likelihood

$$\ell_{\text{mob}}(\mu(\mathbf{x}), \tau(\mathbf{x}), \sigma) = -\log \left(\Phi \left(\frac{\bar{y} - \mu(\mathbf{x}) - \tau(\mathbf{x})w}{\sigma} \right) - \Phi \left(\frac{y - \mu(\mathbf{x}) - \tau(\mathbf{x})w}{\sigma} \right) \right)$$

allows us to implement a variant of model-based forests applicable to imprecise interval-censored observations. In a Tobit model, this is the negative log-likelihood contributed by an observation $(-\infty, 0]$ left-censored at zero (Schlosser, Hothorn, Stauffer, and Zeileis 2019, equation (2.1)). A similar likelihood, however without the strict normal assumption, will be introduced for interval-censored blood loss in Section 5.1. In this sense, model-based forests can be understood as a conceptual and computational framework for method construction, rather than a model with a special domain of application.

3. Strategies and research questions for blended approaches

When applied to data well-described by the additive model (1) in the randomized setting, the principles underlying causal forests and model-based forests are conceptually the same, the only difference is that causal forests follow a sequential two-step approach and model-based forests implement a simultaneous approach to parameter estimation. We are now interested in assessing the impact of implementation details in causal forests and model-based forests on HTE estimation performance by the two algorithms. The theoretical understanding from Section 2 motivates straightforward adaptations to model-based forests such that the procedure can also be applied to observational studies. The flexibility of its implementation in **model4you** allows to define and evaluate blended estimation approaches transferring the concept of local centering from causal forests to model-based forests. Along with these new algorithms, we propose a set of five research questions which we investigate empirically in Section 4. An overview of the questions is given in Table 1. We begin with the standard implementations of causal forests (cf) and model-based (mob) forests without centering.

RQ 1 How do cf and mob, as implemented in the two R add-on packages **grf** (for cf) and **model4you** (for mob), compare to each other in randomized and observational settings?

After addressing RQ 1, the question remains if and to what extent local centering inherent in cf leads to more robustness against confounding effects. To answer that we will incorporate orthogonalization in mob as explained in the following. Causal forests apply local centering to both the outcome Y and treatment indicator w , and mob do not center locally at all. To bring cf and mob closer, we define a method which applies mob to the model

$$\mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}, W = w) = \hat{\mu}(\mathbf{x}) + \bar{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x})),$$

i.e. after centering the treatment indicator w and the outcome Y . By using $\bar{\mu}(\mathbf{x})$ instead of $\mu(\mathbf{x})$, we emphasize that $\bar{\mu}(\mathbf{x})$ is now the prognostic effect for the *centered* Y .

The rationale is to estimate the marginal mean and propensities $\pi(\mathbf{x})$ as in cf first and then apply mob to the centered treatment $w - \hat{\pi}(\mathbf{x})$ and centered outcome $Y - \hat{\mu}(\mathbf{x})$ to obtain the prognostic and predictive effect. We call this approach $\text{mob}(\hat{W}, \hat{Y})$. The bivariate score function for mob is changed from (6) to

$$s_{\text{mob}(\hat{W}, \hat{Y})}(\hat{\mu}, \hat{\tau}) := (Y - \hat{\mu}(\mathbf{x}) - \hat{\tau}(w - \hat{\pi}(\mathbf{x}))(1, w - \hat{\pi}(\mathbf{x}))^\top.$$

RQ	Question	Methods	Linear predictors
1	Comparison of causal forests and model-based forests	cf	$\hat{\eta}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$
		mob	$\hat{\mu}(\mathbf{x}) + \tau(\mathbf{x})w$
2	Effect of splitting only in $\tau(\mathbf{x})$ vs. in $\tau(\mathbf{x})$ and $\hat{\mu}(\mathbf{x})$	mobcf	$\hat{\eta}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$
		mob(\hat{W}, \hat{Y})	$\hat{\eta}(\mathbf{x}) + \hat{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$
3	Comparison of causal forests implemented in grf vs. model4you	cf	$\hat{\eta}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$
		mobcf	$\hat{\eta}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$
4	Effect of locally centering W in model-based forests	mob(\hat{W})	$\hat{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$
		mob	$\hat{\mu}(\mathbf{x}) + \tau(\mathbf{x})w$
5	Effect of additionally centering Y in model-based forests centering W	mobcf	$\hat{\eta}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$
		mob(\hat{W}, \hat{Y})	$\hat{\eta}(\mathbf{x}) + \hat{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$
		mob(\hat{W})	$\hat{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$

Table 1: Overview of research questions

In cases where local centering of Y effectively regresses out the effect of \mathbf{X} on Y , $\hat{\mu}(\mathbf{x})$ will be close to 0. Since removing $\hat{\mu}$ leads to the conditional mean function underlying cf

$$\mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}, W = w) = \hat{\eta}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x})),$$

we call this version ‘‘mobcf’’. Both the outcome and the treatment indicator are centered and only splitting with respect to scores corresponding to the treatment effect τ is performed, while intercept scores are ignored in this process. The only difference between mobcf and mob(\hat{W}, \hat{Y}) is that simultaneous splitting in both the intercept and treatment effect parameters is performed by the latter, whereas the intercept is ignored in the former.

RQ 2 How does mob(\hat{W}, \hat{Y}) perform compared to mobcf?

The mobcf approach helps us to directly compare the different more technical aspects, such as variable and split point selection or stopping criteria, of tree induction implemented in **grf** and **model4you**, because it can be seen as a re-implementation of cf using the computational infrastructure of the **model4you** package.

RQ 3 How does mobcf perform compared to cf implemented in **grf**?

Centering the response is straightforward under L_2 loss but more difficult under other forms of the likelihood as discussed in Section 2.5. The questions arise if and to what extent solely centering of the treatment indicator w already improves the estimation accuracy in observational settings. To answer that we define a ‘‘hybrid approach’’ mob(\hat{W}) that applies mob to models parameterized by $\hat{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$, *i.e.* after solely centering the w but not the outcome Y . The score function for mob is changed from (6) to

$$s_{\text{mob}(\hat{W})}(\hat{\mu}, \hat{\tau}) := (Y - \hat{\mu} - \hat{\tau}(w - \hat{\pi}(\mathbf{x}))(1, w - \hat{\pi}(\mathbf{x}))^\top.$$

RQ 4 How does solely centering of the treatment indicator ($\text{mob}(\hat{W})$) influence the performance of mob without centering in settings with confounding?

The final research question is whether additional outcome centering improves upon a forest with treatment centering and simultaneous splits in prognostic and predictive effects as implemented by $\text{mob}(\hat{W})$.

RQ 5 How does $\text{mob}(\hat{W})$ perform compared to mob that center both treatment and outcome (mobcf , and $\text{mob}(\hat{W}, \hat{Y})$)?

4. Empirical evaluation

In this section, we provide answers to the research questions defined in Section 3 by evaluating the performance of cf and mob as well as the different blended versions in a simulation study for normal outcomes, different predictive and prognostic effects, and a varying number of observations and covariates. The reference implementations in the **grf** and **model4you** R add-on packages were used for the original cf and mob algorithms. Moreover, the blended approaches from Section 3 are implemented using **model4you**, *i.e.* by fitting model-based forests after centering of treatment indicators ($\text{mob}(\hat{W})$) and additionally of outcomes ($\text{mob}(\hat{W}, \hat{Y})$ and mobcf , with and without explicitly accounting for μ , respectively).

4.1. Data-generating process

The comparison is based on the study settings of Nie and Wager (2021). The authors proposed four study settings - referred to as Setups A, B, C and D. For Setup A, explanatory variables were sampled by $\mathbf{X} \sim U([0, 1]^P)$ and for the other three setups they used $\mathbf{X} \sim N(0, \mathbf{1}_{P \times P})$ - with $P = \{10, 20\}$ (5 informative and $P - 5$ noise variables). Treatment was sampled by $W \mid \mathbf{X} = \mathbf{x} \sim B(1, \pi(\mathbf{x}))$ with propensity function $\pi(\mathbf{x})$ that varied among the four considered setups:

$$\pi(\mathbf{x}) = \begin{cases} \pi_A(x_1, x_2) = \max\{0.1, \min\{\sin(\pi x_1 x_2), 1 - 0.1\}\} \\ \pi_B \equiv 0.5 \\ \pi_C(x_2, x_3) = 1/(1 + \exp(x_2 + x_3)) \\ \pi_D(x_1, x_2) = 1/(1 + \exp(-x_1) + \exp(-x_2)). \end{cases}$$

For Setup B, probability $\pi \equiv 0.5$ referred to a randomized study. The conditional average treatment effect function for each setup was given as

$$\tau(\mathbf{x}) = \begin{cases} \tau_A(x_1, x_2) = (x_1 + x_2)/2 \\ \tau_B(x_1, x_2) = x_1 + \log(1 + \exp(x_2)) \\ \tau_C \equiv 1 \\ \tau_D(x_1, x_2, x_3, x_4, x_5) = \max\{x_1 + x_2 + x_3, 0\} - \max\{x_4 + x_5, 0\}. \end{cases}$$

For Setup C, the treatment effect was constant. The prognostic effects were defined as

$$\mu(\mathbf{x}) = \begin{cases} \mu_A(x_1, x_2, x_3, x_4, x_5) = \sin(\pi x_1 x_2) + 2(x_3 - 0.5)^2 + x_4 + 0.5x_5 \\ \mu_B(x_1, x_2, x_3, x_4, x_5) = \max\{x_1 + x_2, x_3, 0\} + \max\{x_4 + x_5, 0\} \\ \mu_C(x_1, x_2, x_3) = 2 \log(1 + \exp(x_1 + x_2 + x_3)) \\ \mu_D(x_1, x_2, x_3, x_4, x_5) = (\max\{x_1 + x_2 + x_3, 0\} + \max\{x_4 + x_5, 0\})/2. \end{cases}$$

Overall, Setup A has complicated confounding that needs to be overcome before a relatively simple treatment effect function $\tau(\mathbf{x})$ can be estimated. In Setup B, it is possible to accurately estimate τ without explicitly controlling for confounding. Setup C has strong confounding but the propensity score function is easier to estimate than the prognostic effect while the treatment effect is constant. In Setup D, the treatment and control arms are unrelated, in the sense that $\mathbb{E}[Y | \mathbf{X}, W = 1]$ and $\mathbb{E}[Y | \mathbf{X}, W = 0]$ are uncorrelated and there is no benefit to jointly learn them.

As in Nie and Wager (2021), we studied a normal linear regression model

$$(Y | \mathbf{X} = \mathbf{x}, W = w) \sim N(\mu(\mathbf{x}) + \tau(\mathbf{x})(w - 0.5), 1),$$

where half of the predictive effect was added to the prognostic effect.

All procedures were applied to 100 learning samples of size $N \in \{800, 1600\}$ and number of explanatory variables $P \in \{10, 20\}$. In order to minimize the impact of different implementation details, cf, mob and the blended versions were grown with the same hyperparameter options, see Section 7. Propensities $\pi(\mathbf{x})$ and means $m(\mathbf{x})$ were estimated by **grf** regression forests for local centering in all forest variants. For the causal forest, the outcome was always centered by $\hat{m}(\mathbf{x})$. In case of randomized data (Setup B), the treatment indicator was centered by $\pi \equiv 0.5$, in all other settings, estimated propensities $\hat{\pi}(\mathbf{x})$ were used.

Performance was assessed by the ability of the methods to estimate the predictive effect $\tau(\mathbf{x})$. The mean squared error $\mathbb{E}_{\mathbf{X}}\{(\hat{\tau}(\mathbf{X}) - \tau(\mathbf{X}))^2\}$, evaluated on a test sample of size 1000, was used to compare the predictive performance of all candidate models in the 16 different scenarios. The results are shown in Figure 1.

The results were also analyzed statistically by means of a normal linear mixed model with log-link, explaining the estimated mean squared error for $\hat{\tau}(\mathbf{x})$ by a four-way interaction of data generating process, sample size N , dimension P , and random forest variant. We estimated the mean squared error ratios between cf and mob (RQ 1), between mobcf and mob(\hat{W}, \hat{Y}) (RQ 2), between cf and the mobcf approach (RQ 3), between mob with centered W (mob(\hat{W})) and without (mob) (RQ 4), and between mob(\hat{W}) and mobcf or mob(\hat{W}, \hat{Y}) (RQ 5). For each simulation run, the model featured a corresponding random intercept reflecting the paired simulation design. Simultaneous 95% confidence intervals for the mean squared error ratios are presented along with the estimates. For example, the ratio of the mean squared errors of cf and mob in the first line of Table 2 was 0.663 with confidence interval (0.596, 0.738). This is in line with the performance error of cf being at least 59.6% and at most 73.8% of the performance error of mob, with 66.3% denoting the estimate. Bold, italic and normal fonts are used to indicate superior, inferior, and equivalent prediction performance.

4.2. Results

The results for adaptive forests are presented in Figure 1. In Section 2 of the Supplementary Material A, we report on the effect of honesty on predictive error as well as the mean squared differences in performance to cf for the adaptive and honest versions (Figures S. 1 and S. 2). The statistical analysis of the results is given in Table 2 for the adaptive version of forests and in Table S. 1 of the Supplementary Material A for the honest version.

RQ 1. mob vs. cf In all setups, cf outperformed mob. Especially in Setup C, mob was unable to overcome the strong confounding effect and therefore did not provide accurate

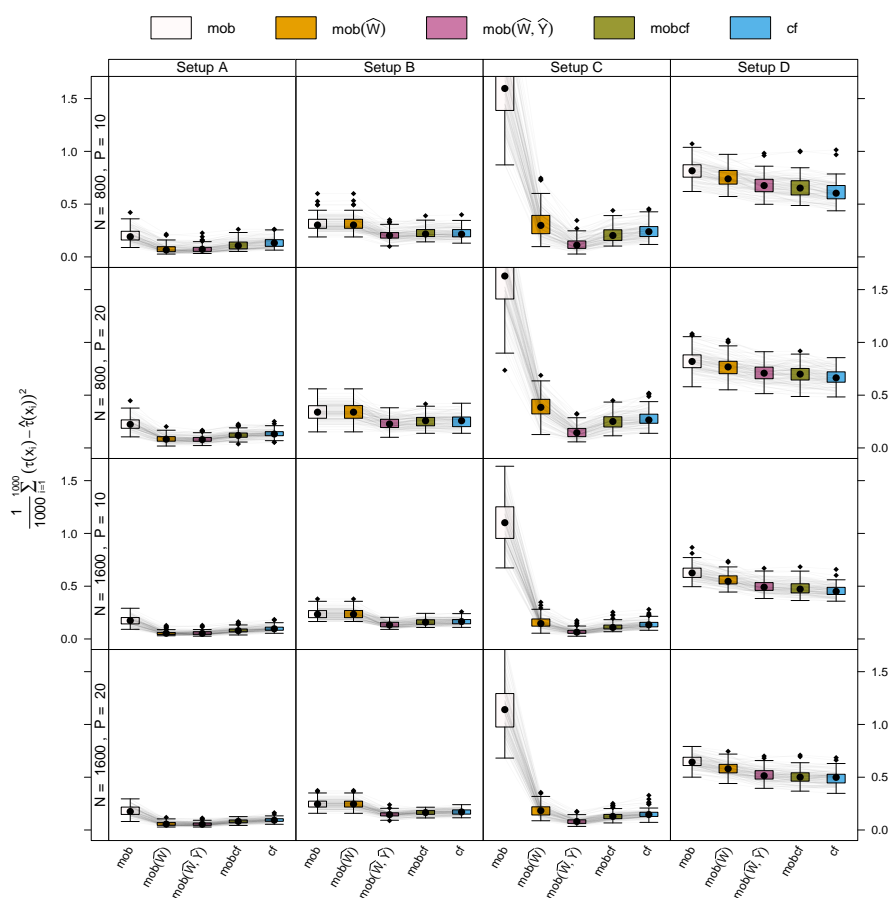


Figure 1: Results for the experimental setups 4.1. Direct comparison of the adaptive versions of causal forests (cf), model-based forests without centering (mob), mob imitating causal forests (mobcf), mob with centered W ($mob(\hat{W})$) and additional of Y ($mob(\hat{W}, \hat{Y})$).

Forest-Based Heterogeneous Treatment Effect Estimators

14

DGP	N	P	Mean squared error ratio									
			(RQ 1)		(RQ 2)		(RQ 3)		(RQ 4)		(RQ 5)	
			cf vs. mob	mobcf vs. mob(W, Y)	cf vs. mobcf	mob(W) vs. mob	mob(W) vs. mobcf	mob(W) vs. mob	mob(W) vs. mobcf	mob(W) vs. mob(W, Y)	mob(W) vs. mob(W, Y)	
Setup A	800	10	<i>0.663 (0.596, 0.738)</i>	1.446 (1.206, 1.734)	1.165 (1.017, 1.335)	<i>0.392 (0.334, 0.461)</i>	<i>0.689 (0.574, 0.826)</i>	<i>0.996 (0.806, 1.230)</i>				
	1600	10	<i>0.596 (0.537, 0.662)</i>	1.465 (1.228, 1.747)	1.111 (0.972, 1.270)	<i>0.385 (0.331, 0.446)</i>	<i>0.777 (0.604, 0.850)</i>	1.050 (0.859, 1.284)				
	1600	20	<i>0.575 (0.499, 0.663)</i>	1.458 (1.123, 1.893)	1.201 (0.991, 1.485)	<i>0.324 (0.258, 0.408)</i>	<i>0.677 (0.521, 0.881)</i>	0.988 (0.727, 1.342)				
Setup B	800	10	<i>0.517 (0.447, 0.598)</i>	1.453 (1.117, 1.889)	1.150 (0.944, 1.401)	<i>0.328 (0.265, 0.407)</i>	<i>0.730 (0.567, 0.940)</i>	1.061 (0.788, 1.438)				
	1600	10	<i>0.707 (0.662, 0.756)</i>	1.099 (1.015, 1.190)	0.987 (0.914, 1.065)	1.000 (0.947, 1.056)	1.395 (1.306, 1.491)	1.533 (1.429, 1.646)				
	1600	20	<i>0.745 (0.701, 0.791)</i>	1.093 (1.018, 1.174)	1.001 (0.935, 1.071)	1.000 (0.951, 1.052)	1.345 (1.266, 1.428)	1.470 (1.380, 1.567)				
Setup C	800	10	<i>0.695 (0.635, 0.762)</i>	1.166 (1.036, 1.313)	1.034 (0.929, 1.152)	1.000 (0.929, 1.076)	1.487 (1.355, 1.633)	1.734 (1.563, 1.924)				
	1600	10	<i>0.683 (0.625, 0.746)</i>	1.110 (0.992, 1.243)	1.037 (0.934, 1.152)	1.000 (0.932, 1.073)	1.518 (1.387, 1.662)	1.686 (1.529, 1.859)				
	1600	20	<i>0.148 (0.141, 0.156)</i>	1.693 (1.514, 1.893)	1.150 (1.067, 1.240)	<i>0.197 (0.190, 0.205)</i>	1.529 (1.429, 1.636)	2.589 (2.335, 2.870)				
Setup D	800	10	<i>0.170 (0.162, 0.177)</i>	1.673 (1.520, 1.841)	1.123 (1.051, 1.199)	<i>0.236 (0.229, 0.244)</i>	1.563 (1.474, 1.657)	2.615 (2.395, 2.856)				
	1600	10	<i>0.124 (0.113, 0.136)</i>	1.651 (1.348, 2.023)	1.184 (1.032, 1.359)	<i>0.143 (0.132, 0.155)</i>	1.368 (1.201, 1.558)	2.258 (1.868, 2.731)				
	1600	20	<i>0.131 (0.121, 0.142)</i>	1.573 (1.320, 1.875)	1.166 (1.030, 1.320)	<i>0.163 (0.153, 0.174)</i>	1.452 (1.295, 1.628)	2.284 (1.943, 2.684)				
Setup D	800	10	<i>0.756 (0.737, 0.775)</i>	<i>0.970 (0.945, 0.996)</i>	<i>0.934 (0.909, 0.960)</i>	<i>0.917 (0.897, 0.938)</i>	1.133 (1.105, 1.162)	1.099 (1.072, 1.127)				
	1600	10	<i>0.807 (0.788, 0.826)</i>	0.983 (0.939, 1.008)	<i>0.955 (0.933, 0.982)</i>	<i>0.926 (0.906, 0.947)</i>	1.100 (1.074, 1.126)	1.081 (1.056, 1.107)				
	1600	20	<i>0.720 (0.696, 0.744)</i>	0.970 (0.936, 1.005)	<i>0.939 (0.904, 0.974)</i>	<i>0.886 (0.859, 0.912)</i>	1.155 (1.116, 1.194)	1.120 (1.083, 1.157)				
			<i>0.763 (0.739, 0.787)</i>	0.967 (0.935, 1.001)	<i>0.982 (0.949, 1.018)</i>	<i>0.894 (0.869, 0.920)</i>	1.151 (1.114, 1.189)	1.113 (1.078, 1.149)				

Table 2: Results for the experimental setups 4.1 for the *adaptive* versions of the methods. Comparison of mean squared errors for $\hat{f}(\mathbf{x})$ in the different scenarios. Estimates and simultaneous 95% confidence intervals were obtained from a normal linear mixed model with log-link. Cells printed in bold font correspond to a superior reference (mob in the first and fourth columns, mob(W, Y) in the second column, mobcf in the third column and mob(W) in the last column), cells printed in italics indicate an inferior reference.

estimates for the (constant) treatment effect.

RQ 2. $\text{mob}(\hat{W}, \hat{Y})$ vs. mobcf The $\text{mob}(\hat{W}, \hat{Y})$ approach performed better than the mobcf approach in almost all scenarios except for Setup D. (However, uncorrelated treatment and control arms rarely occur in reality. All methods had a higher MSE than in the other setups.) These performance differences suggest that splitting by treatment *and* prognostic effect is beneficial.

RQ 3. mobcf vs. cf Despite the fundamentally different internal splitting and stopping criteria, the original implementation of cf from package `grf` had very similar performance to our re-implementation mobcf from package `model4you` in Setup A and B. In Setup C with strong confounding, the mobcf approach performed slightly better than cf , while in Setup D cf performed slightly better.

RQ 4. $\text{mob}(\hat{W})$ vs. mob In case of confounding (Setup A, C), local centering of W ($\text{mob}(\hat{W})$) significantly improved the performance of mob . In Setup B without confounding, both approaches performed equally since $\text{mob}(\hat{W})$ is equal to mob applied to $w = 0.5$.

RQ 5. Methods centering the outcome (mobcf , $\text{mobmob}(\hat{W}, \hat{Y})$) vs. $\text{mob}(\hat{W})$ By centering the outcome Y in addition to the treatment W , $\text{mob}(\hat{W}, \hat{Y})$ and mobcf performed better than $\text{mob}(\hat{W})$ except for Setup A – centering the outcome did not further improve the results. The improvements by additionally centering Y were relatively small for mob compared to the improvements due to centering the treatment W (see RQ 4).

Overall, our results reveal treatment effect centering ($\text{mob}(\hat{W})$) as the most relevant ingredient to random forests for HTE estimation in observational studies. If possible, additional centering Y in combination with simultaneous estimation of predictive and prognostic effects ($\text{mob}(\hat{W}, \hat{Y})$) is recommended.

5. Effect of cesarean section on postpartum blood loss

In this section, we discuss random forest-based HTEs expressing the additional amount of blood loss explained by prepartum variables, comparing cesarean sections with vaginal deliveries. We analyze data from 1309 women who participated in a prospective study conducted from October 2015 to November 2016 at the University Hospital Zurich (details and data are available from [Haslinger, Korte, Hothorn, Brun, Greenberg, and Zimmermann 2020](#)). The outcome is defined as measured blood loss (MBL) in mL and the authors ensured application of a standardized measurement procedure for all study participants ([Kahr, Brun, Zimmermann, Franke, and Haslinger 2018](#)). For our study, we removed one outlier observation with a blood loss of 5700 mL and eight observations with missing values for BMI so that a sample of size $N = 1300$ remains. MBL was recorded as an interval-censored variable, because it is impossible to exactly determine the amount of blood loss in the sometimes hectic environment of a delivery ward ([Kahr et al. 2018](#)). Potential inaccuracies in the measuring process are represented by an interval width of 50 mL for blood losses ≤ 1 L and an interval width of 100 mL when the mother lost more than one liter of blood. Measured blood loss can a priori be

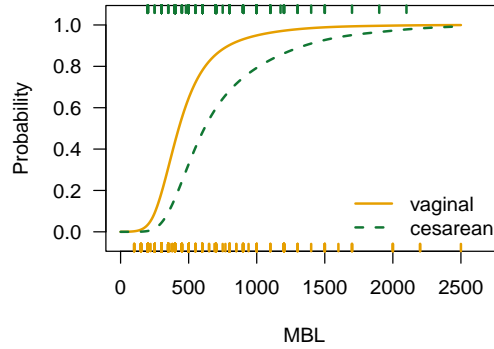


Figure 2: Marginal distribution of measured blood loss (mL) for cesarean section and vaginal delivery. Rugs indicate measured blood loss observations.

considered a positive real and right-skewed variable (Figure 2). Table 3 gives a summary of the eight considered prepartum characteristics ($P = 8$).

Variable	Description	Range
GA	Gestational age	177–297 (days)
AGE	Maternal age	18–48 (years)
MULTIPAR	Multiparity	no/yes
BMI	Body mass index	15.4–66
MULTIFET	Multifetal pregnancy	no/yes
NW	Neonatal weight	360–4630 (g)
IOL	Induction of labor	no/yes
AIS	Chorioamnionitis	no/yes

Table 3: Prepartum characteristics

As the outcome variable MBL is skewed and interval-censored not all assumptions for causal forests are fulfilled as they estimate a conditional mean of some continuous outcome optimizing L_2 risk. The extensibility of model-based forests discussed in Section 2.5 allows us to take into account the structural assumptions of MBL by substituting ℓ_{mob} in (5) with the negative log-likelihood of a more appropriate model. We set up a model-based transformation forest with treatment centering by combining the $\text{mob}(\hat{W})$ approach using local centering of the treatment indicator within a transformation model.

5.1. Transformation base model

The reasoning in Section 2 is based on the normal linear model (4) and its corresponding likelihood (5) for absolutely continuous observations. While the latter can easily be adapted to interval-censored observations, more effort is needed for allowing skewness in the response distribution. Adopting a standard normal distribution for the error term Z like in Section 2.5,

model (4) can be written as a conditional distribution function

$$P(\text{MBL} \leq y \mid \mathbf{X} = \mathbf{x}, W = w) = \Phi\left(\frac{y - \mu(\mathbf{x}) - \tau(\mathbf{x})w}{\sigma}\right).$$

In this model, symmetry is achieved by a linear transformation of the y argument on the probit scale. Replacement of this linear transformation by a potentially nonlinear one gives rise to transformation models. In combination with the probit link, this model is a Box-Cox-type linear regression model that transforms the skewed outcome variable to normality. Instead of using the traditional Box-Cox power transformation, we estimate a suitable transformation of MBL by means of a flexible polynomial in Bernstein form (Hothorn, Möst, and Bühlmann 2018). Ignoring covariates and the local centering of W for a moment, our transformation model describes the conditional distribution of the positive skewed real variable MBL using mode of delivery W as treatment indicator for vaginal delivery ($W = 0$) vs. cesarean section ($W = 1$):

$$P(\text{MBL} \leq y \mid W = w) = \Phi(h(y) - \mu - \tau w).$$

Deviations from normality are captured by the nonlinear transformation function h in this model. Because the transformation function h contains an intercept term, the parameter μ is not identified. We thus estimate the transformation base model under the constraint $\mu \equiv 0$. The intercept function h varies with the chosen MBL cut-off y and is smooth and monotonically increasing; a polynomial in Bernstein form of order six was used to parameterize this function. The parameter $\tau = \mathbb{E}(h(Y(1)) - h(Y(0)))$ is not identical to an average treatment effect on the untransformed scale which could be interpreted directly in terms of the original units of the outcome (here blood loss in mL). Nevertheless, τ in our transformation model has an intuitive interpretation corresponding to Cohen's d : the units of the treatment effect correspond to standard deviations under the normal model.

The parameters of the transformation base model were estimated by minimization of the negative log-likelihood for an interval-censored observation $(y, \bar{y}]$

$$\begin{aligned} \ell_{\text{Trafo}}(\mu, \tau, \boldsymbol{\vartheta}) &= -\log(P(y < Y \leq \bar{y} \mid W = w)) \\ &= -\log(\Phi(h(\bar{y} \mid \boldsymbol{\vartheta}) - \mu - \tau w) - \Phi(h(y \mid \boldsymbol{\vartheta}) - \mu - \tau w)) \end{aligned}$$

where all parameters, including $\boldsymbol{\vartheta}$ for the transformation function, are estimated in each node. A parameterisation of h in terms of a polynomial in Bernstein form $h(\cdot \mid \boldsymbol{\vartheta})$ ensures uniform convergence to any continuous unknown transformation function h on some interval by Weierstrass' approximation theorem (Farouki 2012).

5.2. Personalized transformation model

The results of Section 2–4 motivate the application of model-based forests to a Box-Cox type transformation model for the estimation of HTEs of cesarean sections on PPH. The transformation base model provides skewness and interval-censoring, whereas the locally centered treatment indicator controls for potential confounding. In more detail, we used a $\text{mob}(\hat{W})$ forest in combination with the transformation base model, *i.e.* with local centered treatment indicator \hat{w} , to compute personalized treatment effects $\tau(\mathbf{x})$ and prognostic effects $\mu(\mathbf{x})$ of the model

$$P(\text{MBL} \leq y \mid \mathbf{X} = \mathbf{x}, W = w) = \Phi(h(y) - \mu(\mathbf{x}) - \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))). \quad (7)$$

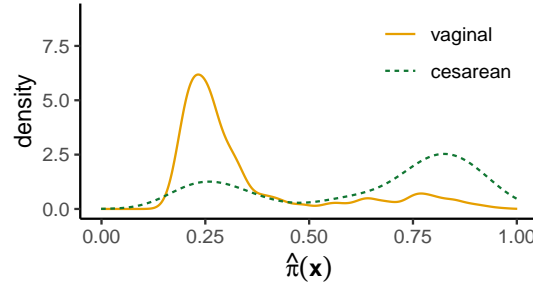


Figure 3: Estimates of propensity scores $\pi(\mathbf{x})$ returned by the regression forest for orthogonalization of the treatment indicator

As in the simulation study, a regression forest was applied to estimate propensities $\pi(\mathbf{x})$. We only used locally centered propensities because the empirical results of Section 4 showed that centering W was the main driver for good performance in observational settings. Furthermore, while centering W is straightforward for the transformation model at hand, implementing centering on the outcome Y is less clear.

Figure 3 shows that the distribution functions of $\hat{\pi}(\mathbf{x})$ for each treatment group greatly differ. This indicates that prepartum characteristics indeed influence the mode of delivery and that the treated and control group are dissimilar with respect to these characteristics.

We first fitted the transformation base model without covariates but with propensity-centered mode of delivery to estimate a constant effect adjusted for potential confounding. The corresponding effect $\hat{\tau}$, *i.e.* the marginal Cohen’s d , was 0.823 ($CI_{0.95} = (0.686, 0.959)$), indicating that women giving birth by cesarean section have a higher postpartum blood loss compared to women giving birth by vaginal delivery.

The model-based transformation forest was fitted with the same hyperparameter settings as in the simulation study (Section 7). We did not adjust the hyperparameters because random forests have been shown to be insensitive to hyperparameter changes (Probst, Boulesteix, and Bischl 2021). Figure S. 3 in the Supplementary Material A demonstrates this for the `mtry` parameter – the number of chosen variables per split. We only analysed the `mtry` parameter since Probst, Wright, and Boulesteix (2019) found that the “`mtry` parameter is most influential [...]” while “[s]ample size and node size have a minor influence on the performance [...]”.

Figure 4 depicts the distribution of the estimated out-of-bag (OOB) heterogeneous treatment effects $\hat{\tau}(\mathbf{x})$ of cesarean section compared to vaginal delivery. The distribution is unimodal and slightly left-skewed. For almost all births, a cesarean section increases the risk for higher blood losses compared to vaginal delivery. For comparison, the average treatment effect of $\hat{\tau} = 0.823$ of the transformation base model is included.

The interval-censored negative log-likelihood of the transformation base model was 3613.972. The model-based transformation forest improved upon this, yielding a likelihood of 3413.989 (estimated in-bag to make it comparable to the transformation base model).

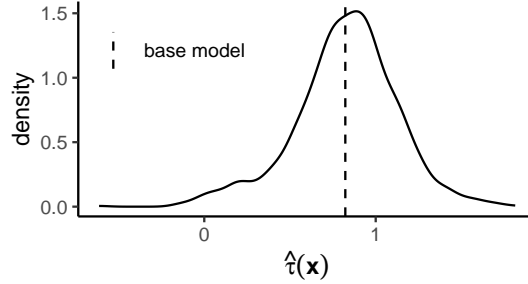


Figure 4: Kernel density estimates of the personalized treatment estimates of the model-based transformation forest. The dashed line presents the estimated effect of the transformation base model.

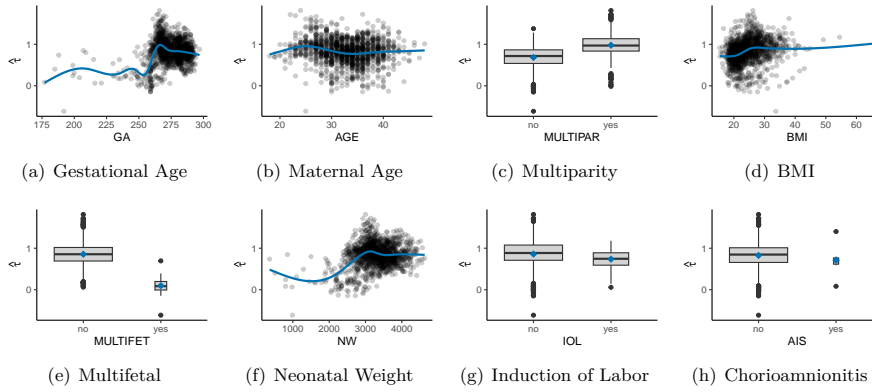


Figure 5: Dependency plots of the individual treatment effects calculated by the model-based transformation forest. Values $\hat{\tau} > 0$ mean that cesarean section increases the blood loss compared to vaginal delivery. Lines and diamond points depict (smooth conditional) mean effects.

5.3. Dependence plots

The dependency of the treatment effect τ on the prepartum variables is visualized by dependence plots (Figure 5). Scatter plots are used for continuous covariates and boxplots for categorical covariates. We also provide mean effects per group for categorical covariates and the smooth conditional mean effect function for continuous covariates. The latter was estimated by a generalized additive model (GAM) with a single smooth term depending on the considered variable. Births with higher gestational age, higher neonatal weight and singleton pregnancy have a higher risk for elevated blood loss due to cesarean section compared to vaginal delivery. The effect differences were most pronounced between multifetal and singleton births. For multifetal pregnancies, treatment effects are closer to 0 than for singleton preg-

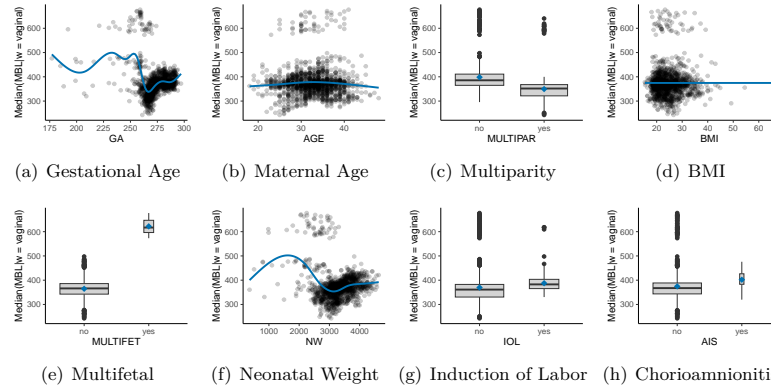


Figure 6: Dependency plots of median measured blood losses calculated by the model-based transformation forest. Higher values mean higher blood loss. Lines and diamond points depict (smooth conditional) means.

nancies. For a very premature multifetal birth (gestational age of 192 days) of a 25-year-old mother with an elevated BMI of 33.7, a cesarean section was determined to be most effective ($\hat{\tau} = -0.614$). Because the distribution of the gestational age (GA) is left-skewed, the curve of the smoothed conditional mean effects is somewhat erratic. It might also indicate that GA was often used as a splitting variable. While interpreting these results, it should be noted that violations of the unconfoundedness assumption do not seem implausible.

5.4. Model interpretation and communication

Interpretation and risk communication in terms of predicted $\hat{\tau}(\mathbf{x})$ is difficult because the effect is defined by Cohen's d on a transformed latent normal scale in model (7). However, the model allows conditional quantiles to be computed and thus information about the conditional MBL distribution for given prepartum covariates and propensities $\hat{\pi}(\mathbf{x})$ can be expressed on the quantile scale for both modes of delivery.

To assess the prognostic effects on MBL, we computed median measured blood losses for $W = 0$ (vaginal delivery) given the covariates and propensities. Figure 6 indicates that a gestational age of about 270 days, a birth weight around 3050 g and singleton births are associated with small median postpartum blood losses for vaginal deliveries.

The predictive effect of a cesarean section on MBL in such a low-risk group can be communicated by comparing the MBL distributions under vaginal delivery and cesarean section. The median blood loss for a hypothetical woman in this low-risk group (aged 32.7 years with a BMI of 24.7, the mean values in the study population) is predicted to increase from 329 mL (vaginal delivery, 80% prediction interval 209–507 mL) to 470 mL (cesarean section, 80% prediction interval 305–817 mL) by our model. The asymmetric prediction intervals reflect skewness in the MBL distribution and the wider interval for a cesarean section suggests variance heterogeneity is captured by the model. The risk of PPH (defined by the 500 mL cut-off) is small for vaginal deliveries but substantial under a cesarean section.

6. Discussion and outlook

6.1. Effects of cesarean sections of postpartum blood loss

The lives of many of us have been, or will be, impacted by a cesarean section directly or indirectly. Empowering women for making an informed decision, especially in an elective setting, crucially relies on evidence about the short- and long-term consequences for them and their children (Antoine and Young 2021). Providing an estimate of the individual predicted excess blood loss caused by a cesarean section, in comparison to a vaginal delivery, to pregnant women and their obstetricians not only offers the possibility to decide based on a personalized risk assessment, but has also the potential to help the overarching goal of reducing the prevalence of cesarean sections. The question to perform a cesarean section or not is less imminent in women with obvious risk factors which make a cesarean section inevitable (*e.g.* prematurity and multiple fetus pregnancy), but is of utmost clinical interest in women with a prepartum low-risk profile (singleton pregnancy at term with normal fetal weight estimation). To the best of our knowledge, this is the first study to predict excess postpartum blood loss in low-risk women. Our approach of modeling the continuous blood loss distribution for arbitrary cut-off values is also unique in the sense that published prognostic models provide risk estimates for events $MBL > 500$ mL, or other prespecified cut-off values, only.

Our results were estimated based on data originating from a prospective study employing a standardized and validated assessment of blood loss under both modes of delivery. Such efforts can only be successfully implemented in a controlled setting and hardly apply to retrospective collections of routine clinical data from multiple study centers. However, the detection of smaller but still relevant patterns in HTEs might require more information than available from the $N = 1300$ study participants. The random forest methodology would allow differentiation between planned and unplanned cesarean sections (Section 2.3) in a single model, however, the sample sizes in the present study seem too limited for such an analysis. It remains to be seen if refined analyses of large-scale routine clinical data will provide results similar to those reported here.

6.2. Forest-based HTE estimation

From a statistical perspective, estimating heterogeneous treatment effects (HTEs) is a difficult task, both when data from randomized trials and observational studies are analyzed. Based on a common theoretical understanding of two strands of random forest algorithms for HTE estimation, we hypothesized that centering the treatment with corresponding propensities helps to address confounding. The empirical results suggest that this simple modification of the data is instrumental for the analysis of observational and thus potentially confounded data.

Centering the outcome is equally simple in models for conditional means, but may be much harder in other models. Empirically, we found that the combination of centered treatment and simultaneous split selection (with respect to both prognostic and predictive effects) performed at least as well as explicit outcome centering. This may seem surprising from a theoretical point of view, because a nuisance parameter is dealt with in two completely different ways. Even more interesting is the overall strong performance of a variant employing both principles

at the same time: The $\text{mob}(\hat{W}, \hat{Y})$ forest is grown on centered outcomes and treatments and additionally also splits nodes with respect to both prognostic and predictive effects, leading to a performance at least as well as the best-performing competitor. Other aspects of tree and forest induction, such as exhaustive search versus association tests for variable selection, internal stopping criteria based on sample-size constraints etc., did not explain much variability in performance.

Based on our current theoretical and empirical understanding of the elements of both model-based and causal random forests for HTE estimation, we can make the following recommendations for their application in practice – especially when the conditional mean of a numeric outcome captures all relevant aspects: Data from randomized trials can be analyzed by causal forests (with outcome centering and known treatment probability π for treatment indicator centering) or model-based forests (with or without outcome centering) under the intention-to-treat principle. Under potential confounding, it is important to accurately model treatment propensities as in causal forests (with outcome and treatment centering). When combined with treatment centering, model-based forests will lead to approximately the same results. Additionally centering the outcome may even offer a small performance gain compared to standard causal forests.

The empirical performances reported in Section 4 coupled with established asymptotic results for causal random forests with treatment centering (Athey *et al.* 2019) and the benign asymptotic behavior of other ingredients, such as transformation models (Hothorn *et al.* 2018) or uniform convergence of polynomials in Bernstein form, suggests favorable asymptotic properties for special flavors of model-based forests. We leave the presentation of formal results to future work.

6.3. Outlook

The blending of model-based and causal forests discussed here seems to be a promising approach for HTE estimation beyond mean regression. Under potential confounding with binary, ordinal, count, or survival outcomes, it is easy to combine model-based forests with treatment centering ($\text{mob}(\hat{W})$) following the path outlined in Section 2.5. For example, for a binary outcome $Y \in \{0, 1\}$ a logistic regression-based causal forest can estimate models of the form

$$\text{logit}(\text{P}(Y = 1 \mid \mathbf{X} = \mathbf{x}, W = w)) = \mu(\mathbf{x}) + \tau(\mathbf{x})w.$$

The HTE $\tau(\mathbf{x})$ can then be interpreted as a covariate-dependent log-odds ratio. In practice, this model can be estimated by package **model4you**, with appropriate treatment centering being the only modification necessary (under the usual assumptions, of course). We leave an in-depth analysis and evaluation of this principle to future research which should also address the question of how to achieve outcome centering in such models similar to $\text{mob}(\hat{Y}, \hat{W})$.

Finally, going beyond these recommendations and insights, our results are interesting from two further perspectives. First, the empirical application to postpartum blood loss in Section 5 has shown that blended model-based causal forests can be tailored to specific setups by adapting the underlying loss function. Second, we empirically demonstrated that two independent implementations of random forests for HTE estimation performed akin in comparable settings. This form of external software validation is important in its own right because the underlying algorithms and implementations are rather complex, and external validity can only be assessed with the help of an independent implementation. In case of **grf** and **model4you**, past, current,

and future users of these software packages can have higher confidence in HTEs estimated using either package.

7. Computational details

All computations were performed using R version 4.1.1 (R Core Team 2021), with the following add-on packages: **grf** (Tibshirani *et al.* 2021), **model4you** (Seibold, Zeileis, and Hothorn 2021), **trtf** (Hothorn 2021), and **partykit** (Hothorn and Zeileis 2015, 2021a).

In all empirical experiments, both causal forests and all variants of model-based forests were grown with $M = 500$ trees (**model4you::pmforest** default) with minimum node size of `node = 14`, number of chosen variables per split `mtry = P` and subsampling (the latter two being **causal_forest** defaults for $P = 10, 20$). We chose a minimum node size of 14 because the default of **partykit::ctree_control** (which **model4you** is based on) is 7 but we require this minimum node size for each of the two treatment groups. For adaptive forests 50 % of data were used to build each tree and for honest forests subsamples were further cut in half (25 % to determine splits, 25 % for estimation, all **grf** defaults). To implement local centering of W in case of randomized data for causal forests, we set `W.hat` to 0.5 within **grf::causal_forest**.

We used the transformation forest implementation of the **trtf** package (Hothorn 2021; Hothorn and Zeileis 2021b) for fitting the transformation-based forest in Section 5.

Ratios and confidence intervals presented in Table 2 and Table S. 1 (Supplementary Material A) were computed by generalized linear mixed models fitted by the **glmmTMB** package (Brooks *et al.* 2021) and post-hoc inference was performed by the **multcomp** package (Hothorn, Bretz, and Westfall 2021).

We implemented all study settings in a dedicated R package called **htesim**. We also included the code and performance results of the empirical study as well as the code and dataset on postpartum blood loss. This should facilitate full reproducibility of all findings in this paper. The package is published on Github: <https://github.com/dandls/htesim>.

Acknowledgments

Torsten Hothorn received funding from the Swiss National Science Foundation, with the Grant No. 200021_184603, Horizon 2020 Research and Innovation Programme of the European Union under grant agreement number 681094, and is supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 15.0137.

References

- Akazawa M, Hashimoto K, Katsuhiko N, Kaname Y (2021). “Machine Learning Approach for the Prediction of Postpartum Hemorrhage in Vaginal Birth.” *Scientific Reports*, **11**, 22620. doi:10.1038/s41598-021-02198-y.
- Antoine C, Young BK (2021). “Cesarean Section one Hundred Years 1920–2020: the Good, the Bad and the Ugly.” *Journal of Perinatal Medicine*, **49**(1), 5–16. doi:doi:10.1515/jpm-2020-0305.

- Athey S, Tibshirani J, Wager S (2019). “Generalized Random Forests.” *The Annals of Statistics*, **47**(2), 1148–1178. doi:10.1214/18-aos1709.
- Athey S, Wager S (2019). “Estimating Treatment Effects with Causal Forests: An Application.” *Observational Studies*, **5**(2), 37–51. doi:10.1353/obs.2019.0001.
- Breiman L (2001). “Random Forests.” *Machine Learning*, **45**(1), 5–32. doi:10.1023/a:1010933404324.
- Brooks M, Bolker B, Kristensen K, Maechler M, Magnusson A, Skaug H, Nielsen A, Berg C, van Bentham K (2021). *glmmTMB: Generalized Linear Mixed Models Using Template Model Builder*. R package version 1.1.2, URL <https://CRAN.R-project.org/package=glmmTMB>.
- Buri M, Hothorn T (2020). “Model-Based Random Forests for Ordinal Regression.” *International Journal of Biostatistics*, **16**(2), 20190063. doi:10.1515/ijb-2019-0063.
- 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. doi:10.1111/ectj.12097.
- Chipman HA, George EI, McCulloch RE (2010). “BART: Bayesian Additive Regression Trees.” *The Annals of Applied Statistics*, **4**(1), 266–298. doi:10.1214/09-aos285.
- Cui Y, Kosorok MR, Sverdrup E, Wager S, Ruoqing (2022). “Estimating Heterogeneous Treatment Effects with Right-Censored Data via Causal Survival Forests.” *arXiv 2001.09887 v3*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2001.09887. URL <https://arxiv.org/abs/2001.09887>.
- Dasgupta A, Szymczak S, Moore J, Bailey-Wilson J, Malley JD (2014). “Risk Estimation Using Probability Machines.” *BioData Mining*, **7**(2), 2. doi:10.1186/1756-0381-7-2.
- Ende HB (2022). “Risk Assessment Tools to Predict Postpartum Hemorrhage.” *Best Practice & Research Clinical Anaesthesiology*. doi:10.1016/j.bpa.2022.08.003. Online first.
- Erickson EN, Carlson NS (2020). “Predicting Postpartum Hemorrhage After Low-Risk Vaginal Birth by Labor Characteristics and Oxytocin Administration.” *Journal of Obstetric, Gynecologic & Neonatal Nursing*, **49**(6), 549–563. doi:10.1016/j.jogn.2020.08.005.
- Farouki RT (2012). “The Bernstein Polynomial Basis: A Centennial Retrospective.” *Computer Aided Geometric Design*, **29**(6), 379–419. doi:10.1016/j.cagd.2012.03.001.
- Feng P, Zhou XH, Zou QM, Fan MY, Li XS (2012). “Generalized Propensity Score for Estimating the Average Treatment Effect of Multiple Treatments.” *Statistics in Medicine*, **31**(7), 681–697. doi:10.1002/sim.4168.
- Fokkema M, Smits N, Zeileis A, Hothorn T, Kelderman H (2018). “Detecting Treatment-Subgroup Interactions in Clustered Data with Generalized Linear Mixed-Effects Model Trees.” *Behavior Research Methods*, **50**(6), 2016–2034. doi:10.3758/s13428-017-0971-x.
- Foster JC, Taylor J, Ruberg S (2011). “Subgroup Identification from Randomized Clinical Trial Data.” *Statistics in Medicine*, **30**(24), 2867–2880. doi:10.1002/sim.4322.

- Haslinger C, Korte W, Hothorn T, Brun R, Greenberg C, Zimmermann R (2020). “The Impact of Prepartum Factor XIII Activity on Postpartum Blood Loss.” *Journal of Thrombosis and Haemostasis*, **18**, 1310–1319. doi:10.1111/jth.14795.
- Hill JL (2011). “Bayesian Nonparametric Modeling for Causal Inference.” *Journal of Computational and Graphical Statistics*, **20**(1), 217–240. doi:10.1198/jcgs.2010.08162.
- Hothorn T (2021). **trtf**: *Transformation Trees and Forests*. R package version 0.3-8, URL <http://ctm.R-forge.R-project.org>.
- Hothorn T, Bretz F, Westfall P (2021). **multcomp**: *Simultaneous Inference in General Parametric Models*. R package version 1.4-17, URL <https://CRAN.R-project.org/package=multcomp>.
- Hothorn T, Hornik K, Zeileis A (2006). “Unbiased Recursive Partitioning: A Conditional Inference Framework.” *Journal of Computational and Graphical Statistics*, **15**(3), 651–674. doi:10.1198/106186006x133933.
- Hothorn T, Lausen B, Benner A, Radespiel-Tröger M (2004). “Bagging Survival Trees.” *Statistics in Medicine*, **23**(1), 77–91. doi:10.1002/sim.1593.
- Hothorn T, Möst L, Bühlmann P (2018). “Most Likely Transformations.” *Scandinavian Journal of Statistics*, **45**(1), 110–134. doi:10.1111/sjos.12291.
- Hothorn T, Zeileis A (2015). “**partykit**: A Modular Toolkit for Recursive Partytioning in R.” *Journal of Machine Learning Research*, **16**, 3905–3909. URL <https://jmlr.org/papers/v16/hothorn15a.html>.
- Hothorn T, Zeileis A (2021a). **partykit**: *A Toolkit for Recursive Partytioning*. R package version 1.2-15, URL <http://partykit.r-forge.r-project.org/partykit/>.
- Hothorn T, Zeileis A (2021b). “Predictive Distribution Modelling Using Transformation Forests.” *Journal of Computational and Graphical Statistics*, **14**, 144–148. doi:10.1080/10618600.2021.1872581.
- Imbens G, Athey S (2016). “Recursive Partitioning for Heterogeneous Causal Effects.” *Proceedings of the National Academy of Sciences of the United States of America*, **113**(27), 7353–7360. doi:10.1073/pnas.1510489113.
- Imbens GW, Rubin DW (2015). *Causal Inference for Statistics, Social, and Biomedical Sciences*. Cambridge University Press, New York.
- Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS (2008). “Random Survival Forests.” *The Annals of Applied Statistics*, **2**(3), 841–860. doi:10.1214/08-aos169.
- Ishwaran H, Malley JD (2014). “Synthetic Learning Machines.” *BioData Mining*, **7**(28). doi:10.1186/s13040-014-0028-y.
- Kahr MK, Brun R, Zimmermann R, Franke D, Haslinger C (2018). “Validation of a Quantitative System for Real-time Measurement of Postpartum Blood Loss.” *Archives of Gynecology and Obstetrics*, **298**, 1071–1077. doi:10.1007/s00404-018-4896-0.

- Kawakita T, Mokhtari N, Huang JC, Landy HJ (2019). “Evaluation of Risk-Assessment Tools for Severe Postpartum Hemorrhage in Women Undergoing Cesarean Delivery.” *Obstetrics & Gynecology*, **134**(6), 1308–1316. doi:10.1097/AOG.0000000000003574.
- Korepanova N, Seibold H, Steffen V, Hothorn T (2020). “Survival Forests under Test: Impact of the Proportional Hazards Assumption on Prognostic and Predictive Forests for ALS Survival.” *Statistical Methods in Medical Research*, **29**(5), 1403–1419. doi:10.1177/0962280219862586.
- Künzel SR, Sekhon JS, Bickel PJ, Yu B (2019). “Metalearners for Estimating Heterogeneous Treatment Effects Using Machine Learning.” *Proceedings of the National Academy of Sciences of the United States of America*, **116**(10), 4156–4165. doi:10.1073/pnas.1804597116.
- Lin Y, Jeon Y (2006). “Random Forests and Adaptive Nearest Neighbors.” *Journal of the American Statistical Association*, **101**(474), 578–590. doi:10.1198/016214505000001230.
- Lu M, Sadiq S, Feaster DJ, Ishwaran H (2018). “Estimating Individual Treatment Effect in Observational Data Using Random Forest Methods.” *Journal of Computational and Graphical Statistics*, **27**(1), 209–219. doi:10.1080/10618600.2017.1356325.
- MacDorman MF, Declercq E, Cabral H, Morton C (2016). “Recent Increases in the U.S. Maternal Mortality Rate: Disentangling Trends From Measurement Issues.” *Obstetrics & Gynecology*, **128**(3), 447–455. doi:10.1097/AOG.0000000000001556.
- Mayer I, Sverdrup E, Gauss T, Moyer JD, Wager S, Josse J (2020). “Doubly Robust Treatment Effect Estimation with Missing Attributes.” *The Annals of Applied Statistics*, **14**(3), 1409–1431. doi:10.1214/20-aos1356.
- Meinshausen N (2006). “Quantile Regression Forests.” *Journal of Machine Learning Research*, **7**, 983–999. doi:10.1007/s10994-014-5452-1.
- Nie X, Wager S (2021). “Quasi-Oracle Estimation of Heterogeneous Treatment Effects.” *Biometrika*, **108**(2), 299–319. doi:10.1093/biomet/asaa076.
- Powers S, Qian J, Jung K, Schuler A, Shah NH, Hastie T, Tibshirani R (2018). “Some Methods for Heterogeneous Treatment Effect Estimation in High Dimensions.” *Statistics in Medicine*, **37**(11), 1767–1787. doi:10.1002/sim.7623.
- Probst P, Boulesteix AL, Bischl B (2021). “Tunability: Importance of Hyperparameters of Machine Learning Algorithms.” *Journal of Machine Learning Research*, **20**(1), 1934–1965.
- Probst P, Wright MN, Boulesteix AL (2019). “Hyperparameters and tuning strategies for random forest.” *WIREs Data Mining and Knowledge Discovery*, **9**(3). doi:10.1002/widm.1301.
- R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Robinson PM (1988). “Root-N-Consistent Semiparametric Regression.” *Econometrica*, **56**(4), 931–954. doi:10.2307/1912705.

- Rosenbaum PR, Rubin DB (1983). “The Central Role of the Propensity Score in Observational Studies for Causal Effects.” *Biometrika*, **70**(1), 41–55. doi:10.1093/biomet/70.1.41.
- Say L, Chou D, Gemmill A, Tunçalp O, Moller AB, Daniels J, Gülmezoglu AM, Temmerman M, Alkema L (2014). “Global Causes of Maternal Death: a WHO Systematic Analysis.” *The Lancet Global Health*, **2**(6), e323–e333. doi:10.1016/S2214-109X(14)70227-X.
- Schlosser L, Hothorn T, Stauffer R, Zeileis A (2019). “Distributional Regression Forests for Probabilistic Precipitation Forecasting in Complex Terrain.” *The Annals of Applied Statistics*, **13**(3), 1564–1589. doi:10.1214/19-AOAS1247.
- Schnell P, Tang Q, Müller P, Carlin BP (2017). “Subgroup Inference for Multiple Treatments and Multiple Endpoints in an Alzheimer’s Disease Treatment Trial.” *The Annals of Applied Statistics*, **11**(2), 949–966. doi:10.1214/17-aoas1024.
- Seibold H, Zeileis A, Hothorn T (2016). “Model-Based Recursive Partitioning for Subgroup Analyses.” *International Journal of Biostatistics*, **12**(1), 45–63. doi:10.1515/ijb-2015-0032.
- Seibold H, Zeileis A, Hothorn T (2018). “Individual Treatment Effect Prediction for Amyotrophic Lateral Sclerosis Patients.” *Statistical Methods in Medical Research*, **27**(10), 3104–3125. doi:10.1177/0962280217693034.
- Seibold H, Zeileis A, Hothorn T (2019). “**model4you**: An R Package for Personalised Treatment Effect Estimation.” *Journal of Open Research Software*, **7**(17), 1–6. doi:10.5334/jors.219.
- Seibold H, Zeileis A, Hothorn T (2021). **model4you**: *Stratified and Personalised Models Based on Model-Based Trees and Forests*. R package version 0.9-7, URL <https://CRAN.R-project.org/package=model4you>.
- Starling JE, Murray JS, Lohr PA, Aiken ARA, Carvalho CM, Scott JG (2021). “Targeted Smooth Bayesian Causal Forests: An Analysis of Heterogeneous Treatment Effects for Simultaneous vs. Interval Medical Abortion Regimens Over Gestation.” *The Annals of Applied Statistics*, **15**(3), 1194–1219. doi:10.1214/20-AOAS1438.
- Tang F, Ishwaran H (2017). “Random Forest Missing Data Algorithms.” *Statistical Analysis and Data Mining*, **10**(6), 363–377. doi:10.1002/sam.11348.
- Tibshirani J, Athey S, Sverdrup E, Wager S (2021). **grf**: *Generalized Random Forests*. R package version 2.0.2, URL <https://CRAN.R-project.org/package=grf>.
- Venkatesh KK, Strauss RA, Grotegut CA, Heine RP, Chescheir NC, Stringer JSA, Stamilio DM, Menard KM, Jelovsek JE (2020). “Machine Learning and Statistical Models to Predict Postpartum Hemorrhage.” *Obstetrics & Gynecology*, **135**(4), 935–944. doi:10.1097/AOG.0000000000003759.
- 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. doi:10.1080/01621459.2017.1319839.

WHO (2012). “WHO Recommendations for the Prevention and Treatment of Postpartum Haemorrhage.” World Health Organization, Geneva, Switzerland.

Zanutto E, Lu B, Hornik R (2005). “Using Propensity Score Subclassification for Multiple Treatment Doses to Evaluate a National Antidrug Media Campaign.” *Journal of Educational and Behavioral Statistics*, **30**(1), 59–73. doi:10.3102/10769986030001059.

Zeileis A, Hothorn T, Hornik K (2008). “Model-Based Recursive Partitioning.” *Journal of Computational and Graphical Statistics*, **17**(2), 492–514. doi:10.1198/106186008x319331.

Supplementary Material

A.1. Cut-point selection in detail

In this section, we compare cut-point selection of model-based forests with causal forests. For ease of exposition, we only consider $p = 1$ covariate. Our aim is to divide a parent node with n samples into two child nodes.

Model-based forests allow splits both based on the intercept μ and treatment effect τ in the model $Y = \mu + \tau w + \epsilon$, where Y is the outcome and w is the treatment assignment. These two can be centered or not without loss of generality, i.e. $Y_i := Y_i - \hat{Y}_i$ and $w_i := w_i - \pi(\mathbf{X}_i)$. Contrary to model-based forests, causal forests only split according to τ .

We define W_i as the intercept augmented vector $(1 \ w_i)$. We denote the score function for the above model evaluated in the parent node as ψ , a $n \cdot 2$ matrix with columns corresponding to μ and τ . Let n_L and n_R be the number of samples in the left and right child node, respectively.

A.1.1. Model-based forest criterion

Model-based forests first select a splitting variable using permutation tests before a split point is found. Since we only consider one covariate, we skip this step and continue with the selection of cut points. Let $\Sigma\psi_L$ be the sum of the score vector in the left child. Let $Vh = \frac{1}{n} \sum_{i=1}^n \psi^{\otimes 2}$ be a $2 \cdot 2$ weight matrix. We define $E = n_L \bar{\psi}$ with $\bar{\psi} = (\bar{\psi}_\mu, \bar{\psi}_\tau)$ as the vector of average scores in the parent node. With $Z_{\text{mob}} = \Sigma\psi_L - E$ and the weight matrix $V_{\text{mob}} = ((n n_L / (n-1) - n_L^2 / (n-1)) Vh)^{-1}$ the model-based forest objective is:

$$C_{\text{mob}} = Z'_{\text{mob}} V_{\text{mob}} Z_{\text{mob}}.$$

A.1.2. Causal forest criterion

Causal forests apply CART splitting on pseudo-outcomes ρ . The objective is displayed in Equation 5 of [Athey et al. \(2019\)](#):

$$C_{\text{cf}} = n_L n_R / n^2 \|\bar{\rho}_L - \bar{\rho}_R\|^2,$$

where $\bar{\rho}_L$ is the average ρ in the left child, and likewise for the right child. The weight value is $A_p = \frac{1}{n} \sum_{i=1}^n w_i^2$. The $n \cdot 2$ matrix of pseudo-outcomes ρ are then $\rho = \psi_\tau A_p^{-1}$.

The criterion C_{cf} can also be written as a quadratic form similar to model-based forests: Define $Z_{\text{cf}} = \bar{\psi}_{\tau,L} - \bar{\psi}_{\tau,R}$ and $V_{\text{cf}} = n_L n_R / n^2 A_p^{-2}$ with $\bar{\psi}_{\tau,L}$ and $\bar{\psi}_{\tau,R}$ as the average scores in the left and right child. Then $C_{\text{cf}} = Z'_{\text{cf}} V_{\text{cf}} Z_{\text{cf}}$ will have the same argmax as above's C_{cf} .

A.2. Empirical results for honest forests

Comparative results of adaptive and honest forests are presented in Figures S. 1 and S. 2 for the study setting of Section 4. As for adaptive forests we statistically analyzed honest forests (Table S. 1). Rankings of the methods in their honest versions were in line with the results for the adaptive versions. Most pronounced differences occurred for RQ 2: While $\text{mob}(\hat{Y}, \hat{W})$

performed slightly better than mobcf in their adaptive versions, they performed akin in their honest versions. Additional splitting based on the prognostic effect in model-based forests thus had a smaller impact on performance. Honesty was beneficial in Setups A and C with strong or complicated confounding. For Setup B, the results differed only slightly in favor of the adaptive versions. For Setup D, honesty worsened the results of all forest approaches.

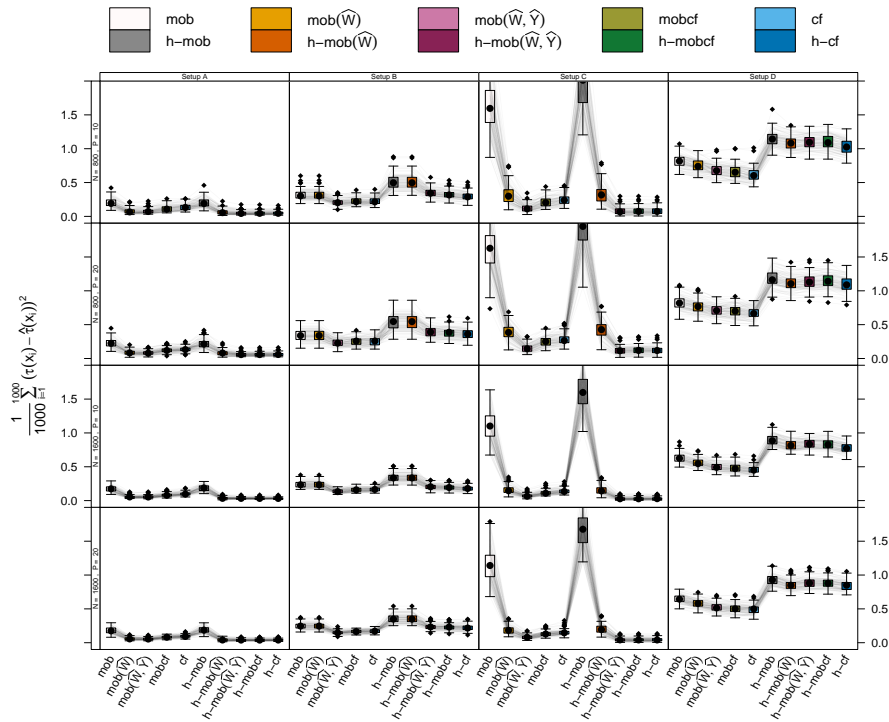


Figure S. 1: Results for the experimental setups of Section 4. Direct comparison of the adaptive and honest versions of causal forests, model-based forests without centering (mob), mob imitating causal forests (mobcf), mob with centered W ($\text{mob}(\hat{W})$) and additional of Y ($\text{mob}(\hat{W}, \hat{Y})$). 'h-' denotes the honest version of a forest.

DGP	N	P	Mean squared error ratio							
			(RQ 1)	(RQ 2)	(RQ 3)	(RQ 4)	(RQ 5)	(RQ 6)	(RQ 7)	
Setup A	800	10	<i>0.2657 (0.2127, 0.3241)</i>	<i>1.007 (0.752, 1.348)</i>	<i>1.008 (0.754, 1.347)</i>	<i>0.336 (0.284, 0.398)</i>	<i>1.292 (0.996, 1.677)</i>	<i>1.301 (1.001, 1.690)</i>	<i>1.351 (1.080, 1.690)</i>	
		20	<i>0.284 (0.237, 0.342)</i>	<i>1.008 (0.782, 1.299)</i>	<i>1.012 (0.787, 1.300)</i>	<i>0.377 (0.327, 0.434)</i>	1.340 (1.073, 1.675)	1.351 (1.080, 1.690)	1.351 (1.080, 1.690)	
	1600	10	<i>0.189 (0.137, 0.261)</i>	<i>0.979 (0.622, 1.542)</i>	<i>1.028 (0.654, 1.616)</i>	<i>0.202 (0.149, 0.273)</i>	<i>1.096 (0.706, 1.700)</i>	<i>1.073 (0.695, 1.656)</i>	<i>1.059 (0.735, 1.527)</i>	
		20	<i>0.223 (0.171, 0.292)</i>	<i>0.991 (0.680, 1.446)</i>	<i>1.025 (0.705, 1.490)</i>	<i>0.233 (0.180, 0.301)</i>	<i>1.069 (0.740, 1.543)</i>	<i>1.059 (0.735, 1.527)</i>	<i>1.059 (0.735, 1.527)</i>	
	Setup B	800	10	<i>0.578 (0.553, 0.604)</i>	<i>0.926 (0.885, 0.971)</i>	<i>0.915 (0.869, 0.963)</i>	<i>1.000 (0.970, 1.030)</i>	1.584 (1.520, 1.650)	1.497 (1.411, 1.525)	1.497 (1.411, 1.525)
			20	<i>0.677 (0.652, 0.703)</i>	<i>0.969 (0.930, 1.010)</i>	<i>0.962 (0.922, 1.004)</i>	<i>1.000 (0.971, 1.030)</i>	1.422 (1.371, 1.474)	1.378 (1.330, 1.428)	1.378 (1.330, 1.428)
Setup C	1600	10	<i>0.526 (0.491, 0.565)</i>	<i>0.957 (0.884, 1.037)</i>	<i>0.924 (0.849, 1.006)</i>	<i>1.000 (0.955, 1.048)</i>	1.756 (1.644, 1.876)	1.681 (1.577, 1.793)	1.681 (1.577, 1.793)	
		20	<i>0.621 (0.586, 0.659)</i>	<i>0.976 (0.912, 1.046)</i>	<i>0.973 (0.907, 1.043)</i>	<i>1.000 (0.957, 1.045)</i>	1.565 (1.477, 1.658)	1.528 (1.443, 1.618)	1.528 (1.443, 1.618)	
Setup D	800	10	<i>0.044 (0.039, 0.050)</i>	<i>1.046 (0.868, 1.261)</i>	<i>1.019 (0.850, 1.220)</i>	<i>0.168 (0.163, 0.174)</i>	<i>3.902 (3.416, 4.458)</i>	<i>4.084 (3.654, 4.692)</i>	<i>4.084 (3.654, 4.692)</i>	
		20	<i>0.067 (0.061, 0.073)</i>	<i>1.032 (0.824, 1.263)</i>	<i>1.024 (0.831, 1.163)</i>	<i>0.169 (0.164, 0.174)</i>	<i>3.165 (2.856, 3.536)</i>	<i>3.429 (3.127, 3.744)</i>	<i>3.429 (3.127, 3.744)</i>	
Setup D	1600	10	<i>0.020 (0.015, 0.029)</i>	<i>1.034 (0.820, 1.264)</i>	<i>1.024 (0.831, 1.163)</i>	<i>0.169 (0.164, 0.174)</i>	4.844 (4.402, 5.290)	4.844 (4.402, 5.290)	4.844 (4.402, 5.290)	
		20	<i>0.028 (0.022, 0.036)</i>	<i>0.994 (0.708, 1.396)</i>	<i>1.010 (0.730, 1.417)</i>	<i>0.122 (0.115, 0.128)</i>	4.390 (3.923, 5.620)	4.364 (3.914, 5.579)	4.364 (3.914, 5.579)	
Setup D	800	10	<i>0.895 (0.882, 0.909)</i>	<i>1.001 (0.987, 1.016)</i>	<i>0.936 (0.922, 0.950)</i>	<i>0.942 (0.929, 0.956)</i>	<i>0.985 (0.970, 0.999)</i>	<i>0.986 (0.972, 1.000)</i>	<i>0.986 (0.967, 0.994)</i>	
		20	<i>0.929 (0.916, 0.942)</i>	<i>1.010 (0.996, 1.024)</i>	<i>0.955 (0.941, 0.968)</i>	<i>0.945 (0.932, 0.958)</i>	<i>0.971 (0.957, 0.985)</i>	<i>0.971 (0.957, 0.985)</i>	<i>0.971 (0.957, 0.985)</i>	
Setup D	1600	10	<i>0.868 (0.851, 0.885)</i>	<i>0.992 (0.973, 1.011)</i>	<i>0.931 (0.913, 0.949)</i>	<i>0.915 (0.898, 0.932)</i>	<i>0.981 (0.963, 1.000)</i>	<i>0.973 (0.955, 0.992)</i>	<i>0.973 (0.955, 0.992)</i>	
		20	<i>0.913 (0.896, 0.929)</i>	<i>1.000 (0.982, 1.018)</i>	<i>0.958 (0.940, 0.975)</i>	<i>0.919 (0.905, 0.936)</i>	<i>0.964 (0.947, 0.982)</i>	<i>0.964 (0.946, 0.982)</i>	<i>0.964 (0.946, 0.982)</i>	

Table S. 1: Results for the experimental setups of Section 4 for the *honest* versions of the methods. Comparison of differences in mean squared error for $\hat{\tau}(\mathbf{x})$ in different scenarios. Estimates and 95% confidence intervals were obtained from a normal linear mixed model with log-link. Cells printed in bold font correspond to a superior reference (*mob* in the first and fourth columns, *mob*(\hat{W} , \hat{Y}) in the second column, *mobcf* in the third column and *mob*(\hat{W}) in the last column), cells printed in italics indicate an inferior reference.

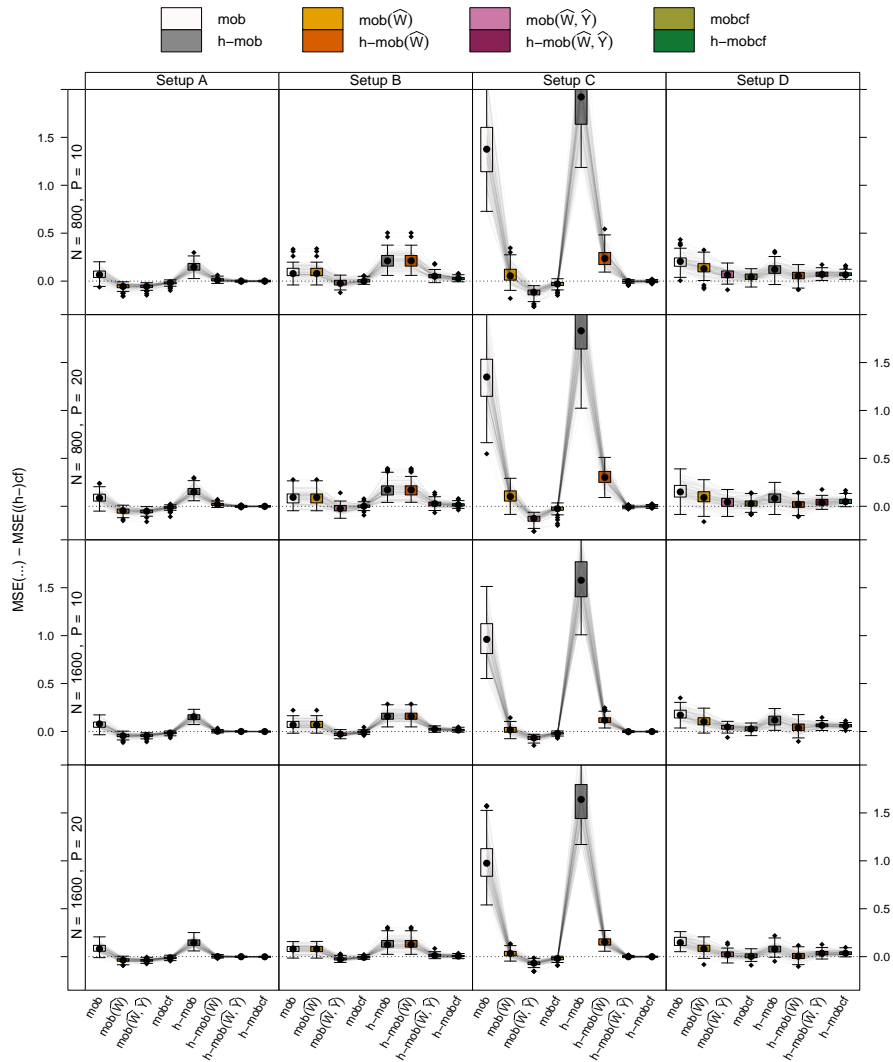


Figure S. 2: Results for experimental setups of Section 4. Direct comparison of the mean squared differences to causal forests for model-based forests without centering (mob), mob imitating causal forests (mobcf), mob with centered W ($\text{mob}(\hat{W})$) and additional of Y ($\text{mob}(\hat{W}, \hat{Y})$). 'h-' denotes the honest version of a forest. In their adaptive versions, methods were compared to adaptive causal forests, while honest versions to honest causal forests.

A.3. Sensitivity of `mtry` parameter

Sensitivity of the random forest for PPH presented in Section 5 of the main manuscript was studied with respect to different choices of the main tuning parameter, `mtry` (the number of randomly selected covariates for split evaluation in each node of the underlying trees). In Figure S. 3, the out-of-bag log-likelihoods for several choices of `mtry` are presented, showing an insignificant amount of variability and thus results can be expected to be quite stable with respect to the choice of `mtry`.

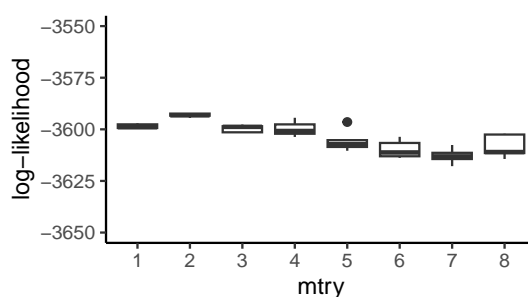


Figure S. 3: Effect of the `mtry` parameter on (out-of-bag) log-likelihood of the transformation forest (Section 5). Forest fitting was repeated 5 times for each `mtry` parameter. All other hyperparameters of the transformation forest were kept at their respective values according to Section 7.

Affiliation:

Susanne Dandl
 Institut für Statistik, Ludwig-Maximilians-Universität München, Munich Center for Machine Learning (MCML), Germany

Christian Haslinger
 Klinik für Geburtshilfe, Universitätsspital und Universität Zürich, Switzerland

Torsten Hothorn
 Institut für Epidemiologie, Biostatistik und Prävention, Universität Zürich
 Hirschengraben 84, CH-8001 Zürich, Switzerland
 E-mail: Torsten.Hothorn@R-project.org

Heidi Seibold
 Institute for Globally Distributed Open Research and Education (IGDORE),
 München, Germany

Erik Sverdrup
Stanford Graduate School of Business, Stanford University, U.S.A.

Stefan Wager
Stanford Graduate School of Business, Stanford University, U.S.A.

Achim Zeileis
Faculty of Economics and Statistics, Universität Innsbruck, Austria

6 Heterogeneous Treatment Effect Estimation for Observational Data using Model-based Forests

Contributing Article

Dandl S, Bender A, Hothorn T (2022a). “Heterogeneous Treatment Effect Estimation for Observational Data using Model-based Forests.” *arXiv 2210.02836*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2210.02836. To appear in *Statistical Methods in Medical Research*

The article was accepted by the journal of *Statistical Methods in Medical Research* shortly before the disputation took place. The following manuscript is the initially submitted version of the work available on arXiv.

Replication Code

The code for replicating the results in this manuscript is available as part of the R package **htesim** available at <https://github.com/dandls/htesim>.

Declaration of Contributions

Together with Torsten Hothorn, Susanne Dandl derived a general framework to adapt the orthogonalization strategy to model-based forests for diverse outcome types. Susanne Dandl implemented the orthogonalization approach, the real-world use case, and the infrastructure to conduct the simulation study in parallel. For simulating data, she extended the package of the contribution in Chapter 5 to cover data-generating processes for diverse outcomes. She performed the experiment, and aggregated and interpreted the results. Susanne Dandl wrote the initial draft of Sections 3-5, including all figures, and edited large parts of Sections 1 and 2. She revised the manuscript according to the feedback of her co-authors and external reviewers.

Contributions of Co-authors

Torsten Hothorn wrote the first drafts of Sections 2 and 6. He also wrote the initial code for simulating data with diverse outcome types and reviewed the simulation study code. Andreas Bender provided valuable advice on the use case. All co-authors helped to revise the manuscript.

Heterogeneous Treatment Effect Estimation for Observational Data using Model-based Forests

Susanne Dandl 
LMU München, MCML

Andreas Bender 
LMU München, MCML

Torsten Hothorn 
Universität Zürich

Abstract

The estimation of heterogeneous treatment effects (HTEs) has attracted considerable interest in many disciplines, most prominently in medicine and economics. Contemporary research has so far primarily focused on continuous and binary responses where HTEs are traditionally estimated by a linear model, which allows the estimation of constant or heterogeneous effects even under certain model misspecifications. More complex models for survival, count, or ordinal outcomes require stricter assumptions to reliably estimate the treatment effect. Most importantly, the noncollapsibility issue necessitates the joint estimation of treatment and prognostic effects. Model-based forests allow simultaneous estimation of covariate-dependent treatment and prognostic effects, but only for randomized trials. In this paper, we propose modifications to model-based forests to address the confounding issue in observational data. In particular, we evaluate an orthogonalization strategy originally proposed by Robinson (1988, *Econometrica*) in the context of model-based forests targeting HTE estimation in generalized linear models and transformation models. We found that this strategy reduces confounding effects in a simulated study with various outcome distributions. We demonstrate the practical aspects of HTE estimation for survival and ordinal outcomes by an assessment of the potentially heterogeneous effect of Riluzole on the progress of Amyotrophic Lateral Sclerosis.

Keywords: Heterogeneous treatment effects, personalized medicine, random forest, observational data, censored survival data, generalized linear model, transformation model.

1. Introduction

Over the past years, there has been emerging interest in methods to estimate heterogeneous treatment effects (HTEs) in various application fields. In healthcare, HTE estimation can be understood as a core principle driving personalized medicine. As opposed to average treatment effects, which assume a constant effect of a treatment on an outcome for the whole population, HTEs account for the heterogeneity in the effect for subgroups or individuals based on their characteristics. Most research on HTE estimation has mainly focused on continuous and binary response variables. These methods have typically built upon Rubin's potential outcomes framework, a statistical approach to formulating and inferring causal effects in various designs (Rubin 1974, 2005).

Traditionally, statistical models were used to estimate the treatment effect, but machine learning methods have been more and more adapted for these tasks over the past decade. Machine learning models rely on weaker assumptions and can automatically learn complex relation-

arXiv:2210.02836v1 [stat.ME] 6 Oct 2022

ships such as higher order interaction effects, resulting in greater predictive performance in a variety of applications. In the case of continuous or binary responses, prominent methods to estimate HTEs are based on random forests (Foster, Taylor, and Ruberg 2011; Lu, Sadiq, Feaster, and Ishwaran 2018; Athey, Tibshirani, and Wager 2019; Powers, Qian, Jung, Schuler, Shah, Hastie, and Tibshirani 2018; Su, Peña, Liu, and Levine 2018; Li, Levine, and Fan 2022), Bayesian additive regression trees (BART) (Hill 2011; Hu, Gu, Lopez, Ji, and Wisnivesky 2020), or neural networks (Shalit, Johansson, and Sontag 2017; Curth, Lee, and van der Schaar 2021; Chapfuwa, Assaad, Zeng, Pencina, Carin, and Henao 2021). Künzel, Sekhon, Bickel, and Yu (2019) proposed general frameworks – T-learners, S-learners, U-learners, and X-learners – that base treatment effect estimates on arbitrary machine learning models. Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018) coined the term double/debiased machine learning models, which uses machine learning models for nuisance parameter estimations. The approach still relies on parametric models for estimating treatment effects, but Nie and Wager (2021) derived so-called R-learners that allow for arbitrary (nonparametric or semiparametric) models.

Beyond continuous or binary responses, research on machine learning methods for HTE estimation have primarily focused on (right-censored) survival data. Methods have been proposed based on Bayesian additive regression trees (BART) (Henderson, Louis, Rosner, and Varadhan 2018), random forest-type methods (Cui, Kosorok, Sverdrup, Wager, and Ruoqing 2022; Tabib and Larocque 2020), or deep learning approaches (Curth *et al.* 2021; Chapfuwa *et al.* 2021). Theoretically, any machine learning model for survival analysis – such as random survival forests (Ishwaran, Kogalur, Blackstone, and Lauer 2008) or a Cox regression-based deep neural network (deepSurv) (Katzman, Shaham, Cloninger, Bates, Jiang, and Kluger 2018) – can estimate HTEs (Hu, Ji, and Li 2021). These models can estimate survival or hazard functions in both treatment groups separately; HTEs are then defined as the difference in derived properties of the two functions, e.g., as differences in the median survival time. However, Hu *et al.* (2021) found that methods specifically designed for HTE estimation, like the adapted BART (Henderson *et al.* 2018), produce more reliable estimates.

In general, for a continuous or binary outcome Y conditional on treatment w and covariates \mathbf{x} , the conditional average treatment effect $\tau(\mathbf{x})$ (CATE) can be estimated from the model $\mathbb{E}(Y \mid W = w, \mathbf{X} = \mathbf{x}) = \mu(\mathbf{x}) + \tau(\mathbf{x})w$ even if the model is misspecified, e.g., when the prognostic effect $\mu(\mathbf{x})$ cannot be fully estimated due to missing covariate information. Beyond mean regression, stricter assumptions are necessary both for randomized and for observational studies to estimate HTEs. For example, under a true Cox model with survivor function $\exp(-\exp(h(t) + \mu(\mathbf{x}) + \tau w))$ with log-cumulative baseline hazard $h(t)$ at time t and log-hazard ratio τ , the prognostic effect $\mu(\mathbf{x})$ must be specified correctly, even in a randomized trial. Estimated marginal log-hazard ratios $\hat{\tau}$ – i.e., when the model is fitted under the constraint $\mu(\mathbf{x}) \equiv 0$ – are shrunken towards zero if this constraint is unrealistic (Aalen, Cook, and Røysland 2015). Naturally, this problem carries over to heterogeneous log-hazard ratios $\tau(\mathbf{x})$.

Consequently, HTE estimation in more complex models requires the simultaneous estimation of both the prognostic part $\mu(\mathbf{x})$ and the predictive HTE $\tau(\mathbf{x})$. Model-based forests have been demonstrated to allow estimation of $\mu(\mathbf{x})$ and $\tau(\mathbf{x})$ in randomized trials (Seibold, Zeileis, and Hothorn 2016, 2018; Korepanova, Seibold, Steffen, and Hothorn 2020; Buri and Hothorn 2020; Fokkema, Smits, Zeileis, Hothorn, and Kelderman 2018; Hothorn and Zeileis 2021b). In a nutshell, model-based forests combine the parametric modeling framework with random

forests to estimate individual treatment effects (Seibold *et al.* 2018). By using generalized linear models and transformation models, model-based forests can be adapted for survival data (Seibold *et al.* 2016, 2018; Korepanova *et al.* 2020), ordinal data (Buri and Hothorn 2020), or clustered data (Fokkema *et al.* 2018). A unique feature of model-based forests is the simultaneous estimation of both treatment and prognostic effects in the same forest model.

In observational studies the treatment group assignment is not under control of the researcher and confounding effects could bias the estimation of HTEs. In this work, we propose and evaluate novel variants of model-based forests for HTE estimation in observational studies. Adaptions of Robinson’s orthogonalization strategy for generalized linear models and transformation models are discussed and implemented. We review key components of model-based forests for HTE estimation in randomized trials in Section 2. In Section 3, we start introducing the orthogonalization approach by Robinson (1988), which is instrumental for achieving robustness to confounding effects in the non-randomized situation. We motivate previous developments using linear models (Dandl, Hothorn, Seibold, Sverdrup, Wager, and Zeileis 2022) and leverage adaptations to more complex models discussed by Gao and Hastie (2022) to define novel model-based forest variants suitable for HTE in the observational setting. These variants’ performances are empirically assessed in a simulation study with a range of outcome distributions in Section 4. Finally, in Section 5 presenting a re-analysis of the patient-specific effect of Riluzole in patients with Amyotrophic Lateral Sclerosis (ALS), practical aspects of model estimation and interpretation are discussed.

2. Review of model-based forests for randomized trials

We are interested in estimating HTEs based on i.i.d. observations (y, \mathbf{x}, w) , where y , \mathbf{x} and w are realizations of the outcome Y , covariates $\mathbf{X} \in \mathcal{X}$, and control vs. treatment indicator $W \in \{0, 1\}$. $Y(0)$ and $Y(1)$ denote the potential outcomes under the two treatment conditions $W \in \{0, 1\}$. Throughout this paper, we assume that \mathbf{X} includes all relevant variables to explain heterogeneity both in the treatment effect and the outcome Y , and that the base model underlying model-based forests is correctly specified.

We review model-based forests for HTE estimation based on randomized trials as introduced by Seibold *et al.* (2018) and Korepanova *et al.* (2020). Within this section, we only consider settings where the treatment assignment is randomized and, therefore, follows a binomial model $W \mid \mathbf{X} = \mathbf{x} \sim B(1, \pi(\mathbf{x}))$ with constant propensities $\pi(\mathbf{x}) \equiv \pi$. We omit discussion of the abstract framework underlying model-based forests and instead discuss the important linear, generalized linear (Seibold *et al.* 2018), and transformation models (Korepanova *et al.* 2020) in detail.

2.1. Linear model

For a continuous outcome $Y \in \mathbb{R}$ with symmetric error distribution, a model-based forest might be defined based on the model

$$(Y \mid \mathbf{X} = \mathbf{x}, W = w) = \mu(\mathbf{x}) + \tau(\mathbf{x})w + \phi Z \quad (1)$$

where the residuals are given by the error term ϕZ with $\mathbb{E}(Z \mid \mathbf{X}, W) = 0$ and standard deviation $\phi > 0$ (Dandl *et al.* 2022). We are mainly interested in estimating $\tau(\mathbf{x})$, the treatment effect that depends on *predictive* variables in \mathbf{x} . With model-based forests, however,

we also obtain an estimated value for the prognostic effect $\mu(\mathbf{x})$, which depends on *prognostic* variables in \mathbf{x} . A variable might be predictive and prognostic at the same time. We refer to these situations as “overlays”.

Because we assume in this section that $\pi(\mathbf{x}) \equiv \pi$ applies, $W \perp\!\!\!\perp \mathbf{X}$ holds. Consequently, $\tau(\mathbf{x})$ can be interpreted as a CATE

$$\tau(\mathbf{x}) = \text{CATE}(\mathbf{x}) = \mathbb{E}(Y(1) - Y(0) \mid \mathbf{X} = \mathbf{x}) \quad (2)$$

on the absolute scale. To estimate $(\mu(\mathbf{x}), \tau(\mathbf{x}))^\top$ the L_2 loss

$$\ell(\mu(\mathbf{x}), \tau(\mathbf{x})) = 1/2 (Y - \mu(\mathbf{x}) - \tau(\mathbf{x})w)^2 \quad (3)$$

is minimized w.r.t. μ and τ using an ensemble of trees. Inspired by recursive partitioning techniques (Hothorn, Hornik, and Zeileis 2006; Zeileis, Hothorn, and Hornik 2008), split variable and split point selection are separated. The split variable is the variable that has the lowest p -value for the bivariate permutation tests for the H_0 -hypothesis that μ and τ are constant and independent of any split variable. The cut-point is the point of the chosen split variable at which the score functions

$$s(\hat{\mu}, \hat{\tau}) := (Y - \hat{\mu} - \hat{\tau}w)(1, w)^\top$$

in the two resultant subgroups differ the most; details are available in Appendix 2 of Seibold *et al.* (2018).

Once $B \in \mathbb{N}$ trees were fitted to subsamples of the training data, predictions for the treatment effect for a new observation \mathbf{x} are obtained via local maximum likelihood aggregation (Hothorn, Lausen, Benner, and Radespiel-Tröger 2004; Meinshausen 2006; Lin and Jeon 2006; Athey *et al.* 2019; Hothorn and Zeileis 2021b). First, for the i -th training sample, the frequency α_i with which it falls in the same leaf as \mathbf{x} over all B trees is measured. The obtained weighting vector $(\alpha_1, \dots, \alpha_n)$ is used as an input for minimizing

$$(\hat{\mu}(\mathbf{x}), \hat{\tau}(\mathbf{x}))^\top = \arg \min_{\mu, \tau} \sum_{i=1}^n \alpha_i(\mathbf{x}) \ell_i(\mu, \tau) \quad (4)$$

where ℓ_i denotes the loss for the i -th sample. Model-based forests easily allow adaptations if HTEs for an outcome variable Y that is not well represented by equation (1) should be estimated. In this case, model-based forests can build on generalized linear models or transformation models in the recursive partitioning framework (Zeileis *et al.* 2008). As detailed in the following sections, the loss function ℓ in equation (3) changes from the squared error to the negative (partial) log-likelihood of some appropriate model.

2.2. Generalized linear models

When the conditional outcome distribution is better described through a generalized linear model

$$(Y \mid \mathbf{X} = \mathbf{x}, W = w) \sim \text{ExpFam}(\theta(\mu(\mathbf{x}) + \tau(\mathbf{x})w), \phi)$$

with parameter θ depending on the additive function $\mu(\mathbf{x}) + \tau(\mathbf{x})w$, the conditional mean

$$g(\mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}, W = w)) = \mu(\mathbf{x}) + \tau(\mathbf{x})w =: \eta_w(\mathbf{x}) \quad (5)$$

is linear on the scale of a link function g . Thus, the interpretation of $\tau(\mathbf{x})$ as CATE (2) generally no longer holds. Instead, the predictive effect is understood as the difference in natural parameters (DINA (Gao and Hastie 2022))

$$\tau(\mathbf{x}) = \text{DINA}(\mathbf{x}) = \eta_1(\mathbf{x}) - \eta_0(\mathbf{x}). \quad (6)$$

In contrast to the linear model case, HTEs $\tau(\mathbf{x})$ are now defined on relative scales, such as odds ratios in binary logistic regression models or multiplicative mean effects in a Poisson or Gaussian model with a log-link. The negative log-likelihood contribution of some observation (Y, \mathbf{x}, w) is

$$\ell(\mu, \tau, \phi) = -\log(f(Y | \theta(\mu(\mathbf{x}) + \tau(\mathbf{x})w), \phi))$$

with f as the conditional density of an exponential family distribution

$$f(Y | \theta(\mu(\mathbf{x}) + \tau(\mathbf{x})w), \phi).$$

Model-based trees and forests (Zeileis *et al.* 2008; Seibold *et al.* 2016, 2018) jointly estimate the prognostic effect $\mu(\mathbf{x})$ and the predictive effect $\tau(\mathbf{x})$. The procedure simultaneously minimizes the negative log-likelihood with respect to $\mu(\mathbf{x})$ and $\tau(\mathbf{x})$. In each node of the model-based forest, μ , τ , and potentially ϕ are estimated by minimizing

$$\ell(\mu, \tau, \phi) = -\log(f(Y | \theta(\mu + \tau w), \phi)) \quad (7)$$

and regressing the bivariate gradient

$$\left. \frac{\partial \ell(\mu, \tau, \phi)}{\partial(\mu, \tau)} \right|_{\hat{\mu}, \hat{\tau}, \hat{\phi}}$$

on \mathbf{x} . This means that one is not explicitly looking for changes in the scale parameter ϕ , but this could be implemented by looking at the three-variate gradient

$$\left. \frac{\partial \ell(\mu, \tau, \phi)}{\partial(\mu, \tau, \phi)} \right|_{\hat{\mu}, \hat{\tau}, \hat{\phi}}$$

for example, in a heteroscedastic normal linear model

$$(Y | \mathbf{X} = \mathbf{x}, W = w) = \mu(\mathbf{x}) + \tau(\mathbf{x})w + \phi(\mathbf{x})Z.$$

After the tree fitting phase, a HTE is estimated with equation (4) with $\ell(\mu, \tau, \phi)$ of equation (7) as the corresponding loss function.

Thus, model-based forests can be directly applied to estimate HTEs on relative scales for binary outcomes (binary logistic or probit regression, for example), counts (Poisson or quasi-Poisson regression), or continuous outcomes where a multiplicative effect is of interest (normal model with log-link).

2.3. Transformation models

More complex responses like ordered categorical or time-to-event outcomes are not covered by generalized linear models but can be analysed using transformation models; corresponding

model-based forests for survival analysis have been introduced by [Korepanova et al. \(2020\)](#). For some at least ordered outcome Y , we write the conditional distribution function as

$$\mathbb{P}(Y \leq y \mid \mathbf{X} = \mathbf{x}, W = w) = F(h(y) - \underbrace{(\mu(\mathbf{x}) + \tau(\mathbf{x})w)}_{=: \eta_w(\mathbf{x})}). \quad (8)$$

The transformation function h is monotone non-decreasing and the inverse link function F governs the interpretability of τ as log-odds ratios ($F = \text{logit}^{-1}$), log-hazard ratios ($F = \text{cloglog}^{-1}$), log-reverse time hazard ratios ($F = \text{loglog}^{-1}$), or shift effects ($F = \Phi$, the cumulative distribution function of the standard normal). The shift term $\eta_w(\mathbf{x})$ differs between the two treatment groups $w \in \{0, 1\}$. The distribution functions of the potential outcomes are $F(h(y) - \mu(\mathbf{x}))$ for $Y(0)$ and $F(h(y) - \mu(\mathbf{x}) - \tau(\mathbf{x}))$ for $Y(1)$. The negative log-likelihood of a discrete or interval-censored observation $(\underline{y}, \bar{y}]$ (where \underline{y} is the lower interval bound, \bar{y} is the upper) is

$$\begin{aligned} \ell_{\text{Trafo}}(h, \mu, \tau) &= -\log(\mathbb{P}(\underline{y} < Y \leq \bar{y} \mid \mathbf{X} = \mathbf{x}, W = w)) \\ &= -\log(F(h(\bar{y}) - \mu(\mathbf{x}) - \tau(\mathbf{x})w) - F(h(\underline{y}) - \mu(\mathbf{x}) - \tau(\mathbf{x})w)). \end{aligned}$$

For a continuous datum $y \in \mathbb{R}$, we obtain

$$\ell_{\text{Trafo}}(h, \mu, \tau) = -\{\log(F'(h(y) - \mu(\mathbf{x}) - \tau(\mathbf{x})w)) + \log(h'(y))\};$$

details are given in [Hothorn, Möst, and Bühlmann \(2018\)](#). Transformation forests apply the model-based recursive partitioning principle and estimate τ in each node along with the transformation function h (a “nuisance” parameter) by minimising $\ell_{\text{Trafo}}(h, \mu \equiv 0, \tau)$ ([Hothorn and Zeileis 2021b](#)). Because h contains an intercept term, the parameter μ is not identified. We thus estimate the model under the constraint $\mu \equiv 0$. Variable and cut-points are selected using the bivariate gradient

$$\left. \frac{\partial \ell_{\text{Trafo}}(h, \mu \equiv 0, \tau)}{\partial(\mu, \tau)} \right|_{\mu=0, \hat{\tau}}$$

This model family includes proportional odds logistic regression (for ordered categorical, count or continuous outcomes), Box-Cox type models, Cox proportional hazards model, Weibull proportional hazards models for discrete and continuous outcomes, reverse time proportional hazards models relying on Lehmann alternatives, and many more ([Hothorn et al. 2018](#)). Forests for ordinal outcomes were evaluated by [Buri and Hothorn \(2020\)](#), and a general approach to “transformation forests” is described in [Hothorn and Zeileis \(2021b\)](#).

Application of the ideas underlying model-based forests allows HTEs to be estimated for such outcomes under all types of random censoring and truncation ([Korepanova et al. 2020](#)). For example, for Weibull distributed outcomes under right censoring, $h(y) = \nu_1 + \nu_2 \log(y)$ is chosen for the conditional distribution function in equation (8) ([Hothorn et al. 2018](#)).

In this case, we define Y as the event time, C as the censoring time and $T = \min(Y, C)$ as the observed time. For identification of $\tau(\mathbf{x})$ under potential censoring, the following assumption must hold ([Cui et al. 2022](#)):

Assumption 1 (Ignorable censoring). *Censoring time C is independent of survival time Y conditional on treatment indicator W and covariates X*

$$(Y(0), Y(1)) \perp\!\!\!\perp C \mid \mathbf{X} = \mathbf{x}, W = w.$$

An important special case represents the Cox proportional hazards model, where the profile likelihood over the baseline hazard function defines the partial log-likelihood $\ell_{\text{PL}}(\mu, \tau)$ with $\mu \equiv 0$. The scores with respect to the constant $\mu \equiv 0$ are known as martingale residuals. Model-based forests for such models, and extensions to time-varying prognostic and predictive effects, are discussed in [Korepanova et al. \(2020\)](#).

2.4. Noncollapsibility

As mentioned in the introduction, one problem with the Cox model is that misspecifications of prognostic effects $\mu(\mathbf{x})$ lead to biased estimates such that the estimated hazard ratios cannot be interpreted causally. This issue arises from the noncollapsibility of the Cox model, the notion of which is characterized by the fact that in these models, the mean of the conditional effect estimates defined over covariates \mathbf{X} does not coincide with the marginal effect over \mathbf{X} . Because the noncollapsibility of the Cox model arises from its nonadditivity of the hazard function, models such as the Weibull model do not suffer from this issue because they satisfy the additivity condition. Consequently, misspecifications of prognostic effects do not affect treatment effect estimates ([Aalen et al. 2015](#)).

The noncollapsibility issue is not limited to the Cox model but also affects members of the exponential family without identity or linear link functions. Without adjustments, effect estimates can only be interpreted causally if there is no treatment effect ($\tau \equiv 0$) or there are no prognostic covariates ([Daniel, Zhang, and Farewell 2021](#)).

If this is not the case, specific methods are needed; ignoring the estimation of $\mu(\mathbf{x})$ at all and only focusing on $\tau(\mathbf{x})$ does not solve the problem. Conditioning on available prognostic variables is a common solution and is already applied by model-based forests, because they estimate both the prognostic effect $\mu(\mathbf{x})$ and $\tau(\mathbf{x})$. The ensemble of trees used to estimate these effects provides a high degree of flexibility and might therefore retain some of the potential complexity in the underlying $\mu(\mathbf{x})$ to mitigate misspecification. Whether conditioning resolves the non-collapsibility issue depends heavily on the assumption that all prognostic variables are known which is often not the case in the real world ([Aalen et al. 2015](#)).

For members of the exponential family and the Cox model, [Gao and Hastie \(2022\)](#) derived a method to account for noncollapsibility in the context of observational data with confounding effects. While we consider the noncollapsibility issue beyond the scope of this work, we briefly review the work of Gao and Hastie and discuss its applicability to model-based forests in Section A of the Supplementary Material.

3. Model-based forests for observational studies

In the previous section, we described model-based forests in the randomized setting under the assumption that $\pi(\mathbf{x}) = \pi$. In observational studies in which the treatment group assignment is not under the control of the researcher, the propensity score (and therefore, the probability of being in the treatment group) often depends on covariates \mathbf{x}

$$\pi(\mathbf{x}) := \mathbb{P}(W = 1 \mid \mathbf{X} = \mathbf{x}) = \mathbb{E}(W \mid \mathbf{X} = \mathbf{x}). \quad (9)$$

In this case, confounding effects could bias the estimation of treatment effects $\tau(\mathbf{x})$, and stricter assumptions are necessary in order to interpret $\tau(\mathbf{x})$ causally ([Rosenbaum and Rubin 1983](#)).

Assumption 2 (Ignorability/Unconfoundedness). *The treatment assignment is independent of the potential outcomes conditional on covariates \mathbf{x}*

$$(Y(0), Y(1)) \perp\!\!\!\perp W \mid \mathbf{X} = \mathbf{x}.$$

Assumption 3 (Positivity). *The propensity score $\pi(\mathbf{x})$ must be bounded away from 0 and 1*

$$0 < \pi(\mathbf{x}) = \mathbb{P}(W = 1 \mid \mathbf{X} = \mathbf{x}) = \mathbb{E}(W \mid \mathbf{X} = \mathbf{x}) < 1.$$

Assumption 2 could be violated by an unmeasured confounder, while Assumption 3 could be violated if all observations in a certain group (defined via \mathbf{x}) are in the treatment group.

Dandl *et al.* (2022) showed for mean regression models that model-based forests are not robust to confounding effects and need further adaptations to estimate causal effects in case of observational data. One strategy for dealing with confounding effects is the orthogonalization strategy originally introduced by Robinson (1988), which has received considerable attention in recent years (Chernozhukov *et al.* 2018; Athey *et al.* 2019; Nie and Wager 2021). The reformulation of the linear model

$$(Y \mid \mathbf{X} = \mathbf{x}) = \mu(\mathbf{x}) + \tau(\mathbf{x})W + \phi Z \quad (10)$$

to

$$\begin{aligned} (Y \mid \mathbf{X} = \mathbf{x}) &= m(\mathbf{x}) - m(\mathbf{x}) + \mu(\mathbf{x}) + \tau(\mathbf{x})W + \phi Z \\ &= m(\mathbf{x}) + \tau(\mathbf{x})(W - \pi(\mathbf{x})) + \phi Z \end{aligned} \quad (11)$$

given the conditional mean function

$$m(\mathbf{x}) := \mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}) = \mu(\mathbf{x}) + \tau(\mathbf{x})\pi(\mathbf{x}), \quad (12)$$

motivates this approach (Dandl *et al.* 2022).

Overall, the orthogonalization strategy consists of two steps: First, nuisance parameters $m(\mathbf{x}) = \mathbb{E}(Y \mid \mathbf{X} = \mathbf{x})$ and $\pi(\mathbf{x}) = \mathbb{P}(W = 1 \mid \mathbf{X} = \mathbf{x})$ are estimated. Originally, Robinson (1988) used kernel estimators, but any machine learning method could be employed (Chernozhukov *et al.* 2018; Nie and Wager 2021). Regressing $Y - \hat{m}(\mathbf{x})$ on $W - \hat{\pi}(\mathbf{x})$ then yields unbiased estimates for $\tau(\mathbf{x})$. Subtracting $\hat{m}(\mathbf{x})$ and $\hat{\pi}(\mathbf{x})$ from Y and W , respectively, partially eliminates the association between \mathbf{X} and Y and between \mathbf{X} and W , respectively. The orthogonalization strategy has the distinct advantage over other methods against confounding – such as inverse propensity weighting and matching – that it is stable for extreme propensity scores and forgoes stratification (Gao and Hastie 2022).

Robinson (1988) and Chernozhukov *et al.* (2018) use parametric models to estimate treatment effects based on residualized W and Y , but these models could be replaced by non-parametric or local parametric models (Nie and Wager 2021; Wager and Athey 2018) – such as model-based forests. For mean regression, Dandl *et al.* (2022) adapted the orthogonalization strategy to model-based forests. Their approach closely follows causal forests, which were the first to combine the orthogonalization strategy with tree-based estimators for $\tau(\mathbf{x})$.

Gao and Hastie (2022) proposed extensions of Robinson’s strategy to members of the exponential family and the Cox model, where DINA (6) is of interest. Gao and Hastie (2022) assume $\tau(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\beta}$ and use parametric models to estimate $\tau(\mathbf{x})$, but they conclude that

non-parametric or local parametric models could be applied instead. We review model-based forests in combination with linear models for observational data in the next section and summarize the idea by Gao and Hastie (2022) in Section 3.2. On this basis, we assess how the orthogonalization strategy could be employed in model-based forests beyond mean regression with generalized linear models and transformation models as base models.

3.1. Review of Dandl et al. (2022)

As noted above, Athey et al. (2019) were the first to combine the orthogonalization strategy of Robins with tree-based estimators to estimate $\tau(\mathbf{x})$. First, the marginal model $m(\mathbf{x}) = \mathbb{E}(Y | \mathbf{X} = \mathbf{x})$ and propensity score $\pi(\mathbf{x}) = \mathbb{E}(W | \mathbf{X} = \mathbf{x})$ are estimated by regression forests. Afterwards, causal forests estimate individual treatment effects $\tau(\mathbf{x})$ in the model

$$(Y | \mathbf{X} = \mathbf{x}, W = w) = \hat{m}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x})) + \phi Z \quad (13)$$

using the “locally centered” outcomes $Y - \hat{m}(\mathbf{x})$ and treatment indicators $W - \hat{\pi}(\mathbf{x})$.

Equation (13) shows that causal forests and model-based forests share common foundations for mean regression. The main difference is that the splitting scheme of model-based forests allows splitting according to heterogeneity in both treatment and prognostic effects, whereas causal forests only split with respect to heterogeneity in treatment effects (in equation (11), $\mu(\mathbf{x})$ cancels out).

Dandl et al. (2022) identified which elements of both approaches lead to improved performance in randomized trials and observational studies by defining and evaluating blended versions of model-based forests and causal forests:

- (1) mob(\hat{W}, \hat{Y}), which applies model-based forests to the model

$$\mathbb{E}(Y | \mathbf{X} = \mathbf{x}, W = w) = \hat{m}(\mathbf{x}) + \tilde{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x})),$$

i.e. after centering the treatment indicator w and the outcome Y . Both parameters $\tilde{\mu}$ and τ are estimated simultaneously.

- (2) mob(\hat{W}), which applies model-based forests to the model

$$\mathbb{E}(Y | \mathbf{X} = \mathbf{x}, W = w) = \mu(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x})),$$

i.e. after only centering the treatment indicator w but *not* outcome Y . Both μ and τ are estimated.

- (3) cfmob, a method that applies model-based forests to the model

$$\mathbb{E}(Y | \mathbf{X} = \mathbf{x}, W = w) = \hat{m}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x})),$$

i.e. after only centering the treatment indicator w and splitting only according to $\hat{\pi}$. That is, only the parameters τ are estimated in this variant.

Their blended approaches competed with the original implementations of (uncentered) model-based forests and causal forests in an extensive simulation study. In case of confounding, the authors identified local centering of treatment indicator w and simultaneous estimation

of both predictive *and* prognostic effects of the treatment indication ($\text{mob}(\hat{W})$) as the key driver for good performance. Additionally, centering Y ($\text{mob}(\hat{W}, \hat{Y})$) is recommended, since it further improved performances in some cases. Splitting only according to $\hat{\tau}$ but not $\hat{\mu}$ (cfmob) resulted in lower performance. Even for settings with confounding, the performance of cfmob was inferior to that of uncentered model-based forests.

3.2. Review of Gao and Hastie (2022)

Robinson (1988) derived the orthogonalization strategy only for semi-parametric additive models with $Y \in \mathbb{R}$. Gao and Hastie (2022) extended the idea to a broader class of distributions including the exponential family and Cox' model.

Local centering of the treatment indicator works analogously to mean regression. First, propensity scores $\pi(\mathbf{x}) = \mathbb{P}(W | \mathbf{X} = \mathbf{x})$ are estimated. The effects of the covariates \mathbf{X} on the treatment assignment are then regressed out by subtracting $\hat{\pi}(\mathbf{x})$ from W .

Orthogonalization of Y is not straightforward due to the link function that relates the linear predictor $\eta_w(\mathbf{x})$ in equation (5) to the outcome Y . To understand how Gao and Hastie derived $m(\mathbf{x})$ to center Y , we consider equation (10) as a model of the exponential family with identity link function g . Now we can rewrite equation (12) to

$$\begin{aligned} g(\mathbb{E}(Y | \mathbf{X} = \mathbf{x})) &= \mathbb{E}_W(g(\mathbb{E}(Y | \mathbf{X} = \mathbf{x}, W = w))) \\ &= \pi(\mathbf{x}) \underbrace{(\mu(\mathbf{x}) + \tau(\mathbf{x}))}_{=\eta_1(\mathbf{x})} + (1 - \pi(\mathbf{x})) \underbrace{\mu(\mathbf{x})}_{=\eta_0(\mathbf{x})} \\ &= \mu(\mathbf{x}) + \pi(\mathbf{x})\tau(\mathbf{x}) = m(\mathbf{x}). \end{aligned}$$

Similarly, we derive $g(\mathbb{E}(Y | \mathbf{X} = \mathbf{x}))$ for all other distributions of the exponential family by

$$m(\mathbf{x}) = \pi(\mathbf{x})\eta_1(\mathbf{x}) + (1 - \pi(\mathbf{x}))\eta_0(\mathbf{x}). \quad (14)$$

We can regard the estimated $m(\mathbf{x})$ as an offset in the linear predictor

$$\hat{m}(\mathbf{x}) + \tau(\mathbf{x})(W - \hat{\pi}(\mathbf{x})).$$

Note that equation (14) states that (only) for the Gaussian distribution we can directly estimate $m(\mathbf{x}) = \mathbb{E}(Y | \mathbf{X} = \mathbf{x})$ without estimating $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$. We can also derive $\hat{m}(\mathbf{x})$ for transformation models based on the definition of η_0 and η_1 in equation (8). As mentioned in Section 2.4, compared to the difference in conditional means, the difference in natural parameters additionally suffers from the noncollapsibility issue (Greenland, Pearl, and Robins 1999). Gao and Hastie (2022) also extend the Robinson strategy to tackle not only the confounding but also the noncollapsibility issue for members of the exponential family (without a linear or log link function, otherwise confounding is not an issue) and the Cox model. While the noncollapsibility issue is beyond the scope of this work, we briefly summarize and discuss the work of Gao and Hastie in Section A of the Supplementary Material.

3.3. Novel model-based forests for observational data

As stated above, our main goal is to assess how the orthogonalization strategy proposed for continuous outcomes could be extended to models beyond mean regression, specifically

generalized linear models and transformation models. Based on Dandl *et al.* (2022) and Gao and Hastie (2022) we propose two different versions of model-based forests, which should be more robust against confounding. Following Dandl *et al.* (2022), we formulate research questions for these versions, which we aim to answer empirically in Section 4. An overview of all proposed versions is given in Table 1.

The first version of model-based forests directly applies Robinson’s orthogonalization strategy: First, we estimate propensities $\pi(\mathbf{x})$ as well as $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$ to derive $\hat{m}(\mathbf{x})$. Then, we update the linear predictor of equation (5) by centering W by $\hat{\pi}(\mathbf{x})$ and by adding the offset $\hat{m}(\mathbf{x})$. For generalized linear models, we obtain

$$g(\mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}, W = w)) = \hat{m}(\mathbf{x}) + \tilde{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$$

and for the conditional distribution function of equation (8) in case of transformation models

$$F[h(y) - \{\hat{m}(\mathbf{x}) + \tilde{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))\}].$$

Based on the updated models, both prognostic and predictive effects $\tilde{\mu}(\mathbf{x})$ and $\tau(\mathbf{x})$ are simultaneously estimated by model-based forests.

In the simulation study and practical example in Sections 4 and 5, we use regression forests to estimate $\pi(\mathbf{x})$ and gradient boosting machines (with tailored loss functions) to estimate η_0 and η_1 . In the following, we denote this version of model-based forests as *Robinson* in recognition of Robinson (1988) while model-based forests without centering W and without offset $\hat{m}(\mathbf{x})$ are called *Naive*.

RQ 1 *To what extent does centering W by $\hat{\pi}(\mathbf{x})$ and including $\hat{m}(\mathbf{x})$ as an offset affect the performance of model-based forests in the presence of confounding?*

Similar to Dandl *et al.* – who saw an improvement in performance when only centering W (compared to the naive model-based forests) – we define an approach called *Robinson_W* that applies model-based forests to models with linear predictors

$$\mu(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x})).$$

RQ 2 *Do centered treatment indicator model-based forests perform better than uncentered model-based forests in the presence of confounding?*

RQ 3 *Are model-based forests with centered treatment indicators relevantly outperformed by model-based forests with $\hat{m}(\mathbf{x})$ as an additional offset in the presence of confounding?*

4. Empirical evaluation

We evaluated the performance of our proposed model-based forest versions (Table 1) in a simulation study. The study includes different outcome types, different predictive and prognostic effects, and a varying number of observations and covariates. Model-based forests were

Method	Linear Predictor	Definitions
Naive	$\mu(\mathbf{x}) + \tau(\mathbf{x}) w$	
Robinson _{\hat{W}}	$\mu(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$	$\pi(\mathbf{x}) = \mathbb{P}(W = 1 \mathbf{X} = \mathbf{x})$
Robinson	$\hat{m}(\mathbf{x}) + \hat{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$	$m(\mathbf{x}) = \pi(\mathbf{x})\eta_1(\mathbf{x}) - (1 - \pi(\mathbf{x}))\eta_0(\mathbf{x})$

Table 1: Overview of proposed model-based forest versions.

fitted with the **model4you** R add-on package (Seibold, Zeileis, and Hothorn 2019). Similar to Dandl *et al.* (2022), we base our study settings on the four setups (A, B, C and D) of Nie and Wager (2021). In addition, in Section B of the Supplementary Material, we show the results for simulation settings first proposed by Wager and Athey (2018) and later reused by Athey *et al.* (2019).

4.1. Data generating process

Given $P = \{10, 20\}$, for Setup A, we sampled $\mathbf{X} \sim U([0, 1]^P)$. For all other setups, we used $\mathbf{X} \sim N(0, \mathbf{1}_{P \times P})$. The treatment indicator was binomially distributed with $W | \mathbf{X} = \mathbf{x} \sim B(1, \pi(\mathbf{x}))$. The propensity function $\pi(\mathbf{x})$ differed for the four considered setups:

$$\pi(\mathbf{x}) = \begin{cases} \pi_A(x_1, x_2) = \max\{0.1, \min\{\sin(\pi x_1 x_2), 1 - 0.1\}\} \\ \pi_B \equiv 0.5 \\ \pi_C(x_2, x_3) = 1/(1 + \exp(x_2 + x_3)) \\ \pi_D(x_1, x_2) = 1/(1 + \exp(-x_1) + \exp(-x_2)). \end{cases}$$

$\pi(\mathbf{x}) \equiv 0.5$ in Setup B implies a randomized study. The treatment effect function $\tau(\cdot)$ and the prognostic effect function $\mu(\cdot)$ also differed between the setups

$$\tau(\mathbf{x}) = \begin{cases} \tau_A(x_1, x_2) = (x_1 + x_2)/2 \\ \tau_B(x_1, x_2) = x_1 + \log(1 + \exp(x_2)) \\ \tau_C \equiv 1 \\ \tau_D(x_1, x_2, x_3, x_4, x_5) = \max\{x_1 + x_2 + x_3, 0\} - \max\{x_4 + x_5, 0\}. \end{cases}$$

$$\mu(\mathbf{x}) = \begin{cases} \mu_A(x_1, x_2, x_3, x_4, x_5) = \sin(\pi x_1 x_2) + 2(x_3 - 0.5)^2 + x_4 + 0.5x_5 \\ \mu_B(x_1, x_2, x_3) = \max\{x_1 + x_2, x_3, 0\} + \max\{x_4 + x_5, 0\} \\ \mu_C(x_1, x_2, x_3) = 2 \log(1 + \exp(x_1 + x_2 + x_3)) \\ \mu_D(x_1, x_2, x_3, x_4, x_5) = (\max\{x_1 + x_2 + x_3, 0\} + \max\{x_4 + x_5, 0\})/2. \end{cases}$$

Setup A has extensive confounding that must be eliminated before estimating an easily predictable treatment effect function $\tau(\mathbf{x})$. Setup B needs no confounding adjustment for reliable estimation of τ . Although Setup C contains strong confounding, the propensity score function is easier to estimate than the prognostic effect, while the treatment effect is constant. In Setup D, the treatment and control arms are unrelated, and therefore, learning the conditional expected outcomes of both arms jointly is not beneficial (Nie and Wager 2021; Dandl *et al.* 2022).

We studied four different simulation models

$$(Y \mid \mathbf{X} = \mathbf{x}, W = w) \sim \begin{cases} \text{N}(\mu(\mathbf{x}) + \tau(\mathbf{x})(w - 0.5), 1) & (15a) \\ \text{B}(1, \text{expit}(\mu(\mathbf{x}) + \tau(\mathbf{x})(w - 0.5))) & (15b) \\ \text{M with } \log(O(y_k \mid \mathbf{x}, w)) = \vartheta_k - \mu(\mathbf{x}) - \tau(\mathbf{x})(w - 0.5) & (15c) \\ \text{W with } \log(H(y \mid \mathbf{x}, w)) = 2 \log(y) - \mu(\mathbf{x}) - \tau(\mathbf{x})(w - 0.5) & (15d) \end{cases}$$

Model (15a) is a normal linear regression model, model (15b) is a binary logistic regression model, model (15c) is a 4-nomial model with log-odds function $\vartheta_k - \mu(\mathbf{x}) - \tau(\mathbf{x})(w - 0.5)$ with threshold parameters $\vartheta_k = \text{logit}(k/4)$ for $k = 1, 2, 3$, and model (15d) is a Weibull model with log-cumulative hazard function $2 \log(y) - \mu(\mathbf{x}) - \tau(\mathbf{x})(w - 0.5)$. We added 50% random right-censoring to the Weibull-generated data. Additionally, we applied a Cox proportional hazards model to the Weibull data to determine if the performance of model-based forests degrades when the forests do not take the true underlying model as their base model.

Due to $w - 0.5$ in all scenarios, half of the (negative) predictive effect $\tau(\mathbf{x})$ was added to the prognostic effect. We refer to the implied scenario – where one variable which is both prognostic (impact in $\mu(\mathbf{x})$) and predictive (impact in $\tau(\mathbf{x})$) exists – as overlay. Apart from Setup C in which the treatment effect is constant and independent of any covariate, overlay was present for all scenarios.

Like Dandl *et al.* (2022), we compared all study settings and outcome types for a varying number of samples $N \in \{800, 1600\}$ and dimensions $P \in \{10, 20\}$. All model-based forests were grown with the same hyperparameter options specified in Section 7. We used random forests as implemented in the **grf** package to estimate $\pi(\mathbf{x})$ for centering W (Tibshirani, Athey, Sverdrup, and Wager 2021). To estimate $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$ to derive $\hat{m}(\mathbf{x})$, we relied on different tree-based estimators depending on the outcome type. For normally distributed outcomes (models (15a)), we used **grf** regression forests (Tibshirani *et al.* 2021). For all other outcomes, we relied on gradient boosting machines (with adapted loss functions) as implemented in **mboost** and **gbm** (Hothorn, Bühlmann, Kneib, Schmid, and Hofner 2021b; Greenwell, Boehmke, Cunningham, and Developers 2020). The employed distribution varied depending on the outcome type.

In accordance with Dandl *et al.* (2022), we evaluated the models with respect to the mean squared error $\mathbb{E}_{\mathbf{X}}\{(\hat{\tau}(\mathbf{X}) - \tau(\mathbf{X}))^2\}$ on a test sample of size 1000. The results are shown in Figure 1 and were statistically analyzed by means of a normal linear mixed model with a log-link. The model explained the estimated mean squared error for $\hat{\tau}(\mathbf{x})$ by a four-way interaction of the data generating process, sample size N , dimension P , and random forest variant. We estimated the mean squared error ratios between different model-based forest versions according to the two research questions stated in Section 3.3. The corresponding tables are given in Tables 2 to 4.

4.2. Results

The results for the normal distribution coincide with the results obtained by Dandl *et al.* (2022) summarized in Section 3.1. To some degree, they also hold for the other distributions. The boxplots are not directly comparable between different data generating processes because of different signal-to-noise ratios. In general, a more informative outcome (binary < ordered < right-censored < exact normal), more data (higher N), and less noise (lower P) leads to better results. Using a Cox model compared to a Weibull model (last two rows of Figure 1)

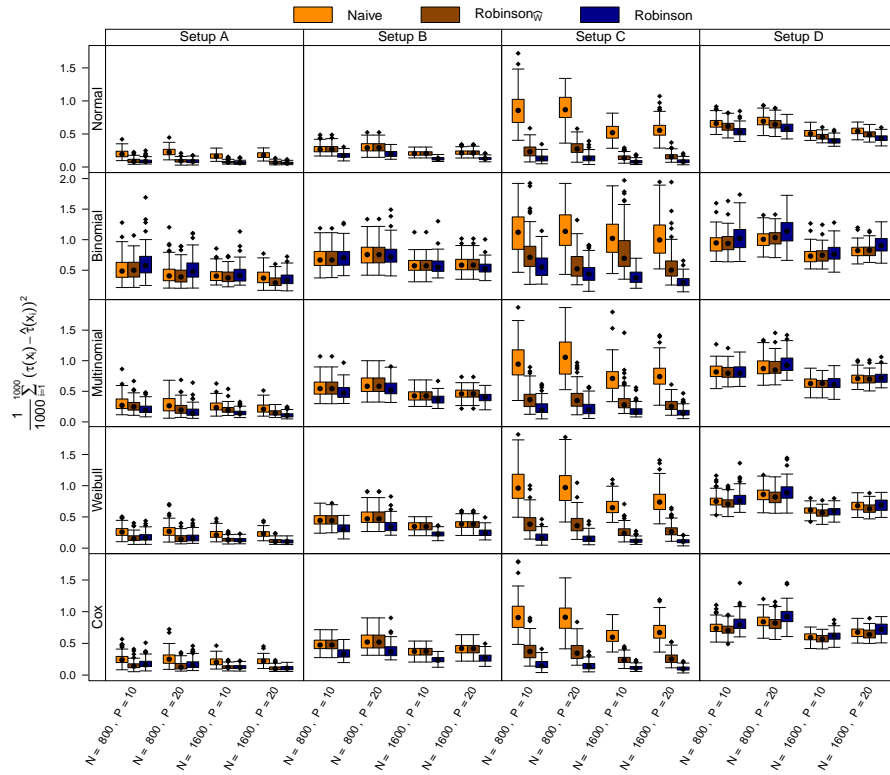


Figure 1: Model-based forest results for the empirical study (Section 4), Cox means a Cox model applied to the Weibull data. For the Weibull and Cox model, treatment effects $\tau(\mathbf{x})$ are estimated as conditional log hazard ratios. Direct comparison of model-based forests without centering (*Naive*), model-based forests with local centering according to [Robinson \(1988\)](#) of Y and W (originally proposed) (*Robinson*) or only of W (*Robinson $_{\hat{W}}$*).

DGP	N	P	Mean squared error ratio for RQ 1 : Robinson vs. Naive				
			Normal	Binomial	Multinomial	Weibull	Cox
Setup A	800	10	<i>0.465 (0.421, 0.512)</i>	1.173 (1.045, 1.316)	<i>0.690 (0.629, 0.758)</i>	<i>0.672 (0.609, 0.742)</i>	<i>0.712 (0.650, 0.781)</i>
	20	<i>0.396 (0.359, 0.438)</i>	1.161 (1.014, 1.330)	<i>0.600 (0.540, 0.666)</i>	<i>0.605 (0.547, 0.669)</i>	<i>0.654 (0.596, 0.718)</i>	
1600	10	<i>0.414 (0.362, 0.474)</i>	<i>1.042 (0.892, 1.216)</i>	<i>0.582 (0.512, 0.662)</i>	<i>0.580 (0.508, 0.663)</i>	<i>0.589 (0.519, 0.669)</i>	
	20	<i>0.341 (0.295, 0.395)</i>	<i>0.898 (0.751, 1.075)</i>	<i>0.503 (0.428, 0.591)</i>	<i>0.471 (0.405, 0.548)</i>	<i>0.495 (0.430, 0.570)</i>	
Setup B	800	10	<i>0.643 (0.607, 0.681)</i>	<i>1.021 (0.929, 1.121)</i>	<i>0.868 (0.830, 0.907)</i>	<i>0.692 (0.653, 0.733)</i>	<i>0.703 (0.668, 0.739)</i>
	20	<i>0.658 (0.625, 0.693)</i>	<i>0.977 (0.894, 1.067)</i>	<i>0.906 (0.870, 0.943)</i>	<i>0.716 (0.681, 0.754)</i>	<i>0.731 (0.699, 0.763)</i>	
1600	10	<i>0.603 (0.557, 0.653)</i>	<i>0.981 (0.873, 1.101)</i>	<i>0.852 (0.803, 0.903)</i>	<i>0.657 (0.607, 0.710)</i>	<i>0.656 (0.612, 0.702)</i>	
	20	<i>0.588 (0.544, 0.636)</i>	<i>0.912 (0.811, 1.026)</i>	<i>0.869 (0.822, 0.917)</i>	<i>0.648 (0.603, 0.697)</i>	<i>0.653 (0.614, 0.695)</i>	
Setup C	800	10	<i>0.153 (0.144, 0.163)</i>	<i>0.474 (0.432, 0.520)</i>	<i>0.250 (0.233, 0.268)</i>	<i>0.174 (0.160, 0.189)</i>	<i>0.176 (0.162, 0.191)</i>
	20	<i>0.156 (0.147, 0.166)</i>	<i>0.359 (0.322, 0.400)</i>	<i>0.219 (0.204, 0.235)</i>	<i>0.157 (0.142, 0.172)</i>	<i>0.161 (0.146, 0.177)</i>	
1600	10	<i>0.154 (0.139, 0.171)</i>	<i>0.361 (0.316, 0.412)</i>	<i>0.260 (0.238, 0.284)</i>	<i>0.181 (0.160, 0.206)</i>	<i>0.187 (0.165, 0.212)</i>	
	20	<i>0.157 (0.142, 0.173)</i>	<i>0.300 (0.256, 0.351)</i>	<i>0.215 (0.195, 0.238)</i>	<i>0.147 (0.129, 0.169)</i>	<i>0.152 (0.139, 0.174)</i>	
Setup D	800	10	<i>0.818 (0.801, 0.835)</i>	1.109 (1.037, 1.187)	<i>0.996 (0.968, 1.026)</i>	1.036 (1.008, 1.065)	1.085 (1.057, 1.113)
	20	<i>0.851 (0.835, 0.867)</i>	1.126 (1.058, 1.199)	1.054 (1.028, 1.082)	1.055 (1.029, 1.081)	1.099 (1.075, 1.124)	
1600	10	<i>0.783 (0.762, 0.805)</i>	<i>1.075 (0.985, 1.175)</i>	<i>0.994 (0.957, 1.032)</i>	<i>0.968 (0.934, 1.004)</i>	<i>1.029 (0.995, 1.063)</i>	
	20	<i>0.803 (0.783, 0.824)</i>	1.131 (1.046, 1.223)	<i>1.016 (0.983, 1.051)</i>	<i>1.021 (0.989, 1.053)</i>	1.076 (1.046, 1.108)	

Table 2: Results of **RQ 1** for the experimental setups in Section 4. Comparison of mean squared errors for $\hat{\tau}(\mathbf{x})$ in the different scenarios. Estimates and simultaneous 95 % confidence intervals were obtained from a normal linear mixed model with log-link. Cells printed in bold font correspond to a superior reference of the *Naive* model-based forests, and cells printed in italics indicate an inferior reference.

DGP	N	P	Mean squared error ratio for RQ 2 : Robinson $_{\hat{W}}$ vs. Naive				
			Normal	Binomial	Multinomial	Weibull	Cox
Setup A	800	10	<i>1.029 (0.910, 1.164)</i>	<i>0.820 (0.729, 0.922)</i>	1.259 (1.142, 1.388)	<i>0.924 (0.820, 1.042)</i>	<i>0.844 (0.752, 0.947)</i>
	20	<i>1.060 (0.933, 1.204)</i>	<i>0.784 (0.679, 0.905)</i>	1.282 (1.144, 1.437)	<i>0.935 (0.825, 1.060)</i>	<i>0.835 (0.740, 0.942)</i>	
1600	10	<i>1.126 (0.953, 1.330)</i>	<i>0.915 (0.781, 1.072)</i>	1.370 (1.194, 1.571)	<i>1.067 (0.911, 1.250)</i>	<i>1.015 (0.870, 1.184)</i>	
	20	<i>1.163 (0.970, 1.395)</i>	<i>0.887 (0.726, 1.084)</i>	1.302 (1.086, 1.561)	<i>1.063 (0.881, 1.283)</i>	<i>0.994 (0.831, 1.188)</i>	
Setup B	800	10	1.555 (1.468, 1.647)	<i>0.980 (0.892, 1.077)</i>	1.152 (1.102, 1.205)	1.445 (1.363, 1.531)	1.423 (1.353, 1.496)
	20	1.520 (1.444, 1.600)	<i>1.024 (0.938, 1.119)</i>	1.104 (1.060, 1.150)	1.396 (1.327, 1.469)	1.368 (1.309, 1.430)	
1600	10	1.658 (1.530, 1.796)	<i>1.019 (0.907, 1.144)</i>	1.174 (1.107, 1.245)	1.524 (1.409, 1.648)	1.525 (1.424, 1.634)	
	20	1.700 (1.574, 1.837)	<i>1.097 (0.975, 1.233)</i>	1.151 (1.090, 1.216)	1.542 (1.435, 1.657)	1.532 (1.440, 1.629)	
Setup C	800	10	1.871 (1.743, 2.009)	1.377 (1.243, 1.526)	1.577 (1.456, 1.708)	2.331 (2.128, 2.553)	2.388 (2.182, 2.614)
	20	2.081 (1.944, 2.226)	1.294 (1.138, 1.470)	1.718 (1.588, 1.859)	2.565 (2.318, 2.839)	2.611 (2.363, 2.886)	
1600	10	1.774 (1.573, 2.001)	2.619 (2.288, 2.999)	1.759 (1.594, 1.942)	2.198 (1.920, 2.517)	2.141 (1.874, 2.446)	
	20	1.817 (1.629, 2.026)	1.800 (1.512, 2.144)	1.675 (1.494, 1.877)	2.541 (2.203, 2.932)	2.566 (2.228, 2.956)	
Setup D	800	10	1.136 (1.113, 1.161)	<i>0.910 (0.851, 0.974)</i>	<i>0.992 (0.964, 1.021)</i>	<i>0.916 (0.890, 0.942)</i>	<i>0.883 (0.860, 0.906)</i>
	20	1.098 (1.077, 1.120)	<i>0.898 (0.844, 0.956)</i>	<i>0.942 (0.918, 0.966)</i>	<i>0.909 (0.886, 0.932)</i>	<i>0.881 (0.861, 0.901)</i>	
1600	10	1.147 (1.114, 1.180)	<i>0.950 (0.871, 1.037)</i>	<i>0.994 (0.958, 1.032)</i>	<i>0.965 (0.929, 1.001)</i>	<i>0.923 (0.892, 0.954)</i>	
	20	1.126 (1.097, 1.157)	<i>0.890 (0.823, 0.961)</i>	<i>0.972 (0.940, 1.005)</i>	<i>0.922 (0.893, 0.952)</i>	<i>0.888 (0.862, 0.914)</i>	

Table 3: Results of **RQ 2** for the experimental setups in Section 4. Comparison of mean squared errors for $\hat{\tau}(\mathbf{x})$ in the different scenarios. Estimates and simultaneous 95 % confidence intervals were obtained from a normal linear mixed model with log-link. Cells printed in bold font correspond to a superior reference of the *Naive* model-based forests, and cells printed in italics indicate an inferior reference.

DGP	N	P	Mean squared error ratio for RQ 3 : Robinson vs. Robinson $_{\hat{W}}$				
			Normal	Binomial	Multinomial	Weibull	Cox
Setup A	800	10	<i>0.972 (0.859, 1.099)</i>	1.220 (1.085, 1.373)	<i>0.794 (0.720, 0.876)</i>	<i>1.082 (0.959, 1.220)</i>	1.185 (1.056, 1.329)
	20	<i>0.944 (0.831, 1.072)</i>	1.276 (1.105, 1.472)	<i>0.780 (0.696, 0.874)</i>	<i>1.070 (0.944, 1.212)</i>	1.197 (1.061, 1.351)	
1600	10	<i>0.888 (0.752, 1.049)</i>	<i>1.093 (0.933, 1.281)</i>	<i>0.730 (0.637, 0.838)</i>	<i>0.937 (0.800, 1.098)</i>	<i>0.985 (0.844, 1.149)</i>	
	20	<i>0.860 (0.717, 1.030)</i>	<i>1.127 (0.922, 1.378)</i>	<i>0.768 (0.641, 0.921)</i>	<i>0.941 (0.780, 1.135)</i>	<i>1.006 (0.841, 1.203)</i>	
Setup B	800	10	<i>0.643 (0.607, 0.681)</i>	<i>1.020 (0.929, 1.121)</i>	<i>0.868 (0.830, 0.907)</i>	<i>0.692 (0.653, 0.733)</i>	<i>0.703 (0.668, 0.739)</i>
	20	<i>0.658 (0.625, 0.692)</i>	<i>0.976 (0.894, 1.067)</i>	<i>0.906 (0.869, 0.943)</i>	<i>0.716 (0.681, 0.754)</i>	<i>0.731 (0.699, 0.764)</i>	
1600	10	<i>0.603 (0.557, 0.654)</i>	<i>0.981 (0.874, 1.102)</i>	<i>0.852 (0.803, 0.903)</i>	<i>0.656 (0.607, 0.710)</i>	<i>0.656 (0.612, 0.702)</i>	
	20	<i>0.588 (0.544, 0.635)</i>	<i>0.912 (0.811, 1.026)</i>	<i>0.869 (0.822, 0.917)</i>	<i>0.649 (0.603, 0.697)</i>	<i>0.653 (0.614, 0.695)</i>	
Setup C	800	10	<i>0.534 (0.498, 0.574)</i>	<i>0.726 (0.655, 0.804)</i>	<i>0.634 (0.586, 0.687)</i>	<i>0.429 (0.392, 0.470)</i>	<i>0.419 (0.383, 0.458)</i>
	20	<i>0.481 (0.449, 0.514)</i>	<i>0.773 (0.680, 0.878)</i>	<i>0.582 (0.538, 0.630)</i>	<i>0.390 (0.352, 0.431)</i>	<i>0.383 (0.346, 0.423)</i>	
1600	10	<i>0.564 (0.500, 0.636)</i>	<i>0.382 (0.333, 0.437)</i>	<i>0.569 (0.515, 0.628)</i>	<i>0.455 (0.397, 0.521)</i>	<i>0.467 (0.409, 0.534)</i>	
	20	<i>0.550 (0.494, 0.614)</i>	<i>0.555 (0.467, 0.661)</i>	<i>0.597 (0.533, 0.669)</i>	<i>0.393 (0.341, 0.454)</i>	<i>0.390 (0.338, 0.449)</i>	
Setup D	800	10	<i>0.880 (0.861, 0.899)</i>	1.099 (1.027, 1.175)	<i>1.008 (0.979, 1.037)</i>	1.092 (1.061, 1.123)	1.133 (1.104, 1.163)
	20	<i>0.911 (0.893, 0.929)</i>	1.113 (1.046, 1.185)	1.062 (1.035, 1.089)	1.101 (1.073, 1.128)	1.136 (1.110, 1.162)	
1600	10	<i>0.872 (0.848, 0.898)</i>	<i>1.052 (0.964, 1.148)</i>	<i>1.006 (0.969, 1.044)</i>	<i>1.037 (0.999, 1.076)</i>	<i>1.084 (1.048, 1.121)</i>	
	20	<i>0.888 (0.865, 0.912)</i>	1.124 (1.040, 1.215)	<i>1.029 (0.995, 1.064)</i>	1.085 (1.050, 1.120)	1.126 (1.094, 1.160)	

Table 4: Results of **RQ 3** for the experimental setups in Section 4. Comparison of mean squared errors for $\hat{\tau}(\mathbf{x})$ in the different scenarios. Estimates and simultaneous 95 % confidence intervals were obtained from a normal linear mixed model with log-link. Cells printed in bold font correspond to a superior reference of *Robinson $_{\hat{W}}$* , and cells printed in italics indicate an inferior reference.

did not lead to a major decrease in performance, although knowledge of the true functional form of the transformation function did not enter the Cox modeling process.

For Setup A, model-based forests without centering (*Naive*) were unable to cope with complex confounding, but solely centering of the treatment indicator ($Robinson_{\hat{W}}$) was valuable. Additionally adding $\hat{m}(\mathbf{x})$ as an offset (*Robinson*) did not further improve the results for the normal, binomial, and Weibull distributions, but an improvement was observed for the multinomial distribution.

For Setup B, the *Robinson* forests performed slightly better in disentangling the more complicated prognostic and predictive effects compared to *Naive* and $Robinson_{\hat{W}}$ model-based forests. An exception is the binomial model: without overlay, $Robinson_{\hat{W}}$ forests performed similarly to *Robinson* forests.

In Setup C, over all distributions, uncentered model-based forests (*Naive*) failed to overcome the strong confounding effect and therefore did not provide accurate estimates for the treatment effect. The performance was fundamentally improved by centering the treatment indicator ($Robinson_{\hat{W}}$) and was further improved by additionally adding $\hat{m}(\mathbf{x})$ as an offset (*Robinson*).

In Setup D – with unrelated treatment and control arms – all methods had a higher mean squared error than in the other setups, as jointly modeling the expected conditional outcomes for both arms has no benefit. Apart from the normal distributions, *Robinson* forests were inferior to the $Robinson_{\hat{W}}$ and *Naive* model-based forests.

The empirical evidence of our simulation study can be summarized as follows: If confounding was present, model-based forests performed better when centering W by $\hat{\pi}(\mathbf{x})$ ($Robinson_{\hat{W}}$) compared to not centering W (*Naive*). Adding $\hat{m}(\mathbf{x})$ as an offset (*Robinson*) further improved the performance – especially in cases with very strong confounding.

5. Effect of Riluzole on progression of ALS

Amyotrophic lateral sclerosis (ALS) is a progressive nervous system disease causing loss of muscle control. The status of the disease as well as the rate of progression is commonly evaluated by the ALS functional rating scale (ALSFRS) (Brooks, Sanjak, Ringel, England, and Brinkmann 1996; Cedarbaum, Stambler, Malta, Fuller, Hilt, Thurmond, and Nakanishi 1999). Here, physical abilities such as speaking, handwriting, and walking are assessed and rated on a scale from 0 (inability) to 4 (normal ability). In 1995, the FDA approved the first drug to manage and slow progression of ALS, named Riluzole. The largest database for study results on the effect of Riluzole offers the Pooled Resource Open-Access Clinical Trials (PROACT) database – initiated by the non-profit organization Prize4Life (<http://www.prize4life.org>). The data comes from different randomized and observational studies not disclosed in the data. Thus, the assumption of random treatment assignment is quite hard to justify in an analysis. Patient characteristics and treatment group sizes might vary greatly between the centers, which affect both the probability of receiving treatment as well as the outcome. To account for these potential confounding effects, we compared the treatment effects estimated by the naive model-based forests to the ones estimated with local centering by *Robinson*. As in Section 4, we use random forests to estimate the propensity scores to center W and gradient boosting machines (with adapted loss functions) to estimate the values of the linear predictors $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$ to center Y . Model-based forests, random forests,

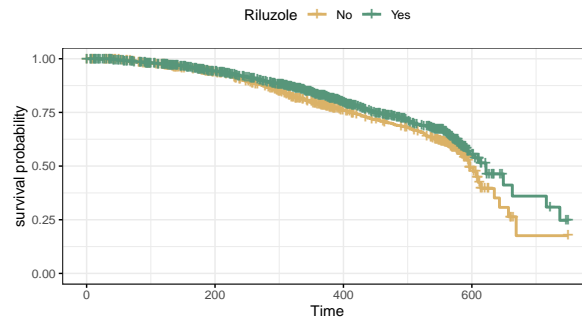


Figure 2: Kaplan-Meier curves of survival probability for both treatment arms.

and gradient boosting machines rely on the hyperparameter values stated in Section 7. As for Seibold *et al.* (2018) and Korepanova *et al.* (2020), 16 phase II and phase III randomized trials and one observational study from the PROACT database serve as a training dataset. We analyze the effect of Riluzole with respect to two outcome variables: survival time and the handwriting ability score approximately six months after treatment – an item of the ALSFRS. We omitted observations with missing outcome values. As splitting variables, Seibold *et al.* (2018) used demographic, medical history, and family history data, which were informative in the sense that not more than half of their values were missing.

5.1. Survival Time

The dataset for the survival time contains 3306 observations and 18 covariates. Of the 3306 observations, 2199 received Riluzole. Because very few patients had event times that exceed those of the others by a factor of two, we artificially censored five observations with (censoring or event) times of more than 750 days. The Kaplan-Meier estimates of survival probabilities for both treatment arms of the preprocessed dataset are shown in Figure 2. Overall, the estimated survival curves are very close to each other, and the treated group has only a slight survival advantage compared to the untreated group. As a base model, we use a Cox proportional hazards model. We compared treatment effects from two approaches: the naive uncentered model-based forests (*Naive*) and the model-based forest with Robinson’s orthogonalization (*Robinson*).

Personalized models

For the naive model-based forests, the underlying Cox proportional hazards base model for the survival outcome T was, on the hazard scale,

$$\lambda(t) = \lambda_0(t) \exp(\mu + \tau w)$$

Because $\lambda_0(t)$ contains an intercept term, μ is not identified (and was constraint to $\mu \equiv 0$). The treatment effect τ is the log-hazard ratio of the treated versus untreated patients and our aim is to replace a constant marginal effect τ with a heterogeneous (and thus conditional) log-hazard ratio $\tau(\mathbf{x})$ and, simultaneously, to estimate prognostic effects $\mu(\mathbf{x})$.

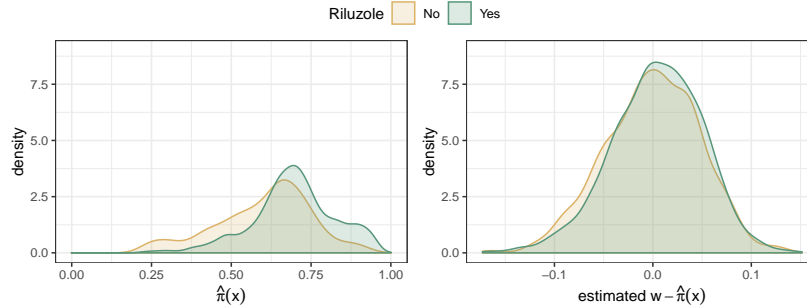


Figure 3: Distribution of estimated propensities $\hat{\pi}(\mathbf{x})$ (left) and estimated propensities of the centered treatment indicators (right, Robinson’s strategy) as estimated by regression forests for the two treatment groups.

For Robinson’s strategy, we first centered the treatment indicator W by estimating the propensity scores $\pi(\mathbf{x}) = \mathbb{P}(W | \mathbf{X} = \mathbf{x})$ using a regression forest. Figure 3 compares the distributions of estimated propensity scores (left) and of the estimated centered treatment $W - \hat{\pi}(\mathbf{x})$ (Robinson’s strategy, right), both obtained from regression forests. We can already see a decent overlap of propensity scores in the two treatment arms without centering, but the overlap increases if the strategy by Robinson was applied.

In addition to centering W , Robinson’s strategy requires the estimation of $m(\mathbf{x})$ to use as an offset (see Section 3). As in Section 4, we used gradient boosting machines (with the negative log partial likelihood of the Cox proportional hazards model as a loss) to estimate the natural parameters $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$ for the control and treatment group, respectively (Friedman 2001). The offset $m(\mathbf{x})$ for each observation is equal to the sum of natural parameter estimates weighted by $\hat{\pi}(\mathbf{x})$ (see equation (14)). The final base model for model-based forests using Robinson’s orthogonalization is

$$\lambda_R(t) = \lambda_0(t) \exp(\mu + \tau(w - \hat{\pi}(\mathbf{x})) + \hat{m}(\mathbf{x})).$$

Model-based forests

The corresponding base models serve as an input for the model-based forests to estimate personalized effects of Riluzole. Figure 4 compares the kernel density estimates of $\tau(\mathbf{x})$ for each forest version (*Naive* and *Robinson*). The naive approach reveals that on average the treatment reduced the hazard compared to no treatment, whereas the model-based forest with centering according to Robinson obtained weaker effects of Riluzole with more mass centered around 0.

A meta-analysis of previous studies by Andrews, Jackson, Heiman-Patterson, Bettica, Brooks, and Pioro (2020), also yielded a mixed picture: only eight of the 15 studies meeting their inclusion criteria showed a statistically significant increase of median survival time due to Riluzole.

Over all strategies, for both approaches there were some patients for which Riluzole was estimated to increase the hazard. The dependency plots in Figures S. 4 and S. 5 in the

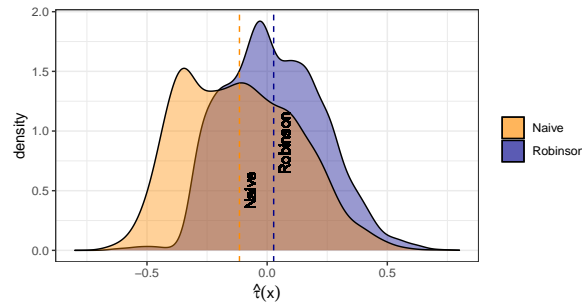


Figure 4: Kernel density estimates of the personalized treatment estimates for the naive model-based forest (*Naive*) and for the model-based forest with Robinson orthogonalization (*Robinson*).

Supplementary Material provide indications of the characteristics of the group of harmed individuals. For example, both the naive and centering approach agree that for patients with atrophy or fasciculation, Riluzole intake would increase the hazard. The estimated effects differed most between the uncentered forest (*Naive*) and the orthogonalized forest (*Robinson*) for the covariate sex (Figure S. 4 (c)), the covariate of whether patients swallow, and for the covariate specifying whether cases in the same generation exist (Figure S. 5 (f) and (i)).

For the variables time onset treatment, age, height and weakness the dependency plots (Figure S. 5 (a), (d), (e) and Figure S. 6 (g)) of the *Naive* forest agree with the ones of Seibold *et al.* (2018): for middle-aged people with a longer time between disease onset and start of treatment, lower height, and no weakness, the treatment appears to be more beneficial. By considering confounding effects due to orthogonalization (*Robinson* forests), these effects diminished. For Korepanova *et al.* (2020) the effect of Riluzole was also rather weak and showed low heterogeneity across covariates.

5.2. Handwriting Ability Score

The dataset for the handwriting ability score – an ordinal outcome with five categories – contains 2538 observations and 58 covariates. Besides the covariate age, all covariates had missing values (but less than 50 % of the values were missing per variable enforced by the preprocessing step stated at the beginning of this section). Of the 2538 observations, 1754 received Riluzole, and 784 did not. Figure 5 displays the frequency of the ability scores for both treatment groups. Most of the patients have an ability score of 3 or 4 (normal ability); only a few have ability scores less than 2. Note that the plot shows the conditional proportions given the treatment indicator. We chose a proportional odds logistic regression model as a base model for the model-based forests – once without further adaptations (*Naive*), and once parameterized with centered W and with an offset (*Robinson*).

In addition to the handwriting ability score after six months, the ability score values at treatment start are also available. In the following, we denote Y_6 as the handwriting score after six months and Y_0 as the handwriting score at the beginning of the treatment period. To account for the ability level at treatment start, Y_0 served as an additional splitting variable

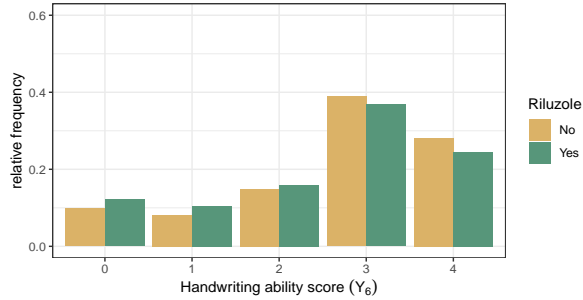


Figure 5: Relative frequency distribution plot of the handwriting ability score (Y_6) (left) and of changes of the handwriting ability score over six months ($Y_6 - Y_0$) (right) for both treatment arms. Frequencies were calculated relative to the treatment indicator.

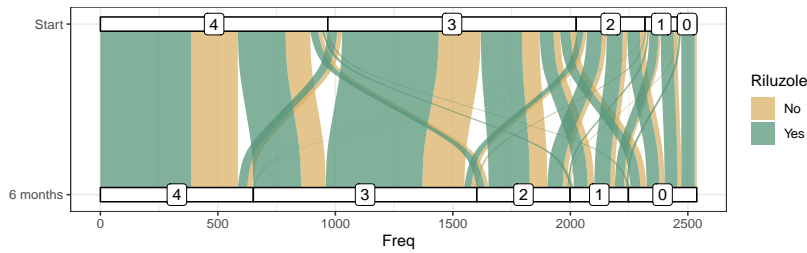


Figure 6: Alluvial plot of the progression of the handwriting ability score over six months for both treatment arms.

for both model-based forests (*Naive* and *Robinson*) and was included in \mathbf{X} . The alluvial plot in Figure 6 breaks down the change in each ability class over six months. Overall, for most patients, the handwriting ability remained constant over the six months or worsened slightly. Rarely, patients experienced a progression to both extremes (0 to 4, or 4 to 0). These results hold regardless of whether patients received Riluzole or not.

Personalized models

The proportional odds logistic regression model for the naive model-based forests is defined as (Agresti 2002; Venables and Ripley 2002)

$$\text{logit}(\mathbb{P}(Y_6 \leq k | \mathbf{X} = \mathbf{x}, W = w, Y_0 = y_0)) = \vartheta_k(\mathbf{x}, y_0) - \tau(\mathbf{x}, y_0)w$$

with $k \in \{0, \dots, 3\}$ as the ordinal ability score classes. The parameters ϑ_k are increasing thresholds, depending on covariates \mathbf{x} and the initial score y_0 . Due to the proportional odds assumption, the treatment effect $\tau(\mathbf{x}, y_0)$ is the same for all scores k . Negative $\tau(\mathbf{x}, y_0)$ indicate a negative effect of Riluzole, as treated patients are expected to have a higher odds of low writing ability scores compared to untreated patients.

As for the survival forest, we used regression forests to estimate propensity scores $\pi(\mathbf{x}, y_0)$ and

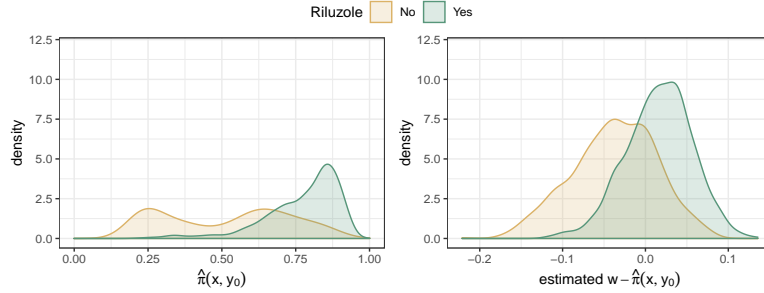


Figure 7: Estimates returned by the regression forest (rf) for orthogonalization of the treatment indicator: left for original W as an outcome in the rf such that it estimates propensity scores $\pi(\mathbf{x}, y_0)$; right for the centered treatment indicator $W - \hat{\pi}(\mathbf{x}, y_0)$ as an outcome in the rf.

a gradient boosting machine (with adapted loss functions for the proportional odds model) to estimate the natural parameters $\eta_0(\mathbf{x}, y_0)$ and $\eta_1(\mathbf{x}, y_0)$. The personalized model for the model-based forest with Robinson orthogonalization was specified as

$$\text{logit}(\mathbb{P}(Y_6 \leq k | \mathbf{X} = \mathbf{x}, W = w, Y_0 = y_0)) = \vartheta_k(\mathbf{x}, y_0) - [\hat{m}(\mathbf{x}, y_0) + \tau(\mathbf{x}, y_0)\{w - \hat{\pi}(\mathbf{x}, y_0)\}]$$

with $\hat{m}(\mathbf{x}, y_0)$ as defined in equation (14).

Figure 7 compares the estimated treatment indicators with W as the outcome in the random forest without centering (left), with $(W - \hat{\pi}(\mathbf{x}, y_0))$ as the outcome in the random forest (right). Before centering, there is a lack of overlap of the propensity scores; the distribution of $\hat{\pi}$ for the control group is bimodal, and the distribution for the treatment group is heavily left-skewed. After centering, the distributions of the estimated $W - \hat{\pi}(\mathbf{x}, y_0)$ for the treatment groups move closer together and have a similar unimodal shape. However, there is still a lack of overlap of the groups, which indicates that important covariates to explain the remaining heterogeneity in the two treatment groups seem to be missing.

Model-based forests

The proportional odds logistic regression models served as a base model for the (*Naive* and *Robinson*) model-based forests to derive personalized treatment effects. Figure 8 displays the kernel density estimates of $\tau(\mathbf{x}, y_0)$ for each forest version (*Naive* and *Robinson*). Both random forests estimate on average a negative effect of Riluzole. Naive model-based forest estimated on average a log-odds of $\bar{\tau} = -0.08$, which indicates that treated patients have a 0.08 points higher log-odds for low writing scores than untreated patients. The distribution of $\hat{\tau}(\mathbf{x}, y_0)$ for the model-based forest relying on the Robinson orthogonalization is slightly shifted to the left ($\bar{\tau} = -0.10$). For a larger subgroup of patients, the naive approach estimates a negative effect of Riluzole ($-1 \leq \tau(\mathbf{x}, y_0) \leq -0.5$), meaning that patients receiving treatment with Riluzole have higher odds of low writing scores than untreated patients. According to the dependency plots (Figures S. 6 to S. 11 in the Supplementary Material), this subgroup could be identified as having the low initial ability scores (left side of Figure S. 6 (a)). For all other splitting variables, the distributions of estimated treatment effects are very similar.

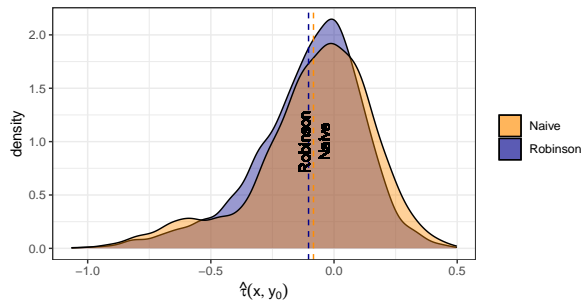


Figure 8: Kernel density estimates of the personalized treatment estimates for the naive model-based forest (*Naive*) vs. the forest with Robinson orthogonalized (*Robinson*).

6. Discussion and outlook

HTE estimation is a challenging problem, especially for observational studies and even more when the outcome cannot be modeled by a linear model. In this work, we investigated several versions of model-based forests for the estimation of potentially complex HTEs $\tau(\mathbf{x})$ based on observational data with various outcome types based on the orthogonalization strategy by Robinson (Robinson 1988). These investigations suggest the following workflow for model-based forests: (1) estimate propensities $\pi(\mathbf{x})$ using some machine learning procedure (binary random forests are a good default), (2) center the treatment indicator $w - \hat{\pi}(\mathbf{x})$ for each observation, (3) setup an appropriate model for the outcome conditioning on the centered treatment and – if possible – add an offset for centering Y , (4) use model-based forests to estimate predictive and prognostic effects $\tau(\mathbf{x})$ and $\mu(\mathbf{x})$ simultaneously. Notably, $\tau(\mathbf{x})$ is the CATE only in specific models, especially a linear or log-linear model. We demonstrate these steps by estimating the individual effects of Riluzole for ALS patients using survival times and ordinal ability scores as outcomes.

Our work still leaves open questions for example how model-based forests perform for survival data for which the censoring procedure is not randomized but depends on \mathbf{X} , or how (k -fold) cross-fitting influences the performance, where only one part of the data is used to estimate nuisance parameters and the other part to estimate $\tau(\mathbf{x})$ (Chernozhukov *et al.* 2018). We leave investigations to these questions to future research.

Last but not least, we want to emphasize that all approaches for estimating HTEs – including those presented in this work – rely on strong and typically untestable assumptions. For example, for models beyond mean regression, $\hat{\tau}(\mathbf{x})$ cannot be expected to be robust against missing covariates or other violations of model assumptions due to non-collapsibility. Consequently, results from these approaches in practical applications should be evaluated with the utmost caution, reservation, and humility.

7. Computational details

For all computations, we used R version 4.1.1 (R Core Team 2022), with the following add-on packages: `model4you` (Seibold, Zeileis, and Hothorn 2021), `trtf` (Hothorn 2021), `partykit`

(Hothorn and Zeileis 2021a), **grf** (Tibshirani *et al.* 2021), **mboost** (Hothorn *et al.* 2021b), and **gbm** (Greenwell *et al.* 2020).

Model-based forests were always grown with $M = 500$ trees (`model4you::pmforest` default) with a minimum node size of `node = 14`, number of chosen variables per split `mtry = P`, and subsampling. These settings were also used by Dandl *et al.* (2022). Transformation forests implemented in the **trtf** package fitted the Weibull transformation forests of Section 4 (Hothorn 2021; Hothorn and Zeileis 2021b).

Propensity scores $\pi(\mathbf{x})$ were estimated with **grf** (honest) regression forests with 125 trees, a minimum node size of 5, and subsampling. Natural parameters $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$ and probability of not being censored were estimated with gradient boosting machines implemented in the **mboost** or **gbm** packages. The used maximum tree depth was 2 (default of `mboost::blackboost`), and a loss function that differed depending on the outcome type was also employed (Hothorn *et al.* 2021b; Greenwell *et al.* 2020).

Ratios and confidence intervals presented in Table 2 were calculated using generalized linear mixed models of the **glmmTMB** package (Magnusson, Skaug, Nielsen, Berg, Kristensen, Maechler, van Benthem, Bolker, and Brooks 2021). Post-hoc inference relied on the **multcomp** package (Hothorn, Bretz, and Westfall 2021a).

All study settings are available in a dedicated R package called **htesim** (Dandl and Hothorn 2021). It is published on Github: <https://github.com/dandls/htesim>.

Funding

TH received funding from the Swiss National Science Foundation, Grant No. 200021_184603, Horizon 2020 Research and Innovation Programme of the European Union under grant agreement number 681094, and is supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 15.0137.

References

- Aalen OO, Cook RJ, Røysland K (2015). “Does Cox Analysis of a Randomized Survival Study Yield a Causal Treatment Effect?” *Lifetime Data Analysis*, **21**(4), 579–593. doi: [10.1007/s10985-015-9335-y](https://doi.org/10.1007/s10985-015-9335-y).
- Agresti A (2002). *Categorical Data Analysis*. John Wiley & Sons, Inc. doi: [10.1002/0471249688](https://doi.org/10.1002/0471249688).
- Andrews JA, Jackson CE, Heiman-Patterson TD, Bettica P, Brooks BR, Piro EP (2020). “Real-world Evidence of Riluzole Effectiveness in Treating Amyotrophic Lateral Sclerosis.” *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, **21**(7–8), 509–518. doi: [10.1080/21678421.2020.1771734](https://doi.org/10.1080/21678421.2020.1771734).
- Athey S, Tibshirani J, Wager S (2019). “Generalized Random Forests.” *The Annals of Statistics*, **47**(2), 1148–1178. doi: [10.1214/18-aos1709](https://doi.org/10.1214/18-aos1709).
- Brooks B, Sanjak M, Ringel S, England J, Brinkmann J (1996). “The Amyotrophic Lateral Sclerosis Functional Rating Scale: Assessment of Activities of Daily Living in Patients

- With Amyotrophic Lateral Sclerosis.” *Archives of Neurology*, **53**(2), 141–147. doi:10.1001/archneur.1996.00550020045014.
- Buri M, Hothorn T (2020). “Model-based Random Forests for Ordinal Regression.” *International Journal of Biostatistics*, **16**(2), 20190063. doi:10.1515/ijb-2019-0063.
- Cedarbaum JM, Stambler N, Malta E, Fuller C, Hilt D, Thurmond B, Nakanishi A (1999). “The ALSFRS-R: A Revised ALS Functional Rating Scale that Incorporates Assessments of Respiratory Function.” *Journal of the Neurological Sciences*, **169**(1), 13–21. doi:https://doi.org/10.1016/S0022-510X(99)00210-5.
- Chapfuwa P, Assaad S, Zeng S, Pencina MJ, Carin L, Henao R (2021). “Enabling Counterfactual Survival Analysis with Balanced Representations.” CHIL ’21, pp. 133–145. Association for Computing Machinery, New York, NY, USA. ISBN 9781450383592. doi:10.1145/3450439.3451875.
- 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. doi:10.1111/ectj.12097.
- Cui Y, Kosorok MR, Sverdrup E, Wager S, Ruoqing (2022). “Estimating Heterogeneous Treatment Effects with Right-Censored Data via Causal Survival Forests.” *Technical report*, arXiv 2001.09887 v3. URL <https://arxiv.org/abs/2001.09887>.
- Curth A, Lee C, van der Schaar M (2021). “SurvITE: Learning Heterogeneous Treatment Effects from Time-to-Event Data.” *Technical report*, arXiv 2110.14001. URL <https://arxiv.org/abs/2110.14001>.
- Dandl S, Hothorn T (2021). *htesim: Conducting Extensive Simulation Studies of Heterogeneous Treatment Effect Estimation*. R package version 0.0.0.9000, URL <https://github.com/susanne-207/htesim>.
- Dandl S, Hothorn T, Seibold H, Sverdrup E, Wager S, Zeileis A (2022). “What Makes Forest-Based Heterogeneous Treatment Effect Estimators Work?” *Technical report*, arXiv 2206.10323. URL <https://arxiv.org/abs/2206.10323>.
- Daniel R, Zhang J, Farewell D (2021). “Making Apples from Oranges: Comparing Non-collapsible Effect Estimators and Their Standard Errors after Adjustment for Different Covariate Sets.” *Biometrical Journal*, **63**(3), 528–557. doi:https://doi.org/10.1002/bimj.201900297.
- Fokkema M, Smits N, Zeileis A, Hothorn T, Kelderman H (2018). “Detecting Treatment-Subgroup Interactions in Clustered Data with Generalized Linear Mixed-Effects Model Trees.” *Behavior Research Methods*, **50**(6), 2016–2034. doi:10.3758/s13428-017-0971-x.
- Foster JC, Taylor J, Ruberg S (2011). “Subgroup Identification from Randomized Clinical Trial Data.” *Statistics in Medicine*, **30**(24), 2867–2880. doi:10.1002/sim.4322.
- Friedman JH (2001). “Greedy Function Approximation: A Gradient Boosting Machine.” *The Annals of Statistics*, **29**(5), 1189–1232. doi:10.1214/aos/1013203451.

- Gao Z, Hastie T (2022). “Estimating Heterogeneous Treatment Effects for General Responses.” *Technical report*, arXiv 2103.04277 v4. URL <https://arxiv.org/abs/2103.04277>.
- Greenland S (1996). “Absence of Confounding Does Not Correspond to Collapsibility of the Rate Ratio or Rate Difference.” *Epidemiology*, **7**, 498–501.
- Greenland S, Pearl J, Robins JM (1999). “Confounding and Collapsibility in Causal Inference.” *Statistical Science*, **14**(1), 29–46. doi:10.1214/ss/1009211805.
- Greenwell B, Boehmke B, Cunningham J, Developers G (2020). **gbm**: *Generalized Boosted Regression Models*. R package version 2.1.8, URL <https://CRAN.R-project.org/package=gbm>.
- Henderson NC, Louis TA, Rosner GL, Varadhan R (2018). “Individualized Treatment Effects with Censored Data via Fully Nonparametric Bayesian Accelerated Failure Time Models.” *Biostatistics*, **21**(1), 50–68. ISSN 1465-4644. doi:10.1093/biostatistics/kxy028.
- Hill JL (2011). “Bayesian Nonparametric Modeling for Causal Inference.” *Journal of Computational and Graphical Statistics*, **20**(1), 217–240. doi:10.1198/jcgs.2010.08162.
- Hothorn T (2021). **trtf**: *Transformation Trees and Forests*. R package version 0.3-8, URL <https://CRAN.R-project.org/package=trtf>.
- Hothorn T, Bretz F, Westfall P (2021a). **multcomp**: *Simultaneous Inference in General Parametric Models*. R package version 1.4-17, URL <https://CRAN.R-project.org/package=multcomp>.
- Hothorn T, Bühlmann P, Kneib T, Schmid M, Hofner B (2021b). **mboost**: *Model-Based Boosting*. R package version 2.9-5, URL <https://CRAN.R-project.org/package=mboost>.
- Hothorn T, Hornik K, Zeileis A (2006). “Unbiased Recursive Partitioning: A Conditional Inference Framework.” *Journal of Computational and Graphical Statistics*, **15**(3), 651–674. doi:10.1198/106186006x133933.
- Hothorn T, Lausen B, Benner A, Radespiel-Tröger M (2004). “Bagging Survival Trees.” *Statistics in Medicine*, **23**(1), 77–91. doi:10.1002/sim.1593.
- Hothorn T, Möst L, Bühlmann P (2018). “Most Likely Transformations.” *Scandinavian Journal of Statistics*, **45**(1), 110–134. doi:10.1111/sjos.12291.
- Hothorn T, Zeileis A (2021a). **partykit**: *A Toolkit for Recursive Partytioning*. R package version 1.2-15, URL <https://R-Forge.R-project.org/projects/partykit/>.
- Hothorn T, Zeileis A (2021b). “Predictive Distribution Modelling Using Transformation Forests.” *Journal of Computational and Graphical Statistics*, **14**, 144–148. doi:10.1080/10618600.2021.1872581.
- Hu L, Gu C, Lopez MJ, Ji J, Wisnivesky J (2020). “Estimation of Causal Effects of Multiple Treatments in Observational Studies with a Binary Outcome.” *Statistical Methods in Medical Research*, **29**, 3218–3234. doi:10.1177/0962280220921909.

- Hu L, Ji J, Li F (2021). “Estimating Heterogeneous Survival Treatment Effect in Observational Data using Machine Learning.” *Statistics in Medicine*, **40**, 4691–4713. doi:10.1002/sim.9090.
- Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS (2008). “Random Survival Forests.” *The Annals of Applied Statistics*, **2**(3), 841–860. doi:10.1214/08-aos169.
- Katzman J, Shaham U, Cloninger A, Bates J, Jiang T, Kluger Y (2018). “DeepSurv: Personalized Treatment Recommender System using a Cox Proportional Hazards Deep Neural Network.” *BMC Medical Research Methodology*, **18**, 24. doi:10.1186/s12874-018-0482-1.
- Korepanova N, Seibold H, Steffen V, Hothorn T (2020). “Survival Forests under Test: Impact of the Proportional Hazards Assumption on Prognostic and Predictive Forests for ALS Survival.” *Statistical Methods in Medical Research*, **29**(5), 1403–1419. doi:10.1177/0962280219862586.
- Künzel SR, Sekhon JS, Bickel PJ, Yu B (2019). “Metalearners for Estimating Heterogeneous Treatment Effects Using Machine Learning.” *Proceedings of the National Academy of Sciences of the United States of America*, **116**(10), 4156–4165. doi:10.1073/pnas.1804597116.
- Li L, Levine RA, Fan J (2022). “Causal Effect Random Forest of Interaction Trees for Learning Individualized Treatment Regimes with Multiple Treatments in Observational Studies.” *Stat*, **11**(1), e457. doi:10.1002/sta4.457.
- Lin Y, Jeon Y (2006). “Random Forests and Adaptive Nearest Neighbors.” *Journal of the American Statistical Association*, **101**(474), 578–590. doi:10.1198/016214505000001230.
- Lu M, Sadiq S, Feaster DJ, Ishwaran H (2018). “Estimating Individual Treatment Effect in Observational Data Using Random Forest Methods.” *Journal of Computational and Graphical Statistics*, **27**(1), 209–219. doi:10.1080/10618600.2017.1356325.
- Magnusson A, Skaug H, Nielsen A, Berg C, Kristensen K, Maechler M, van Bentham K, Bolker B, Brooks M (2021). *glmmTMB: Generalized Linear Mixed Models using Template Model Builder*. R package version 1.1.2.3, URL <https://CRAN.R-project.org/package=glmmTMB>.
- Meinshausen N (2006). “Quantile Regression Forests.” *Journal of Machine Learning Research*, **7**, 983–999. doi:10.1007/s10994-014-5452-1.
- Nie X, Wager S (2021). “Quasi-Oracle Estimation of Heterogeneous Treatment Effects.” *Biometrika*, **108**, 299–319. doi:10.1093/biomet/asaa076.
- Powers S, Qian J, Jung K, Schuler A, Shah NH, Hastie T, Tibshirani R (2018). “Some Methods for Heterogeneous Treatment Effect Estimation in High Dimensions.” *Statistics in Medicine*, **37**(11), 1767–1787. doi:10.1002/sim.7623.
- R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Robinson PM (1988). “Root-N-Consistent Semiparametric Regression.” *Econometrica*, **56**(4), 931–954. doi:10.2307/1912705.

- Rosenbaum PR, Rubin DB (1983). “The Central Role of the Propensity Score in Observational Studies for Causal Effects.” *Biometrika*, **70**(1), 41–55.
- Rubin DB (1974). “Estimating causal effects of treatments in randomized and nonrandomized studies.” *Journal of Educational Psychology*, **66**, 688–701.
- Rubin DB (2005). “Causal Inference Using Potential Outcomes: Design, Modeling, Decisions.” *Journal of the American Statistical Association*, **100**(469), 322–331.
- Seibold H, Zeileis A, Hothorn T (2016). “Model-Based Recursive Partitioning for Subgroup Analyses.” *International Journal of Biostatistics*, **12**(1), 45–63. doi:10.1515/ijb-2015-0032.
- Seibold H, Zeileis A, Hothorn T (2018). “Individual Treatment Effect Prediction for Amyotrophic Lateral Sclerosis Patients.” *Statistical Methods in Medical Research*, **27**(10), 3104–3125. doi:10.1177/0962280217693034.
- Seibold H, Zeileis A, Hothorn T (2019). “**model4you**: An R Package for Personalised Treatment Effect Estimation.” *Journal of Open Research Software*, **7**(17), 1–6. doi:10.5334/jors.219.
- Seibold H, Zeileis A, Hothorn T (2021). **model4you: Stratified and Personalised Models Based on Model-Based Trees and Forests**. R package version 0.9-7, URL <https://R-Forge.R-project.org/projects/partykit/>.
- Shalit U, Johansson FD, Sontag D (2017). “Estimating Individual Treatment Effect: Generalization Bounds and Algorithms.” *Technical report*, arXiv 1606.03976. URL <https://arxiv.org/abs/1606.03976>.
- Su X, Peña AT, Liu L, Levine RA (2018). “Random Forests of Interaction Trees for Estimating Individualized Treatment Effects in Randomized Trials.” *Statistics in Medicine*, **37**(17), 2547–2560. doi:10.1002/sim.7660.
- Tabib S, Larocque D (2020). “Non-parametric Individual Treatment Effect Estimation for Survival Data with Random Forests.” *Bioinformatics*, **36**, 629–636. doi:10.1093/bioinformatics/btz602.
- Tibshirani J, Athey S, Sverdrup E, Wager S (2021). **grf: Generalized Random Forests**. R package version 2.0.2, URL <https://github.com/grf-labs/grf>.
- Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Springer New York. doi:10.1007/978-0-387-21706-2.
- 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. doi:10.1080/01621459.2017.1319839.
- Zeileis A, Hothorn T, Hornik K (2008). “Model-Based Recursive Partitioning.” *Journal of Computational and Graphical Statistics*, **17**(2), 492–514. doi:10.1198/106186008x319331.

A. Noncollapsibility

As mentioned in Section 2.4, for members of the exponential family without an identity or linear link function the marginal and conditional treatment effects are not collapsible. This means that the mean of the conditional treatment effects given a covariate are not equal to the marginal treatment effect estimate over the same covariate (Greenland *et al.* 1999). This happens if the covariate conditioned on is associated with the outcome of interest. Caution is necessary on multiple stages of the estimation process of $\tau(\mathbf{x})$ as soon as we condition on other covariates, for example, because these covariates are assumed to be sufficient to control for confounding (Daniel *et al.* 2021).

In case of Robinson’s orthogonalization, misspecification of $m(\mathbf{x})$ translates into biased estimators for $\tau(\mathbf{x})$, even under randomized treatments. This also applies if one ignores the estimation of $\mu(\mathbf{x})$ at all and only concentrates on $\tau(\mathbf{x})$. This is not the case for the linear model (identity link function) since misspecifications are absorbed in the additive error term and do not influence the estimation of $\tau(\mathbf{x})$ (Gao and Hastie 2022).

A.1. Review Gao and Hastie (2022)

Gao and Hastie (2022) extended the orthogonalization strategy of Robinson (1988) to improve robustness to both confounding and noncollapsibility. The authors propose

$$a(\mathbf{x}) = \frac{\pi(\mathbf{x}) \frac{\partial \gamma(\eta_1(\mathbf{x}))}{\partial \eta}}{\pi(\mathbf{x}) \frac{\partial \gamma(\eta_1(\mathbf{x}))}{\partial \eta} + (1 - \pi(\mathbf{x})) \frac{\partial \gamma(\eta_0(\mathbf{x}))}{\partial \eta}} \quad (16)$$

and

$$\nu(\mathbf{x}) = a(\mathbf{x})n_1(\mathbf{x}) + (1 - a(\mathbf{x}))n_0(\mathbf{x})$$

instead of $\pi(\mathbf{x})$ (equation (9)) and $m(\mathbf{x})$ (equation (14)), respectively, where $\gamma(\eta)$ denotes the inverse of the canonical link function. Its derivative is equal to the variance function of the exponential family. Therefore, $a(\mathbf{x})$ is larger if an observation is likely to be treated (which also holds for Robinson’s orthogonalization) or if the response variance is higher under treatment compared to no treatment. As a consequence of the latter, the influence of spuriously influential natural parameter values is reduced for more robustness to misspecifications (Gao and Hastie 2022).

For Gaussian responses, $a(\mathbf{x}) = \pi(\mathbf{x})$ and $\nu(\mathbf{x}) = m(\mathbf{x})$ holds, while for other distributions the terms differ. For example, for Bernoulli distributed Y , the closed form $a(\mathbf{x})$ is

$$a(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + (1 - \pi(\mathbf{x})) \frac{p_0(\mathbf{x})(1-p_0(\mathbf{x}))}{p_1(\mathbf{x})(1-p_1(\mathbf{x}))}} \quad (17)$$

where $p_w(\mathbf{x}) = \mathbb{P}(Y = 1 | \mathbf{X} = \mathbf{x}, W = w)$.

The noncollapsibility issue is not only present for distributions of the exponential family. Also the Cox model suffers from noncollapsibility (Greenland 1996; Aalen *et al.* 2015). This is in contrast to accelerated failure time models (such as the Weibull proportional hazards model), which can be rewritten as location-scale models and therefore are indeed collapsible (Aalen *et al.* 2015). For the Cox model, Gao and Hastie remark that with knowledge of the baseline hazard function and without censoring, the cumulative hazard function follows an exponential

distribution. For the exponential distribution, $a(\mathbf{x})$ and $\nu(\mathbf{x})$ are equal to $\pi(\mathbf{x})$ and $m(\mathbf{x})$ (Gao and Hastie 2022).

In case of random censoring, the probability of not being censored under both treatment arms needs to be considered for the estimation of $a(\mathbf{x})$ and $\nu(\mathbf{x})$

$$a(\mathbf{x}) = \frac{\pi(\mathbf{x})\mathbb{P}(C \geq Y | \mathbf{X} = \mathbf{x}, W = 1)}{\pi(\mathbf{x})\mathbb{P}(C \geq Y | \mathbf{X} = \mathbf{x}, W = 1) + (1 - \pi(\mathbf{x}))\mathbb{P}(C \geq Y | \mathbf{X} = \mathbf{x}, W = 0)} \quad (18)$$

$$\nu(\mathbf{x}) = a(\mathbf{x})\eta_1(\mathbf{x}) + (1 - a(\mathbf{x}))\eta_0(\mathbf{x}). \quad (19)$$

The nuisance parameter $a(\mathbf{x})$ is larger if an observation is likely to be treated or likely to be not censored. Consequently, the influence of likely to be not censored observations for the estimation of $\tau(\mathbf{x})$ is increased. Above's $a(\mathbf{x})$ and $\nu(\mathbf{x})$ guarantee protection to misspecified nuisance parameter if the baseline hazard is known. If it is unknown and the partial likelihood is used – this is not guaranteed. Despite this lack of guarantee, Gao and Hastie, 2022, obtained promising results in their simulation study (Gao and Hastie 2022).

A.2. Strategies against confounding and noncollapsibility

An interesting question is if replacing $\hat{\pi}(\mathbf{x})$ and $\hat{m}(\mathbf{x})$ by $\hat{a}(\mathbf{x})$ and $\hat{\nu}(\mathbf{x})$, respectively, also helps to additionally tackle noncollapsibility when applying model-based forests. We can update the linear predictor for model-based forests in case of generalized linear models to

$$g(\mathbb{E}(Y | \mathbf{X} = \mathbf{x}, W = w)) = \hat{\nu}(\mathbf{x}) + \tilde{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{a}(\mathbf{x})).$$

Gao and Hastie additionally derived estimators for $a(\mathbf{x})$ and $\nu(\mathbf{x})$ for the Cox model which – compared to the Weibull model – is not collapsible. For the Cox model, the natural parameter of equation (8) could be updated to

$$\eta_w(\mathbf{x}) = \hat{\nu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{a}(\mathbf{x}))$$

with $a(\mathbf{x})$ and $\nu(\mathbf{x})$ as defined in equations (18) and (19).

We call this version of model-based forests in the following *Gao* approach. Before we apply model-based forests, we need to estimate $\pi(\mathbf{x})$, $\eta_0(\mathbf{x})$, $\eta_1(\mathbf{x})$ as well as $\frac{\partial \nu(\eta_1(\mathbf{x}))}{\partial \eta}$ for exponential families and $\mathbb{P}(C \geq Y | \mathbf{X} = \mathbf{x}, W = w)$ for Cox models. As in Section 3.3, we state some research questions that are empirically inspected in the upcoming section.

RQ 4: How do model-based forests centered according to Gao and Hastie (*Gao*) perform compared to model-based forest with Robinson strategy (*Robinson*) for the simulation settings of Section 4?

Similar to RQ 2, we could solely center W by $a(\mathbf{x})$ without including an offset. We call this approach $Gao_{\hat{W}}$ in the following.

RQ 5: How do model-based forest with solely centered W by $\hat{a}(\mathbf{x})$ ($Gao_{\hat{W}}$) perform compared to model-based forests with solely centered W by $\hat{\pi}(\mathbf{x})$ $Robinson_{\hat{W}}$ for the simulation study settings of Section 4?

Method	Linear Predictor	Definitions
<i>Naive</i>	$\mu(\mathbf{x}) + \tau(\mathbf{x}) w$	
<i>Robinson</i> _{\hat{W}}	$\mu(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x}))$	$\pi(\mathbf{x}) = \mathbb{P}(W = 1 \mathbf{X} = \mathbf{x})$
<i>Robinson</i>	$\tilde{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{\pi}(\mathbf{x})) + \hat{m}(\mathbf{x})$	$m(\mathbf{x}) = \pi(\mathbf{x})\eta_1(\mathbf{x}) - (1 - \pi(\mathbf{x}))\eta_0(\mathbf{x})$
<i>Gao</i> _{\hat{W}}	$\mu(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{a}(\mathbf{x}))$	$a(\mathbf{x}) = \frac{\pi(\mathbf{x}) \frac{\partial \gamma(\eta_1(\mathbf{x}))}{\partial \eta}}{\pi(\mathbf{x}) \frac{\partial \gamma(\eta_1(\mathbf{x}))}{\partial \eta} + (1 - \pi(\mathbf{x})) \frac{\partial \gamma(\eta_0(\mathbf{x}))}{\partial \eta}}$
<i>Gao</i>	$\tilde{\mu}(\mathbf{x}) + \tau(\mathbf{x})(w - \hat{a}(\mathbf{x})) + \hat{\nu}(\mathbf{x})$	$\nu(\mathbf{x}) = a(\mathbf{x})n_1(\mathbf{x}) + (1 - a(\mathbf{x}))n_0(\mathbf{x})$

Note: for the Cox model $a(\mathbf{x}) = \frac{\pi(\mathbf{x})\mathbb{P}(C \geq Y | \mathbf{X}=\mathbf{x}, W=1)}{\pi(\mathbf{x})\mathbb{P}(C \geq Y | \mathbf{X}=\mathbf{x}, W=1) + (1 - \pi(\mathbf{x}))\mathbb{P}(C \geq Y | \mathbf{X}=\mathbf{x}, W=0)}$ is used.

Table S. 1: Updated overview of proposed model-based forest versions (Table 1) for observational data.

A.3. Data-generating process

To investigate the research questions of Section A.2., we compared the performance of model-based forests with Gao’s strategy proposed in Section A.2 (*Gao* and *Gao* _{\hat{W}}) to model-based forests with Robinson’s strategy (*Robinson* and *Robinson* _{\hat{W}}) for settings A, B, C, D described in Section 4. Because we expect that the strategy of Gao is especially valuable for settings with misspecified prognostic effect, e.g. because prognostic covariates are missing, we additionally created Setup A’ from Setup A by removing covariate \mathbf{X}_3 from the training data. Therefore, the DGP of Setup A and Setup A’ are identical, the only difference being that the training data did not contain \mathbf{X}_3 although \mathbf{X}_3 affects the prognostic effect.

Because the normal linear model and Weibull model are collapsible and Gao’s strategy is equal to Robinson’s strategy (Sections 2.4 and A.1), we applied our proposed approaches based on [Gao and Hastie \(2022\)](#) only to the binomial model and the Cox model. Transformation models such as the proportional odds model for multinomial data were not covered by the authors.

We used the same model-based forest parameter setup and evaluation scheme as in Section 4.

A.4. Results

For Setup A, solely centering W by $\hat{a}(\mathbf{x})$ (*Gao* _{\hat{W}}) achieved better results than additionally adding the offset $\hat{\nu}(\mathbf{x})$ (*Gao*). Model-based forests with Robinson’s strategy (*Robinson*, *Robinson* _{\hat{W}}) overall performed better than model-based forests with Gao’s strategy (*Gao*, *Gao* _{\hat{W}}). Suppressing X_3 in the training dataset (Setup A’), did not deteriorate the performance of all methods such that the ranking of methods was retained.

For Setup B, model-based forests centered by *Gao* and *Robinson* model-based forests performed akin for binary outcomes. Also *Robinson* _{\hat{W}} and *Gao* _{\hat{W}} model-based forests achieved similar performance.

In Setup C, Gao’s strategy for the Cox and logistic regression model overall fare worse than Robinson’s strategy. In Setup D, *Gao* _{\hat{W}} forests performed as good as *Robinson* _{\hat{W}} forests for the Cox and logistic regression models. Notably, for the Cox model, *Gao* forests outperformed *Robinson* forests.

Overall, the orthogonalization strategy of Gao for the exponential family – that aims at addressing the noncollapsibility issue – did not perform as well as expected. Our expectation

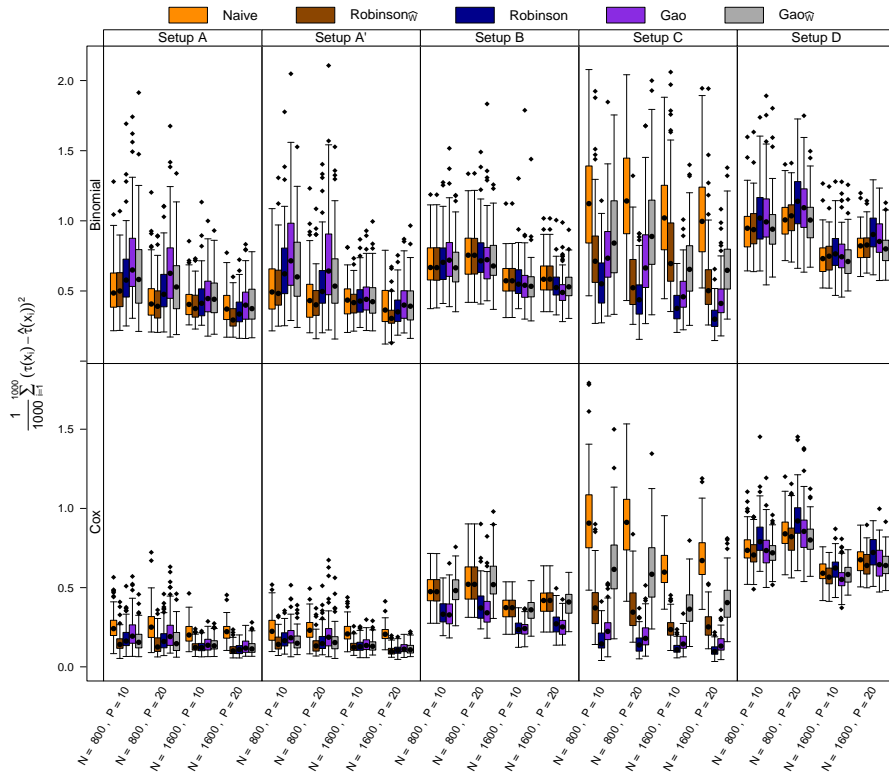


Figure S. 1: Model-based forest results for the empirical study (Section 4), Cox means a Cox model applied to the Weibull data. For the Cox model, treatment effects $\tau(\mathbf{x})$ are estimated as conditional log hazard ratios. Direct comparison of model-based forests without centering (Naive), model-based forests with local centering according to [Robinson \(1988\)](#) or [Gao and Hastie \(2022\)](#) of Y and W (originally proposed) (*Robinson*, *Gao*) or only of W (*Robinson* $_{\hat{W}}$, *Gao* $_{\hat{W}}$).

DGP	N	P	Mean squared error ratio for RQ 4: Gao vs. Robinson	
			Binomial	Cox
Setup A	800	10	1.258 (1.152, 1.373)	1.203 (1.077, 1.344)
		20	1.307 (1.180, 1.449)	1.307 (1.170, 1.461)
	1600	10	1.067 (0.933, 1.220)	1.121 (0.947, 1.326)
Setup A'	800	10	1.183 (1.009, 1.388)	1.155 (0.955, 1.398)
		20	1.201 (1.105, 1.304)	1.140 (1.011, 1.285)
	1600	10	1.354 (1.233, 1.488)	1.272 (1.127, 1.435)
Setup B	800	10	1.047 (0.915, 1.200)	1.055 (0.895, 1.243)
		20	1.184 (1.014, 1.382)	1.114 (0.911, 1.362)
	1600	10	1.042 (0.958, 1.134)	0.984 (0.920, 1.052)
Setup C	800	10	0.987 (0.909, 1.073)	<i>0.906 (0.853, 0.963)</i>
		20	0.987 (0.885, 1.100)	0.977 (0.889, 1.074)
	1600	10	0.926 (0.824, 1.042)	0.922 (0.845, 1.006)
Setup D	800	10	1.388 (1.263, 1.524)	1.417 (1.261, 1.592)
		20	1.616 (1.448, 1.804)	1.401 (1.228, 1.598)
	1600	10	1.276 (1.104, 1.476)	1.360 (1.146, 1.615)
Setup D	800	10	1.485 (1.255, 1.758)	1.400 (1.163, 1.686)
		20	0.996 (0.939, 1.057)	<i>0.916 (0.889, 0.943)</i>
	1600	10	0.965 (0.913, 1.020)	<i>0.925 (0.902, 0.949)</i>
		20	0.964 (0.890, 1.044)	<i>0.910 (0.875, 0.946)</i>
		20	0.948 (0.884, 1.015)	<i>0.907 (0.877, 0.938)</i>

Table S. 2: Results of **RQ 4** for the experimental setups in Section 4. Comparison of mean squared errors for $\hat{\tau}(\mathbf{x})$ in the different scenarios. Estimates and simultaneous 95 % confidence intervals were obtained from a normal linear mixed model with log-link. Cells printed in bold font correspond to a superior reference of *Robinson* forests, cells printed in italics indicate an inferior reference.

DGP	N	P	Mean squared error ratio for RQ 5: Gao _{\hat{W}} vs. Robinson _{\hat{W}}	
			Binomial	Cox
Setup A	800	10	1.299 (1.168, 1.445)	1.127 (0.986, 1.288)
		20	1.425 (1.255, 1.618)	1.190 (1.038, 1.366)
	1600	10	1.162 (1.009, 1.339)	1.110 (0.940, 1.310)
Setup A'	800	10	1.339 (1.128, 1.589)	1.144 (0.944, 1.386)
		20	1.261 (1.139, 1.397)	1.096 (0.952, 1.263)
	1600	10	1.427 (1.264, 1.610)	1.195 (1.033, 1.382)
Setup B	800	10	1.096 (0.950, 1.264)	1.060 (0.896, 1.255)
		20	1.305 (1.101, 1.548)	1.114 (0.906, 1.370)
	1600	10	0.988 (0.904, 1.079)	1.005 (0.959, 1.053)
Setup C	800	10	0.959 (0.883, 1.042)	1.037 (0.995, 1.081)
		20	0.947 (0.849, 1.056)	0.968 (0.910, 1.031)
	1600	10	0.905 (0.811, 1.009)	0.982 (0.929, 1.038)
Setup D	800	10	1.228 (1.141, 1.323)	1.636 (1.561, 1.715)
		20	1.658 (1.524, 1.804)	1.585 (1.510, 1.664)
	1600	10	<i>0.716 (0.660, 0.776)</i>	1.552 (1.437, 1.677)
Setup D	800	10	1.272 (1.149, 1.408)	1.588 (1.481, 1.702)
		20	1.011 (0.948, 1.079)	1.004 (0.973, 1.037)
	1600	10	0.981 (0.923, 1.042)	0.987 (0.960, 1.016)
		20	0.969 (0.891, 1.054)	1.027 (0.987, 1.069)
		20	0.970 (0.899, 1.048)	1.003 (0.968, 1.039)

Table S. 3: Results of **RQ 5** for the experimental setups in Section 4. Comparison of mean squared errors for $\hat{\tau}(\mathbf{x})$ in the different scenarios. Estimates and simultaneous 95 % confidence intervals were obtained from a normal linear mixed model with log-link. Cells printed in bold font correspond to a superior reference of *Robinson _{\hat{W}}* forests, cells printed in italics indicate an inferior reference.

was that the strategy would reduce the effect of overfitting the marginal effect $\hat{\eta}(\mathbf{x})$ on the treatment effect estimate. Overall, however, the estimation of additional nuisance parameters tended to worsen the performance results on average – at least for the binomial model. For the Cox model, Gao’s strategy, which additionally takes the probability for not getting censored into account, did not worsen performance. Further experiments are necessary in which the censoring probability is not constant but depends on covariates \mathbf{x} .

B. Empirical evaluation based on Wager and Athey (2018)

We evaluated the performance of our proposed model-based forest versions also with the study setting of Wager and Athey (2018), which were later reused by Athey *et al.* (2019). Given uniformly distributed covariates $\mathbf{X} \sim U([0, 1]^P)$ of dimensionality $P \in \{10, 20\}$ and a binomially distributed treatment indicator $W \mid \mathbf{X} = \mathbf{x} \sim B(1, \pi(\mathbf{x}))$, the propensity function $\pi(\cdot)$ either did or did not depend on \mathbf{x}

$$\pi(\mathbf{x}) = \begin{cases} \pi \equiv 0.5 \\ \pi(x_1) = 1/4(1 + \beta_{2,4}(x_1)) \\ \pi(x_3) = 1/4(1 + \beta_{2,4}(x_3)) \\ \pi(x_4) = 1/4(1 + \beta_{2,4}(x_4)) \end{cases}$$

where $\beta_{2,4}$ is the β -density with shape 2 and scale 4. The probability $\pi \equiv 0.5$ indicates no confounding and thus a randomized trial. The treatment effect function $\tau(\cdot)$ was either 0 (no treatment effect) or depended on a smooth interaction function of x_1 and x_2

$$\tau(\mathbf{x}) = \begin{cases} \tau \equiv 0 \\ \tau(x_1, x_2) = \prod_{p=1,2} \left(1 + (1 + \exp(-20(x_p - 1/3)))^{-1}\right). \end{cases}$$

The prognostic effect function $\mu(\cdot)$ was either 0 (no prognostic effect) or linear in x_1 or x_3

$$\mu(\mathbf{x}) = \begin{cases} \mu \equiv 0 \\ \mu(x_1) = 2x_1 - 1 \\ \mu(x_3) = 2x_3 - 1. \end{cases}$$

We studied four different simulation models

$$(Y \mid \mathbf{X} = \mathbf{x}, W = w) \sim \begin{cases} N(\mu(\mathbf{x}) + \tau(\mathbf{x})w, 1) & (20a) \\ B(1, \text{expit}(\mu(\mathbf{x}) + \tau(\mathbf{x})w)) & (20b) \\ M \text{ with } \log(O(y_k \mid \mathbf{x}, w)) = \vartheta_k - \mu(\mathbf{x}) - \tau(\mathbf{x})w & (20c) \\ W \text{ with } \log(H(y \mid \mathbf{x}, w)) = 2 \log(y) - \mu(\mathbf{x}) - \tau(\mathbf{x})w & (20d) \end{cases}$$

Model (20a) is a normal linear regression model, model (20b) a binary logistic regression model, model (20c) is a 4-nomial model with log-odds function $\vartheta_k - \mu(\mathbf{x}) - \tau(\mathbf{x})w$ with threshold parameters $\vartheta_k = \text{logit}(k/4)$ for $k = 1, 2, 3$, and model (20d) is a Weibull model with log-cumulative hazard function $2 \log(y) - \mu(\mathbf{x}) - \tau(\mathbf{x})w$. We added 50% random right-censoring to the Weibull-generated data and also applied a Cox proportional hazards model in addition to the Weibull model.

For the additive predictor $\mu(\mathbf{x}) + \tau(\mathbf{x})w$ we considered the 16 scenarios as specified in Table S. 4. Compared to Part A of this table, in Part B half of the (negative) predictive effect is added to the prognostic effect. We term the implied scenario where at least one variable exists which is both prognostic (impact in $\mu(\mathbf{x})$) and predictive (impact in $\tau(\mathbf{x})$) as overlay. $W(x_1)$, $W(x_3)$ and $W(x_4)$ depict that W was drawn from a Bernoulli distribution with $\pi(x_1)$, $\pi(x_3)$ or $\pi(x_4)$, respectively.

In Part A of Table S. 4, the prognostic term and the predictive term are separate and there is only overlay of prognostic and predictive effects when both terms depend on x_1 , *i.e.* x_1 is both prognostic and predictive in this scenario. The treatment assignment probability may

	Additive Predictor	Confounding	Instrument	Heterogeneity	Overlay
Part A	$\mu(x_3) + 0 \cdot W(x_3)$	yes	no	no	no
	$\tau(x_1, x_2)W$	no	no	yes	no
	$\mu(x_1) + \tau(x_1, x_2)W(x_1)$	yes	no	yes	yes
	$\mu(x_1) + \tau(x_1, x_2)W$	no	no	yes	yes
	$\mu(x_3) + \tau(x_1, x_2)W$	no	no	yes	no
	$\mu(x_3) + \tau(x_1, x_2)W(x_3)$	yes	no	yes	no
	$\tau(x_1, x_2)W(x_3)$	no	yes	yes	no
Part B	$\mu(x_3) + \tau(x_1, x_2)W(x_4)$	no	yes	yes	no
	$\mu(x_3) + 0 \cdot (W(x_3) - 0.5)$	yes	no	no	no
	$\tau(x_1, x_2)(W - 0.5)$	no	no	yes	yes
	$\mu(x_1) + \tau(x_1, x_2)(W(x_1) - 0.5)$	yes	no	yes	yes
	$\mu(x_1) + \tau(x_1, x_2)(W - 0.5)$	no	no	yes	yes
	$\mu(x_3) + \tau(x_1, x_2)(W - 0.5)$	no	no	yes	yes
	$\mu(x_3) + \tau(x_1, x_2)(W(x_3) - 0.5)$	yes	no	yes	yes
$\tau(x_1, x_2)(W(x_3) - 0.5)$	no	yes	yes	yes	
	$\mu(x_3) + \tau(x_1, x_2)(W(x_4) - 0.5)$	no	yes	yes	yes

Table S. 4: Experimental setup B. Confounding is present for non-constant propensities $\pi(\mathbf{x})$, an instrumental variable impacts $\pi(\mathbf{x})$ exclusively, heterogeneity of the treatment effect $\tau(\mathbf{x})$ is present when τ is non-constant, and overlay refers to variables being prognostic (impact in $\mu(\mathbf{x})$) and predictive (impact in $\tau(\mathbf{x})$) at the same time.

depend on x_1 , x_3 , or x_4 . In the third scenario, x_1 is a predictive confounder (with impact on μ , τ , and π) and in the last two scenarios, x_3 and x_4 can be understood as instruments with direct impact on treatment assignment but without direct impact on the response. In Part B of this table, half of the predictive effect is added to the prognostic effect, so there is always overlay of both types of effects.

Again, we used random forests to estimate $\pi(\mathbf{x})$ and gradient boosting machines to estimate $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$ as described in Section 4. We also applied the same performance assessment (mean squared error evaluated on 1000 test samples). The results are presented in Figures S. 2 and S. 3. The results for the statistical analysis of RQ 1 to RQ 3 based on a normal linear mixed model are presented in Table S. 5 to S. 7.

Results

For the normal distribution (first row of Figures S. 2 and S. 3), model-based forests with centered W ($Robinson_{\hat{w}}$) performed better than naive model-based forests without centering in case of confounding (columns 1 and 6). If predictive covariates were also prognostic (column 3), the effect of local centering on performance diminished. In case of variables that only influence the treatment assignment but not the outcome (column 7 and 8), solely centering W led to biased results. Especially in this scenario, additional adding $\hat{m}(\mathbf{x})$ as an offset (*Robinson*) is recommended. However, also in all other scenarios *Robinson* model-based forests perform at least as well as $Robinson_{\hat{w}}$ forests – except for the setup without a prognostic effect ($\mu(\mathbf{x}) \equiv 0$, column 2, see also Table reftab:lmeradaptive3).

We obtained similar results for the other distributions as shown in Figures S. 2 and S. 3.

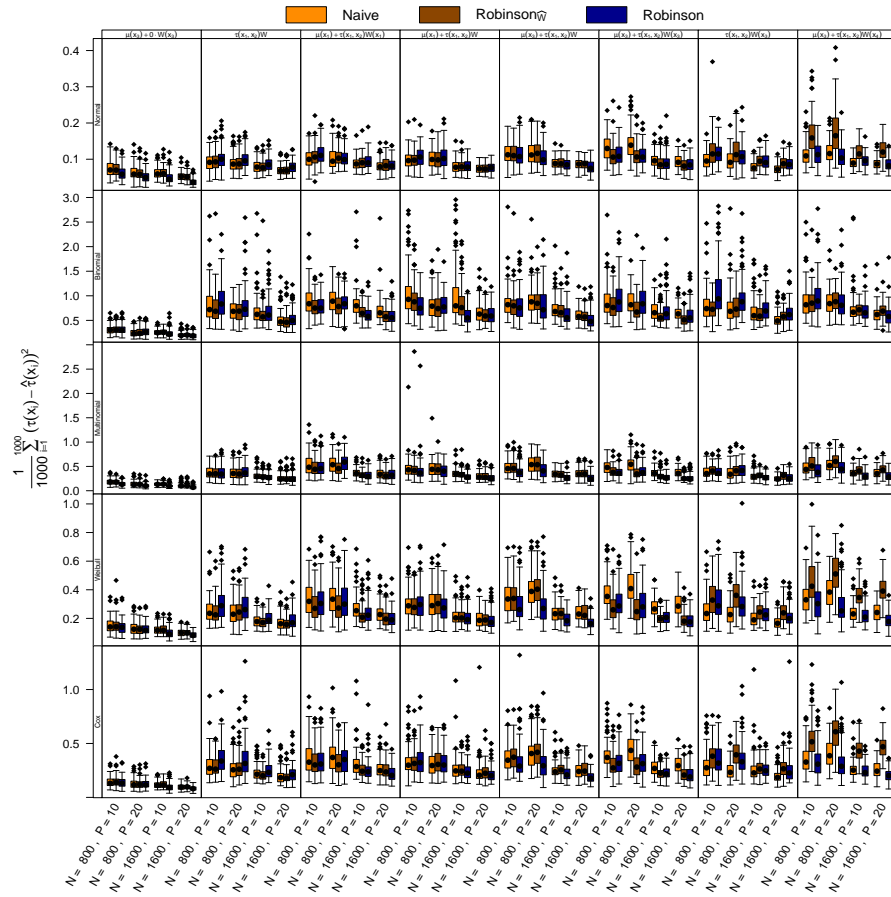


Figure S. 2: Model-based forest results for Part A (Table S. 4), Cox means a Cox model applied to the Weibull data. For the Weibull and Cox model, treatment effects $\tau(\mathbf{x})$ are estimated as conditional log hazard ratios. Direct comparison of model-based forests without centering (Naive), model-based forests with local centering according to Robinson (1988) of Y and W (Robinson) or only of W (Robinson_W).

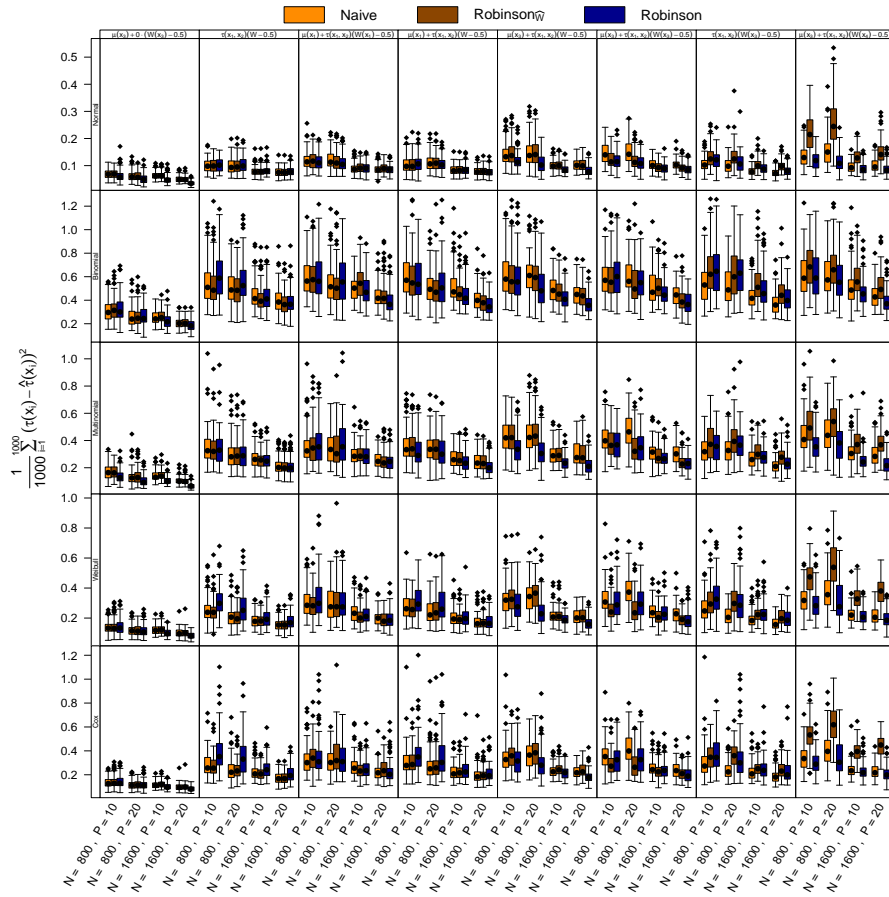


Figure S. 3: Model-based forest results for Part B (Table S. 4), Cox means a Cox model applied to the Weibull data. For the Weibull and Cox model, treatment effects $\tau(\mathbf{x})$ are estimated as conditional log hazard ratios. Direct comparison of model-based forests without centering (Naive), model-based forests with local centering according to Robinson (1988) of Y and W (Robinson) or only of W (Robinson_W).

Forest-based HTE Estimation

Part	DGP	N	P	Mean squared error ratio for RQ 1: Robinson vs. Naive				
				Normal	Bimomial	Multinomial	Weibull	Cox
A	$\mu(x_3) + 0 \cdot W(x_3)$	10	10	0.837 (0.765, 0.919)	1.025 (0.718, 1.474)	0.767 (0.656, 0.888)	0.927 (0.759, 1.118)	0.975 (0.758, 1.254)
		20	10	0.808 (0.725, 0.903)	1.123 (0.826, 1.735)	0.761 (0.624, 0.928)	0.931 (0.756, 1.149)	0.975 (0.738, 1.287)
		40	10	0.792 (0.707, 0.884)	1.152 (0.826, 1.682)	0.742 (0.627, 0.915)	0.924 (0.752, 1.105)	0.982 (0.760, 1.260)
		800	20	0.812 (0.728, 0.900)	1.122 (0.826, 1.682)	0.742 (0.627, 0.915)	0.924 (0.752, 1.105)	0.982 (0.760, 1.260)
		1600	20	1.106 (1.036, 1.180)	3.295 (2.924, 3.714)	0.976 (0.918, 1.037)	1.222 (1.111, 1.344)	1.220 (1.098, 1.356)
		1600	20	1.094 (1.023, 1.171)	1.210 (1.044, 1.402)	1.047 (0.986, 1.112)	1.127 (1.019, 1.248)	1.379 (1.234, 1.541)
	$\tau(x_1, x_2)W$	10	10	1.085 (1.005, 1.171)	1.012 (0.878, 1.167)	0.897 (0.827, 0.972)	1.080 (0.938, 1.245)	1.093 (0.946, 1.262)
		20	10	1.128 (1.034, 1.231)	1.021 (0.826, 1.261)	1.004 (0.918, 1.098)	1.108 (0.954, 1.287)	1.140 (0.969, 1.342)
		40	20	1.118 (1.054, 1.185)	0.805 (0.706, 0.917)	0.924 (0.831, 0.966)	0.970 (0.895, 1.052)	0.916 (0.833, 1.008)
		800	20	0.988 (0.928, 1.051)	0.943 (0.831, 1.070)	1.032 (0.990, 1.074)	0.923 (0.852, 0.998)	0.912 (0.831, 1.002)
		1600	20	1.078 (1.005, 1.156)	0.748 (0.157, 2.252)	0.872 (0.814, 0.935)	0.875 (0.787, 0.979)	0.805 (0.712, 0.910)
		1600	20	1.049 (0.970, 1.133)	0.653 (0.528, 0.753)	0.981 (0.919, 1.048)	0.845 (0.747, 0.956)	0.857 (0.742, 0.990)
B	$\mu(x_3) + \tau(x_1, x_2)W$	10	10	1.076 (1.011, 1.146)	0.465 (0.408, 0.531)	0.361 (0.317, 1.013)	1.013 (0.927, 1.106)	1.039 (0.936, 1.173)
		20	10	1.008 (0.924, 1.150)	0.397 (0.352, 1.159)	0.577 (0.525, 0.929)	0.614 (0.524, 0.929)	0.590 (0.528, 1.014)
		40	10	1.027 (0.943, 1.119)	0.915 (0.759, 1.095)	0.880 (0.809, 0.958)	0.933 (0.812, 1.072)	0.883 (0.740, 1.007)
		800	20	0.984 (0.931, 1.041)	1.166 (1.032, 1.318)	0.860 (0.766, 0.816)	0.881 (0.752, 0.894)	0.951 (0.860, 1.052)
		1600	20	0.875 (0.827, 0.936)	0.897 (0.789, 1.020)	0.800 (0.765, 0.859)	0.690 (0.635, 0.751)	0.724 (0.654, 0.801)
		1600	20	0.951 (0.883, 1.023)	0.946 (0.807, 1.110)	0.776 (0.716, 0.840)	0.890 (0.725, 0.930)	0.880 (0.751, 1.019)
	$\mu(x_3) + \tau(x_1, x_2)W$	10	10	0.902 (0.834, 0.976)	0.848 (0.693, 1.038)	0.741 (0.685, 0.802)	0.688 (0.598, 0.790)	0.735 (0.622, 0.868)
		20	10	0.865 (0.821, 0.911)	1.161 (1.031, 1.308)	0.778 (0.736, 0.832)	0.830 (0.757, 0.889)	0.844 (0.765, 0.912)
		40	20	0.764 (0.726, 0.804)	1.102 (0.974, 1.248)	0.747 (0.711, 0.786)	0.717 (0.665, 0.774)	0.791 (0.666, 0.809)
		800	20	0.944 (0.882, 1.011)	0.282 (0.521, 1.174)	0.744 (0.650, 0.803)	0.748 (0.666, 0.840)	0.804 (0.698, 0.909)
		1600	20	0.948 (0.855, 0.986)	0.387 (0.321, 1.174)	0.697 (0.645, 0.754)	0.632 (0.559, 0.715)	0.701 (0.608, 0.809)
		1600	20	1.218 (1.124, 1.286)	1.377 (1.182, 1.522)	1.037 (0.934, 1.250)	1.295 (1.104, 1.589)	1.228 (1.076, 1.390)
C	$\tau(x_1, x_2)W(x_3)$	10	10	1.184 (1.082, 1.289)	1.454 (1.252, 1.652)	1.170 (0.914, 1.250)	1.458 (1.250, 1.652)	1.458 (1.250, 1.652)
		20	10	1.227 (1.138, 1.333)	1.178 (1.011, 1.372)	1.178 (1.011, 1.372)	1.150 (1.015, 1.304)	0.965 (0.833, 1.105)
		40	20	1.200 (1.106, 1.301)	1.278 (1.052, 1.553)	1.191 (1.063, 1.334)	1.152 (1.051, 1.262)	1.295 (1.094, 1.492)
		800	20	1.017 (0.961, 1.076)	0.999 (0.880, 1.135)	0.901 (0.866, 0.943)	0.877 (0.835, 0.924)	0.915 (0.845, 0.990)
		1600	20	0.907 (0.857, 0.960)	0.841 (0.722, 0.980)	0.901 (0.866, 0.943)	0.780 (0.720, 0.845)	0.810 (0.726, 0.891)
		1600	20	1.048 (0.978, 1.123)	0.841 (0.722, 0.980)	0.847 (0.756, 0.909)	0.946 (0.841, 1.053)	0.932 (0.811, 1.071)
	$\mu(x_3) + \tau(x_1, x_2)W(x_4)$	10	10	0.967 (0.898, 1.042)	0.943 (0.925, 1.133)	0.847 (0.790, 0.910)	0.752 (0.661, 0.857)	0.782 (0.669, 0.913)
		20	10	0.907 (0.820, 1.002)	1.033 (0.928, 1.154)	0.890 (0.827, 0.930)	1.028 (0.867, 1.219)	1.069 (0.850, 1.346)
		40	10	0.818 (0.727, 0.912)	1.047 (0.918, 1.194)	0.749 (0.627, 0.872)	0.980 (0.797, 1.205)	1.025 (0.771, 1.359)
		800	20	0.783 (0.695, 0.885)	0.896 (0.772, 1.041)	0.716 (0.624, 0.879)	0.827 (0.662, 1.032)	0.889 (0.642, 1.177)
		1600	20	0.705 (0.597, 0.831)	0.906 (0.781, 1.092)	0.643 (0.626, 0.821)	0.801 (0.610, 1.050)	0.833 (0.577, 1.204)
		1600	20	1.057 (0.971, 1.107)	1.118 (1.047, 1.194)	0.887 (0.928, 1.050)	1.313 (1.189, 1.440)	1.440 (1.291, 1.606)
D	$\mu(x_3) + \tau(x_1, x_2)W(x_1) - 0.5$	10	10	1.039 (0.956, 1.129)	0.964 (0.880, 1.043)	0.964 (0.886, 1.036)	1.102 (0.974, 1.245)	1.147 (1.004, 1.311)
		20	10	0.954 (0.901, 1.011)	0.971 (0.888, 1.062)	0.983 (0.900, 1.074)	1.140 (0.987, 1.317)	1.240 (1.047, 1.468)
		40	20	0.931 (0.876, 0.986)	0.994 (0.926, 1.068)	0.937 (0.924, 1.054)	0.947 (0.854, 1.051)	1.069 (0.970, 1.179)
		800	20	1.017 (0.944, 1.096)	0.935 (0.858, 1.018)	0.935 (0.904, 1.054)	0.984 (0.870, 1.113)	1.029 (0.892, 1.187)
		1600	20	0.974 (0.900, 1.053)	0.969 (0.916, 1.024)	0.874 (0.827, 0.932)	1.152 (1.061, 1.260)	1.179 (1.078, 1.288)
		1600	20	1.024 (0.962, 1.089)	0.969 (0.916, 1.024)	0.942 (0.838, 0.997)	1.192 (1.091, 1.303)	1.257 (1.138, 1.388)
	$\mu(x_3) + \tau(x_1, x_2)W(x_2) - 0.5$	10	10	0.940 (0.884, 1.000)	0.871 (0.809, 0.947)	0.902 (0.838, 0.971)	1.028 (0.917, 1.153)	1.101 (0.963, 1.259)
		20	10	1.002 (0.928, 1.083)	0.829 (0.752, 0.912)	0.896 (0.828, 0.944)	0.949 (0.849, 1.049)	0.999 (0.898, 1.094)
		40	10	0.831 (0.780, 0.885)	0.957 (0.905, 1.012)	0.845 (0.752, 0.844)	0.790 (0.671, 0.743)	0.807 (0.690, 0.941)
		800	20	0.705 (0.669, 0.744)	0.845 (0.702, 0.900)	0.728 (0.691, 0.766)	0.730 (0.671, 0.773)	0.775 (0.702, 0.856)
		1600	20	0.850 (0.790, 0.915)	0.840 (0.777, 0.909)	0.787 (0.732, 0.847)	0.863 (0.766, 0.972)	0.935 (0.810, 1.081)
		1600	20	0.791 (0.734, 0.851)	0.805 (0.757, 0.878)	0.721 (0.668, 0.779)	0.780 (0.682, 0.892)	0.807 (0.687, 0.948)
E	$\mu(x_3) + \tau(x_1, x_2)W(x_3) - 0.5$	10	10	0.810 (0.776, 0.853)	0.911 (0.857, 0.968)	0.927 (0.888, 0.973)	0.979 (0.908, 1.056)	0.927 (0.844, 1.017)
		20	10	0.720 (0.684, 0.759)	0.909 (0.844, 0.978)	0.757 (0.721, 0.794)	0.787 (0.733, 0.844)	0.783 (0.719, 0.853)
		40	20	0.884 (0.825, 0.948)	0.909 (0.844, 0.978)	0.870 (0.814, 0.930)	0.946 (0.855, 1.045)	0.945 (0.831, 1.075)
		800	20	0.839 (0.772, 0.880)	0.847 (0.776, 0.935)	0.765 (0.701, 0.813)	0.830 (0.738, 0.934)	0.832 (0.710, 0.965)
		1600	20	1.152 (1.085, 1.222)	1.277 (1.202, 1.383)	1.082 (1.023, 1.144)	1.300 (1.200, 1.409)	1.212 (1.101, 1.334)
		1600	20	1.107 (1.040, 1.179)	1.228 (1.154, 1.306)	1.127 (1.058, 1.180)	1.470 (1.335, 1.618)	1.488 (1.335, 1.650)
	$\mu(x_3) + \tau(x_1, x_2)W(x_4) - 0.5$	10	10	1.126 (1.052, 1.227)	1.129 (1.076, 1.226)	1.096 (1.009, 1.194)	1.258 (1.072, 1.410)	1.282 (1.078, 1.477)
		20	10	0.866 (0.816, 0.911)	1.021 (0.966, 1.080)	0.976 (0.921, 1.035)	0.823 (0.784, 0.864)	0.891 (0.786, 0.966)
		40	20	0.718 (0.682, 0.755)	0.903 (0.821, 0.971)	0.843 (0.806, 0.883)	0.863 (0.806, 0.925)	0.850 (0.780, 0.925)
		800	20	0.935 (0.869, 1.006)	0.903 (0.841, 0.971)	0.790 (0.737, 0.846)	0.935 (0.837, 1.045)	0.954 (0.833, 1.093)
		1600	20	0.861 (0.801, 0.935)	0.897 (0.822, 0.978)	0.762 (0.706, 0.821)	0.899 (0.798, 1.011)	0.911 (0.786, 1.053)
		1600	20	0.897 (0.822, 0.978)	0.897 (0.822, 0.978)	0.762 (0.706, 0.821)	0.899 (0.798, 1.011)	0.911 (0.786, 1.053)

Table S. 5: Results of RQ 1 for the experimental setups in Section B. Comparison of mean squared errors for $\hat{\tau}(x)$ in the different scenarios. Estimates and simultaneous 95% confidence intervals were obtained from a normal linear mixed model with log-link. Cells printed in bold font correspond to a superior reference of the naive model-based forests, cells printed in italics indicate an inferior reference of naive model-based forests.

DGP	N	P	Mean squared error ratio for RQ 2: Robinson _W vs. Naive			
			Normal	Binomial	Multinomial	Cox
$\mu(x_3) + 0 \cdot W(x_3)$	800	10	1.152 (1.076, 1.238)	1.012 (0.709, 1.446)	1.284 (1.106, 1.491)	1.109 (0.922, 1.335)
	1600	20	1.186 (1.059, 1.328)	0.917 (0.598, 1.408)	1.304 (1.069, 1.591)	1.018 (0.821, 1.263)
	1600	20	1.277 (1.137, 1.436)	1.131 (0.711, 1.800)	1.375 (1.116, 1.693)	1.238 (0.967, 1.584)
$\tau(x_1, x_2)W$	800	20	1.392 (1.197, 1.618)	0.537 (0.439, 0.577)	1.430 (1.096, 1.866)	1.200 (0.804, 1.790)
	1600	20	0.913 (0.833, 0.976)	0.608 (0.605, 0.938)	0.954 (0.898, 1.013)	0.787 (0.706, 0.876)
	1600	20	0.888 (0.817, 0.969)	0.938 (0.756, 1.164)	0.989 (0.904, 1.061)	0.766 (0.688, 0.854)
$\mu(x_1) + \tau(x_1, x_2)W(x_1)$	800	20	0.948 (0.896, 1.004)	1.000 (0.867, 1.153)	0.989 (0.916, 1.085)	0.836 (0.697, 0.978)
	1600	20	1.038 (0.977, 1.103)	0.941 (0.823, 1.076)	0.817 (0.781, 0.854)	0.898 (0.892, 1.089)
	1600	20	0.951 (0.928, 1.018)	0.969 (0.782, 1.175)	0.924 (0.862, 0.991)	0.911 (0.810, 1.025)
$\mu(x_1) + \tau(x_1, x_2)W$	800	20	0.938 (0.881, 0.998)	1.826 (1.859, 2.098)	0.927 (0.854, 1.161)	1.148 (0.993, 1.327)
	1600	20	0.945 (0.889, 1.004)	2.057 (1.722, 2.456)	1.096 (1.037, 1.158)	0.932 (0.841, 1.033)
	1600	20	0.978 (0.904, 1.058)	1.018 (0.848, 1.221)	1.196 (1.108, 1.291)	1.052 (0.946, 1.170)
$\mu(x_3) + \tau(x_1, x_2)W$	800	10	1.018 (0.963, 1.076)	0.823 (0.727, 0.933)	1.134 (1.042, 1.234)	1.058 (0.919, 1.217)
	1600	20	1.147 (0.974, 1.340)	1.014 (0.863, 1.192)	1.257 (1.191, 1.326)	1.231 (1.130, 1.340)
	1600	20	1.109 (0.975, 1.330)	1.138 (0.927, 1.397)	1.375 (1.170, 1.582)	1.145 (1.016, 1.285)
$\mu(x_3) + \tau(x_1, x_2)W(x_3)$	800	20	1.108 (1.024, 1.199)	0.794 (0.702, 0.898)	1.258 (1.256, 1.469)	1.434 (1.217, 1.690)
	1600	20	0.971 (0.918, 1.027)	0.777 (0.679, 0.889)	0.961 (0.898, 1.023)	0.894 (0.841, 1.045)
	1600	20	0.980 (0.925, 1.038)	0.819 (0.686, 0.977)	1.127 (1.039, 1.222)	0.877 (0.798, 0.964)
	1600	20	0.972 (0.902, 1.049)	0.843 (0.692, 1.026)	1.002 (0.916, 1.096)	0.961 (0.841, 1.099)
$\tau(x_1, x_2)W(x_3)$	800	10	1.043 (0.988, 1.100)	0.570 (0.502, 0.646)	1.074 (1.014, 1.138)	1.006 (0.854, 1.186)
	1600	20	1.103 (1.045, 1.165)	0.721 (0.634, 0.830)	0.982 (0.930, 1.036)	1.055 (0.961, 1.159)
	1600	20	1.014 (0.948, 1.084)	0.774 (0.660, 0.910)	1.140 (1.053, 1.236)	1.116 (1.018, 1.223)
	1600	20	1.021 (0.949, 1.098)	0.883 (0.736, 1.058)	1.151 (1.063, 1.246)	1.028 (0.925, 1.206)
$\mu(x_3) + \tau(x_1, x_2)W(x_4)$	800	20	1.061 (0.920, 1.202)	0.619 (0.588, 0.646)	1.273 (1.212, 1.327)	1.158 (1.028, 1.306)
	1600	20	1.268 (1.193, 1.348)	0.819 (0.808, 1.154)	1.331 (1.292, 1.477)	1.071 (0.964, 1.183)
	1600	20	1.467 (1.375, 1.565)	1.154 (0.967, 1.377)	1.427 (1.336, 1.524)	1.604 (1.451, 1.774)
$\mu(x_3) + 0 \cdot (W(x_3) - 0.5)$	800	10	1.100 (0.995, 1.215)	1.000 (0.897, 1.115)	1.211 (1.068, 1.374)	1.160 (0.927, 2.422)
	1600	20	1.194 (1.059, 1.345)	0.956 (0.839, 1.091)	1.352 (1.147, 1.594)	0.985 (0.787, 1.233)
	1600	20	1.309 (1.160, 1.478)	1.161 (1.008, 1.345)	1.372 (1.167, 1.613)	1.017 (0.827, 1.250)
	1600	20	1.395 (1.181, 1.647)	1.119 (0.932, 1.344)	1.526 (1.193, 1.951)	1.232 (0.989, 1.535)
$\tau(x_1, x_2)(W - 0.5)$	800	10	0.994 (0.931, 1.061)	0.843 (0.794, 0.895)	0.999 (0.946, 1.055)	1.245 (0.949, 1.634)
	1600	20	0.975 (0.913, 1.040)	0.871 (0.814, 0.931)	1.017 (0.956, 1.082)	0.791 (0.725, 0.862)
	1600	20	0.965 (0.888, 1.048)	0.969 (0.893, 1.052)	0.994 (0.924, 1.071)	0.746 (0.675, 0.825)
	1600	20	1.054 (0.996, 1.116)	0.998 (0.946, 1.053)	0.953 (0.906, 1.002)	0.859 (0.780, 0.949)
$\mu(x_1) + \tau(x_1, x_2)(W(x_1) - 0.5)$	800	20	1.035 (0.976, 1.097)	0.881 (0.838, 0.927)	0.878 (0.831, 0.924)	0.668 (0.597, 0.747)
	1600	20	1.022 (0.950, 1.100)	1.104 (1.031, 1.182)	1.029 (0.984, 1.098)	0.937 (0.835, 1.042)
	1600	20	1.061 (0.983, 1.146)	1.113 (1.024, 1.210)	1.020 (0.944, 1.103)	1.003 (0.927, 1.220)
$\mu(x_1) + \tau(x_1, x_2)(W - 0.5)$	800	20	0.986 (0.926, 1.048)	0.952 (0.899, 1.009)	1.128 (1.068, 1.192)	0.960 (0.846, 1.090)
	1600	20	1.066 (1.003, 1.134)	0.875 (0.818, 0.935)	1.057 (0.999, 1.120)	0.849 (0.782, 0.923)
	1600	20	1.004 (0.928, 1.087)	1.084 (1.005, 1.169)	1.075 (1.001, 1.162)	0.851 (0.735, 0.895)
$\mu(x_3) + \tau(x_1, x_2)(W - 0.5)$	800	20	1.237 (1.175, 1.302)	1.091 (0.993, 1.200)	1.144 (1.051, 1.246)	0.984 (0.831, 1.051)
	1600	20	1.421 (1.348, 1.498)	1.167 (1.098, 1.243)	1.260 (1.197, 1.326)	0.964 (0.791, 1.035)
	1600	20	1.852 (1.646, 2.058)	1.391 (1.321, 1.464)	1.302 (1.231, 1.374)	1.175 (1.086, 1.271)
$\mu(x_3) + \tau(x_1, x_2)(W(x_3) - 0.5)$	800	20	1.261 (1.171, 1.359)	1.216 (1.113, 1.328)	1.393 (1.291, 1.501)	1.342 (1.217, 1.480)
	1600	20	1.028 (0.973, 1.087)	0.907 (0.913, 1.024)	1.003 (0.954, 1.055)	1.151 (1.047, 1.266)
	1600	20	1.067 (1.007, 1.131)	0.982 (0.921, 1.047)	0.999 (0.947, 1.055)	1.151 (1.047, 1.266)
	1600	20	1.027 (0.955, 1.105)	1.129 (1.050, 1.214)	1.044 (0.974, 1.118)	1.342 (1.217, 1.480)
$\tau(x_1, x_2)(W(x_3) - 0.5)$	800	10	1.074 (0.997, 1.157)	1.074 (0.981, 1.177)	1.024 (0.943, 1.111)	0.857 (0.776, 0.952)
	1600	20	1.078 (1.022, 1.137)	1.007 (0.937, 1.058)	1.115 (1.060, 1.174)	0.902 (0.809, 1.006)
	1600	20	1.204 (1.140, 1.271)	1.001 (0.947, 1.058)	1.088 (1.037, 1.142)	0.994 (0.874, 1.129)
	1600	20	1.152 (1.075, 1.236)	1.111 (1.039, 1.188)	1.140 (1.068, 1.216)	0.925 (0.846, 1.011)
$\mu(x_3) + \tau(x_1, x_2)(W(x_4) - 0.5)$	800	20	1.192 (1.106, 1.286)	1.116 (1.031, 1.208)	1.192 (1.106, 1.285)	1.006 (0.919, 1.101)
	1600	20	1.308 (1.523, 1.997)	1.124 (1.066, 1.184)	1.438 (1.374, 1.505)	1.024 (0.903, 1.161)
	1600	20	2.476 (2.247, 2.583)	1.291 (1.206, 1.381)	1.482 (1.388, 1.584)	1.044 (0.928, 1.174)
	1600	20	1.484 (1.391, 1.583)	1.325 (1.222, 1.437)	1.686 (1.573, 1.808)	1.613 (1.505, 1.728)
	1600	20	1.728 (1.622, 1.840)	1.325 (1.222, 1.437)	1.686 (1.573, 1.808)	1.774 (1.627, 1.934)

Table S. 6: Results of RQ 2 for the experimental setups in Section B. Comparison of mean squared errors for $\hat{\tau}(x)$ in the different scenarios. Estimates and simultaneous 95 % confidence intervals were obtained from a normal linear mixed model with log-link. Cells printed in bold font correspond to a superior reference of the naive model-based forests, cells printed in italics indicate an inferior reference of naive model-based forests.

Forest-based HTE Estimation

		Mean squared error ratio for RQ 3: Robinson vs. Robinson $_{W^*}$									
Part	DCGP	N	P	Normal	Bimomial	Multinomial	Weibull	Cox			
A	$\mu(x_3) + 0 \cdot W(x_3)$	800	10	0.832 (0.822)	0.985 (1.410)	0.750 (0.202)	0.992 (1.085)	0.941 (1.201)			
		1600	20	0.842 (0.755, 0.944)	1.090 (0.710, 1.673)	0.767 (0.628, 0.905)	0.982 (0.792, 1.218)	1.013 (0.753, 1.345)			
		800	10	0.719 (0.618, 0.835)	0.923 (0.518, 1.644)	0.699 (0.536, 0.913)	0.727 (0.591, 0.866)	0.831 (0.614, 1.125)			
		1600	20	1.095 (1.026, 1.168)	1.863 (1.322, 2.004)	1.002 (0.942, 1.065)	1.274 (1.155, 1.404)	1.271 (1.141, 1.416)			
		800	10	1.096 (1.024, 1.172)	1.238 (1.066, 1.438)	1.049 (0.987, 1.114)	1.103 (0.998, 1.220)	1.305 (1.172, 1.453)			
		1600	20	1.086 (1.007, 1.173)	1.217 (1.041, 1.423)	0.943 (0.869, 1.024)	1.138 (0.983, 1.317)	1.225 (1.051, 1.428)			
		800	10	1.126 (1.032, 1.228)	1.066 (0.856, 1.333)	1.012 (0.925, 1.106)	1.140 (0.979, 1.328)	1.211 (1.023, 1.434)			
		1600	20	1.055 (0.906, 1.116)	1.000 (0.876, 1.153)	1.042 (0.995, 1.092)	1.114 (1.022, 1.214)	1.014 (0.933, 1.121)			
		800	10	0.963 (0.906, 1.023)	1.062 (0.926, 1.214)	0.979 (0.936, 1.049)	1.225 (1.171, 1.281)	1.090 (1.001, 1.187)			
		1600	20	1.052 (0.984, 1.127)	0.994 (0.776, 1.213)	0.942 (0.876, 1.019)	1.082 (1.009, 1.169)	1.098 (0.975, 1.235)			
	$\mu(x_3) + \tau(x_1, x_2)W$	800	10	1.066 (1.002, 1.135)	1.042 (0.725, 1.628)	0.912 (0.752, 1.072)	0.938 (0.920, 0.956)	0.938 (0.920, 0.956)			
		1600	20	1.059 (0.906, 1.125)	1.017 (0.883, 1.171)	0.912 (0.883, 1.171)	0.912 (0.883, 1.171)	0.912 (0.883, 1.171)			
		800	10	1.022 (0.945, 1.106)	0.983 (0.819, 1.179)	0.886 (0.775, 0.992)	0.886 (0.775, 0.992)	0.886 (0.775, 0.992)			
		1600	20	1.025 (0.941, 1.117)	0.983 (0.824, 1.179)	0.886 (0.810, 0.960)	0.945 (0.822, 1.088)	0.897 (0.767, 1.048)			
		800	10	0.982 (0.930, 1.038)	0.940 (0.824, 1.072)	0.796 (0.754, 0.839)	0.813 (0.746, 0.885)	0.697 (0.651, 0.770)			
		1600	20	0.872 (0.824, 0.922)	0.986 (0.838, 1.159)	0.784 (0.725, 0.838)	0.679 (0.624, 0.738)	0.697 (0.651, 0.770)			
		800	10	0.955 (0.885, 1.026)	0.986 (0.838, 1.159)	0.784 (0.725, 0.838)	0.832 (0.725, 0.933)	0.853 (0.728, 0.987)			
		1600	20	0.905 (0.834, 0.976)	0.879 (0.716, 1.079)	0.706 (0.681, 0.736)	0.676 (0.589, 0.777)	0.697 (0.592, 0.821)			
		800	10	1.030 (0.974, 1.090)	1.260 (1.114, 1.425)	0.943 (0.888, 1.002)	1.118 (1.020, 1.226)	1.066 (0.957, 1.189)			
		1600	20	1.021 (0.963, 1.081)	1.267 (1.125, 1.472)	1.037 (0.997, 1.120)	1.140 (1.037, 1.253)	1.106 (0.983, 1.236)			
	$\mu(x_3) + \tau(x_1, x_2)W(x_3)$	800	10	1.028 (0.954, 1.103)	1.187 (0.975, 1.445)	0.998 (0.912, 1.092)	0.970 (0.847, 1.132)	0.994 (0.833, 1.171)			
		1600	20	0.950 (0.909, 1.012)	1.187 (0.975, 1.445)	0.998 (0.912, 1.092)	0.970 (0.847, 1.132)	0.994 (0.833, 1.171)			
		800	10	0.906 (0.858, 0.957)	1.386 (1.158, 1.601)	1.019 (0.965, 1.075)	0.877 (0.799, 0.959)	0.866 (0.796, 0.939)			
		1600	20	0.979 (0.911, 1.053)	1.291 (1.099, 1.516)	0.877 (0.810, 0.950)	0.933 (0.834, 1.044)	0.947 (0.829, 1.081)			
		800	10	0.979 (0.911, 1.053)	1.133 (0.945, 1.359)	0.869 (0.808, 0.941)	0.864 (0.766, 0.973)	0.973 (0.852, 1.110)			
		1600	20	0.671 (0.640, 0.704)	1.229 (1.095, 1.380)	0.772 (0.735, 0.811)	0.677 (0.630, 0.728)	0.626 (0.575, 0.682)			
		800	10	0.586 (0.555, 0.615)	0.981 (0.865, 1.113)	0.786 (0.755, 0.821)	0.588 (0.546, 0.633)	0.552 (0.507, 0.602)			
		1600	20	0.789 (0.742, 0.838)	0.975 (0.830, 1.146)	0.724 (0.677, 0.774)	0.623 (0.564, 0.689)	0.561 (0.499, 0.631)			
		800	10	0.682 (0.639, 0.727)	0.867 (0.726, 1.034)	0.701 (0.656, 0.748)	0.463 (0.415, 0.519)	0.431 (0.377, 0.492)			
		1600	20	0.930 (0.823, 1.005)	1.000 (0.897, 1.113)	0.836 (0.728, 0.937)	0.993 (0.830, 1.173)	1.015 (0.811, 1.270)			
B	$\mu(x_3) + 0 \cdot W(x_3) - 0.5$	800	10	0.838 (0.743, 0.944)	1.046 (0.917, 1.192)	0.740 (0.621, 0.822)	0.984 (0.800, 1.209)	1.013 (0.766, 1.341)			
		1600	20	0.717 (0.607, 0.846)	0.984 (0.744, 1.073)	0.655 (0.513, 0.838)	0.870 (0.632, 1.054)	0.815 (0.566, 1.172)			
		800	10	1.006 (0.942, 1.075)	1.186 (1.118, 1.259)	1.001 (0.948, 1.057)	1.265 (1.160, 1.379)	1.467 (1.331, 1.617)			
		1600	20	1.026 (0.961, 1.096)	1.149 (1.074, 1.228)	0.983 (0.925, 1.046)	1.340 (1.213, 1.482)	1.497 (1.338, 1.675)			
		800	10	1.037 (0.954, 1.126)	1.032 (0.951, 1.120)	1.008 (0.934, 1.082)	1.119 (0.988, 1.266)	1.210 (1.054, 1.389)			
		1600	20	1.027 (0.942, 1.120)	1.035 (0.944, 1.136)	1.008 (0.917, 1.096)	1.151 (0.996, 1.331)	1.229 (1.099, 1.454)			
		800	10	0.948 (0.896, 1.004)	1.002 (0.950, 1.057)	1.050 (0.998, 1.104)	1.173 (1.086, 1.267)	1.056 (0.965, 1.154)			
		1600	20	0.967 (0.912, 1.025)	1.135 (1.067, 1.207)	0.972 (0.911, 1.038)	0.981 (0.907, 1.061)	0.916 (0.836, 1.004)			
		800	10	0.978 (0.909, 1.053)	0.906 (0.846, 0.970)	0.972 (0.911, 1.038)	1.072 (0.960, 1.197)	1.067 (0.937, 1.216)			
		1600	20	0.942 (0.872, 1.018)	0.898 (0.826, 0.976)	0.880 (0.807, 1.000)	1.042 (0.917, 1.183)	1.067 (0.937, 1.216)			
	$\mu(x_3) + \tau(x_1, x_2)W - 0.5$	800	10	1.015 (0.954, 1.079)	1.030 (0.891, 1.115)	0.887 (0.839, 0.937)	1.108 (1.084, 1.280)	1.287 (1.084, 1.269)			
		1600	20	0.996 (0.920, 1.078)	0.927 (0.855, 0.995)	0.927 (0.861, 0.990)	0.970 (0.932, 1.023)	0.970 (0.932, 1.023)			
		800	10	0.964 (0.884, 1.050)	0.916 (0.833, 1.007)	0.874 (0.802, 0.951)	1.007 (0.876, 1.156)	1.037 (0.900, 1.209)			
		1600	20	0.809 (0.768, 0.851)	0.897 (0.841, 1.055)	0.794 (0.752, 0.835)	0.851 (0.787, 0.921)	0.869 (0.790, 0.955)			
		800	10	0.704 (0.668, 0.743)	0.807 (0.804, 0.913)	0.719 (0.688, 0.757)	0.707 (0.651, 0.768)	0.745 (0.676, 0.822)			
		1600	20	0.839 (0.780, 0.903)	0.890 (0.821, 0.965)	0.800 (0.743, 0.861)	0.856 (0.759, 0.965)	0.883 (0.767, 1.017)			
		800	10	0.793 (0.737, 0.854)	0.823 (0.755, 0.899)	0.718 (0.665, 0.775)	0.758 (0.663, 0.865)	0.758 (0.648, 0.887)			
		1600	20	0.972 (0.920, 1.028)	1.034 (0.977, 1.095)	0.977 (0.948, 1.048)	1.218 (1.120, 1.325)	1.167 (1.051, 1.290)			
		800	10	0.977 (0.884, 0.999)	1.018 (0.955, 1.085)	1.001 (0.948, 1.056)	1.183 (1.086, 1.288)	1.164 (1.050, 1.291)			
		1600	20	0.974 (0.903, 1.047)	0.885 (0.823, 0.952)	0.938 (0.894, 1.027)	1.109 (0.989, 1.236)	1.028 (0.899, 1.176)			
	$\mu(x_3) + \tau(x_1, x_2)W(x_3) - 0.5$	800	10	0.931 (0.864, 1.003)	0.931 (0.850, 1.020)	0.977 (0.930, 1.060)	1.006 (0.888, 1.144)	1.099 (0.928, 1.128)			
		1600	20	0.831 (0.784, 0.884)	0.910 (0.826, 0.999)	0.877 (0.832, 0.926)	0.910 (0.826, 0.999)	0.910 (0.826, 0.999)			
		800	10	0.831 (0.784, 0.884)	0.910 (0.826, 0.999)	0.877 (0.832, 0.926)	0.910 (0.826, 0.999)	0.910 (0.826, 0.999)			
		1600	20	0.868 (0.808, 0.931)	0.900 (0.848, 0.963)	0.889 (0.828, 0.970)	0.889 (0.828, 0.970)	0.941 (0.822, 1.077)			
		800	10	0.544 (0.501, 0.548)	0.690 (0.642, 0.738)	0.699 (0.664, 0.738)	0.690 (0.576, 0.664)	0.564 (0.517, 0.615)			
		1600	20	0.441 (0.422, 0.461)	0.872 (0.825, 0.921)	0.719 (0.680, 0.751)	0.675 (0.632, 0.720)	0.588 (0.525, 0.659)			
		800	10	0.674 (0.632, 0.719)	0.775 (0.721, 0.829)	0.775 (0.721, 0.829)	0.690 (0.573, 0.694)	0.588 (0.525, 0.659)			
		1600	20	0.579 (0.543, 0.617)	0.754 (0.696, 0.818)	0.593 (0.553, 0.636)	0.517 (0.468, 0.570)	0.470 (0.416, 0.530)			

Table S. 7: Results of **RQ 3** for the experimental setups in Section B. Comparison of mean squared errors for $\hat{\tau}(x)$ in the different scenarios. Estimates and simultaneous 95% confidence intervals were obtained from a normal linear mixed model with log-link. Cells printed in bold font correspond to a superior reference of $Robinson_{W^*}$, cells printed in italics indicate an inferior reference of $Robinson_{W^*}$.

Overlay of prognostic and predictive effects (Part B compared to Part A) did slightly worsen the performance of all methods in smaller samples (except in the absence of a predictive effect, see first column of both figures).

We also inspected if the performance of model-based forests degrades for the Weibull data when the forests do not take the true underlying model as their base model. We compared the performance of model-based forests when using a Cox model compared to a Weibull model (Last row of Figures S. 2 & S. 3). Although knowledge of the true functional form does not enter the Cox modeling process, it did not lead to a major decrease in performance.

C. Dependence plots

Dependence plots depict the treatment effect τ on the prepartum variables - scatter plots for continuous covariates and boxplots for categorical covariates. For categorical covariates, diamonds display the mean effect per group, and for continuous covariates, we provide the smooth conditional mean effect function calculated by a generalized additive model (GAM) with a single smooth term - the covariate under consideration. This evaluation scheme closely follows [Dandl *et al.* \(2022\)](#).

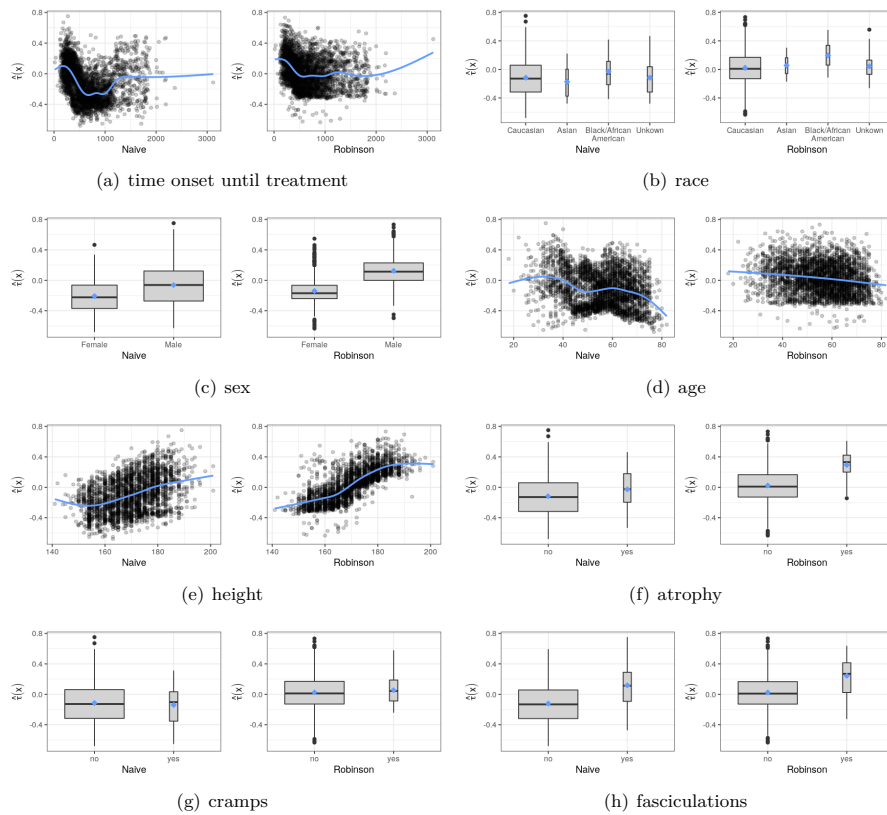


Figure S. 4: Survival time: dependency plot of individual average treatment effects calculated by model-based forest without orthogonalization (left), with Robinson orthogonalization (right). Blue lines and diamond points depict (smooth conditional) mean effects.

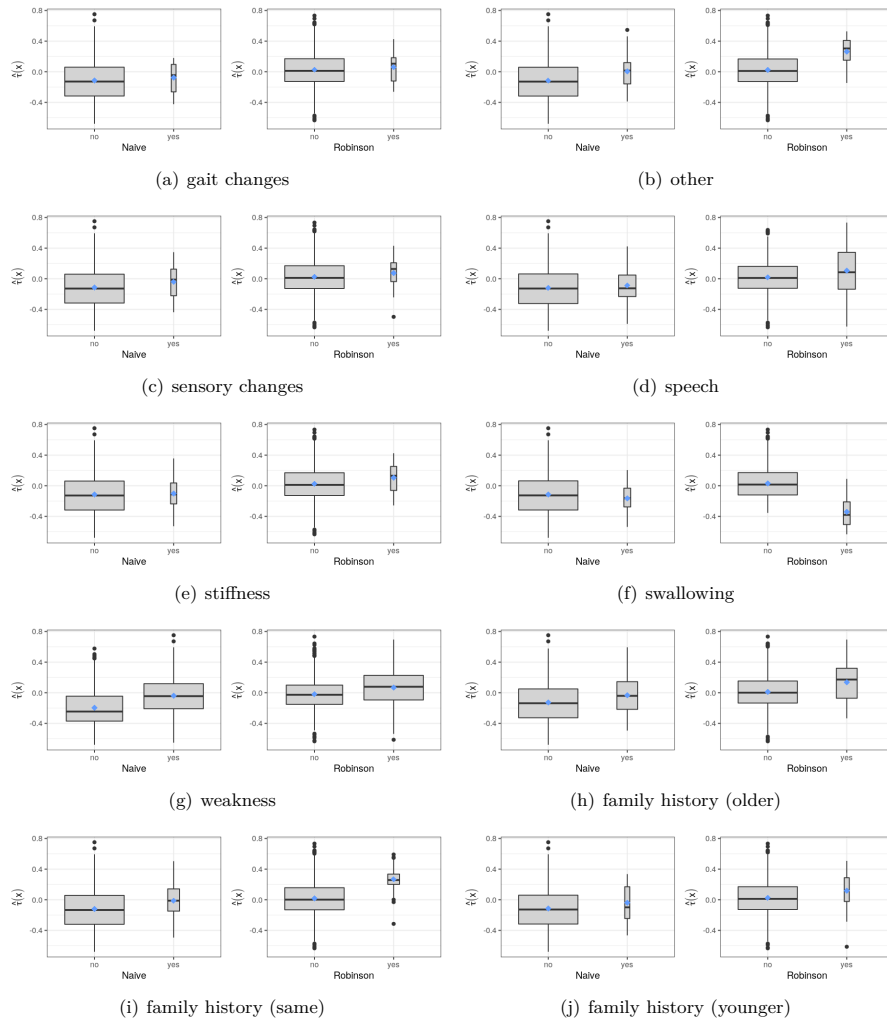


Figure S. 5: Survival time: dependency plot of individual average treatment effects calculated by model-based forest without orthogonalization (left), with Robinson orthogonalization (right). Blue lines and diamond points depict (smooth conditional) mean effects.

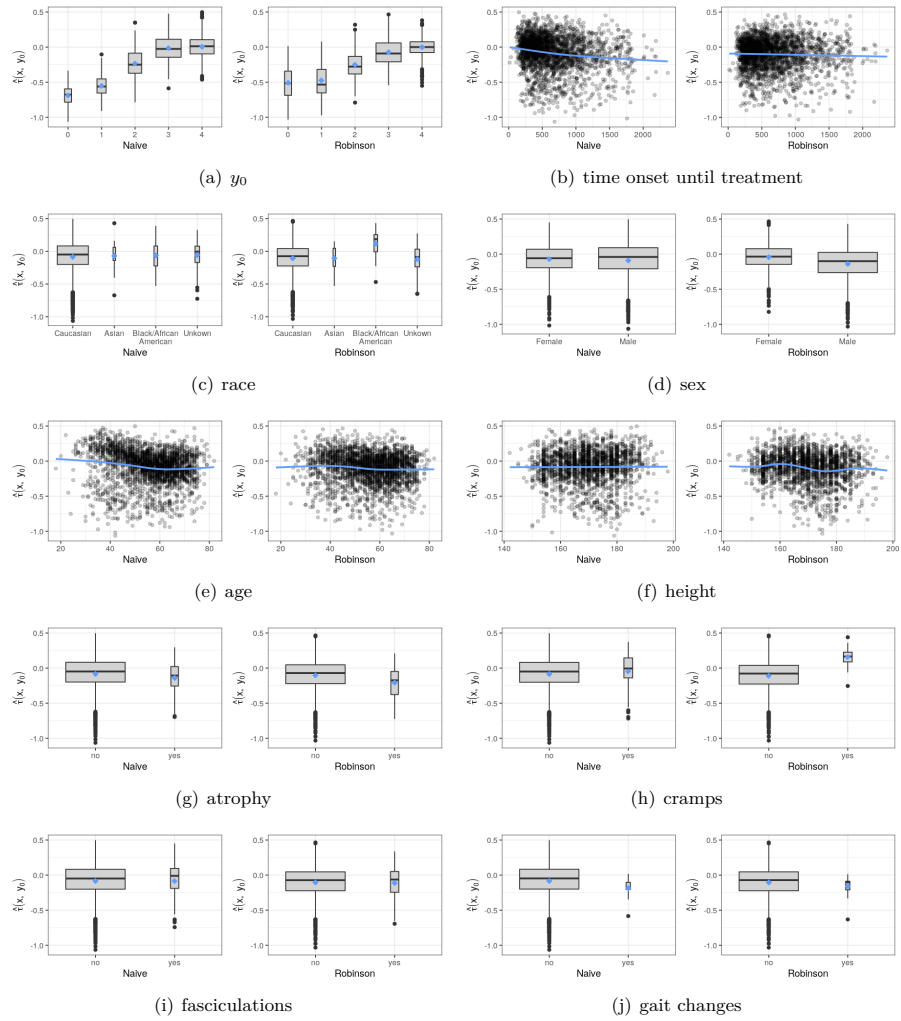


Figure S. 6: Handwriting ability score: dependency plot of individual average treatment effects calculated by model-based forest without (left) and with Robinson centering (right). Blue lines and diamond points depict (smooth conditional) mean effects.

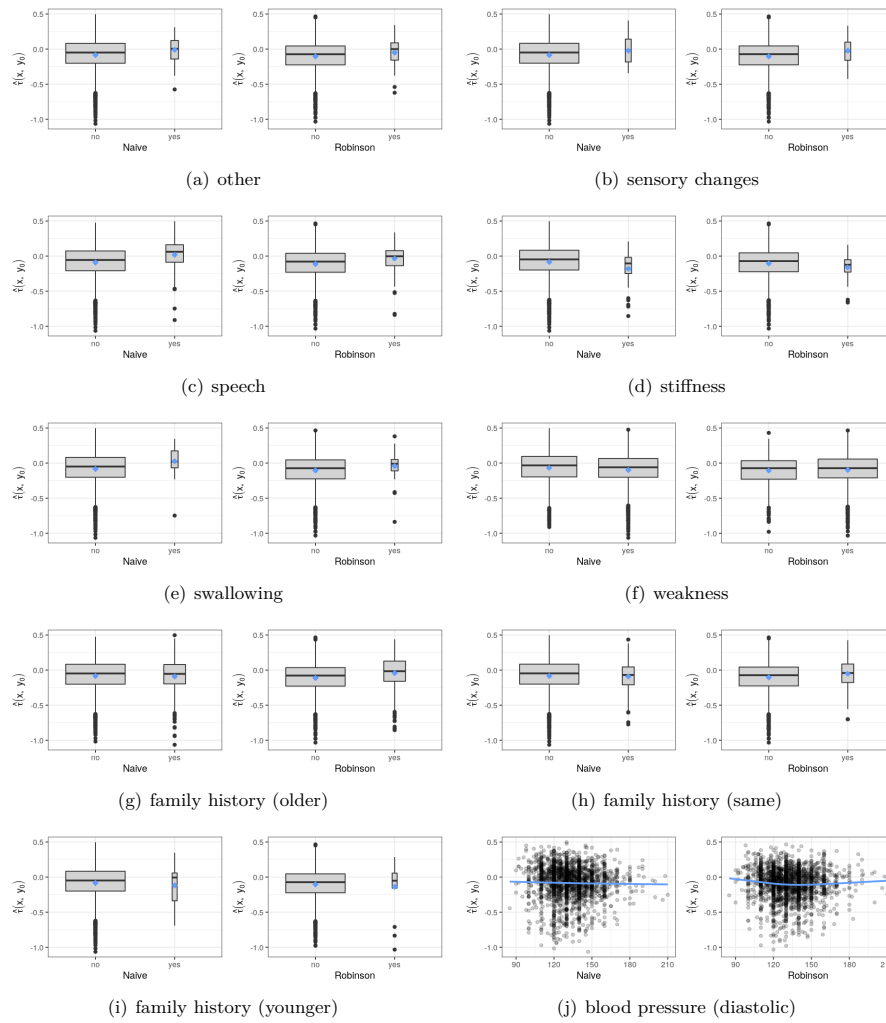


Figure S. 7: Handwriting ability score: dependency plot of individual average treatment effects calculated by model-based forest without (left) and with Robinson centering (right). Blue lines and diamond points depict (smooth conditional) mean effects.

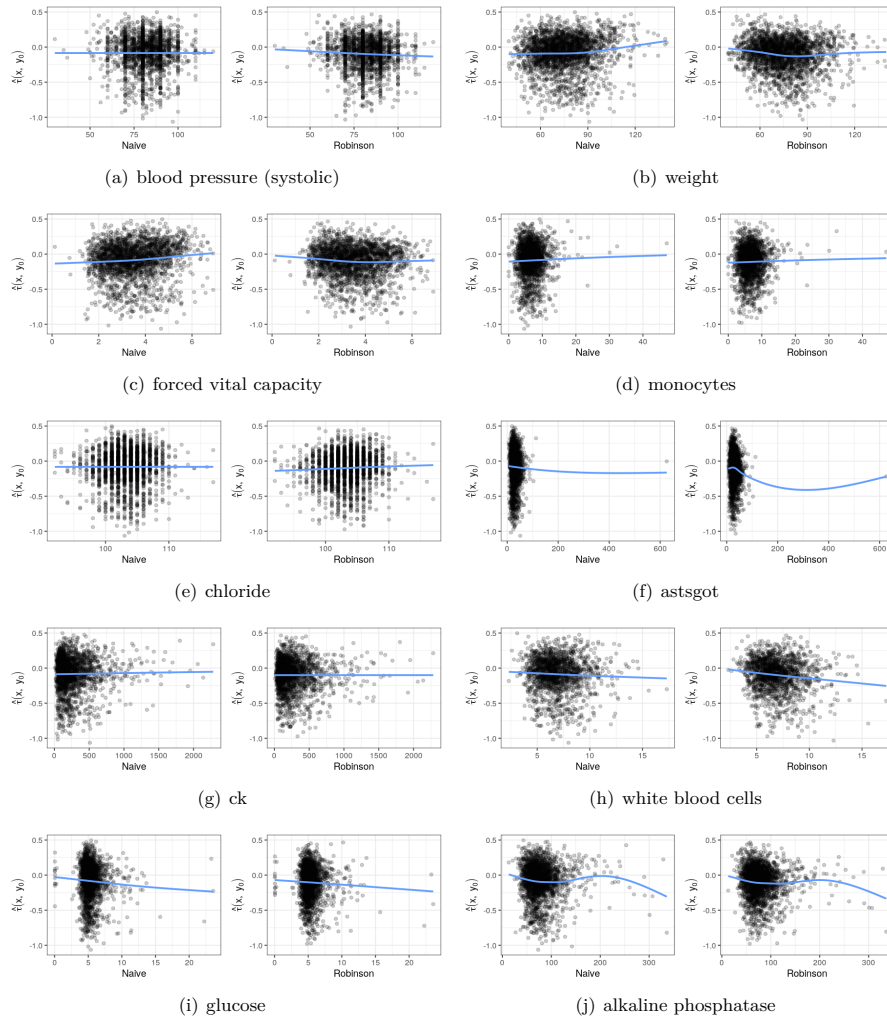


Figure S. 8: Handwriting ability score: dependency plot of individual average treatment effects calculated by model-based forest without (left) and with Robinson centering (right). Blue lines and diamond points depict (smooth conditional) mean effects.

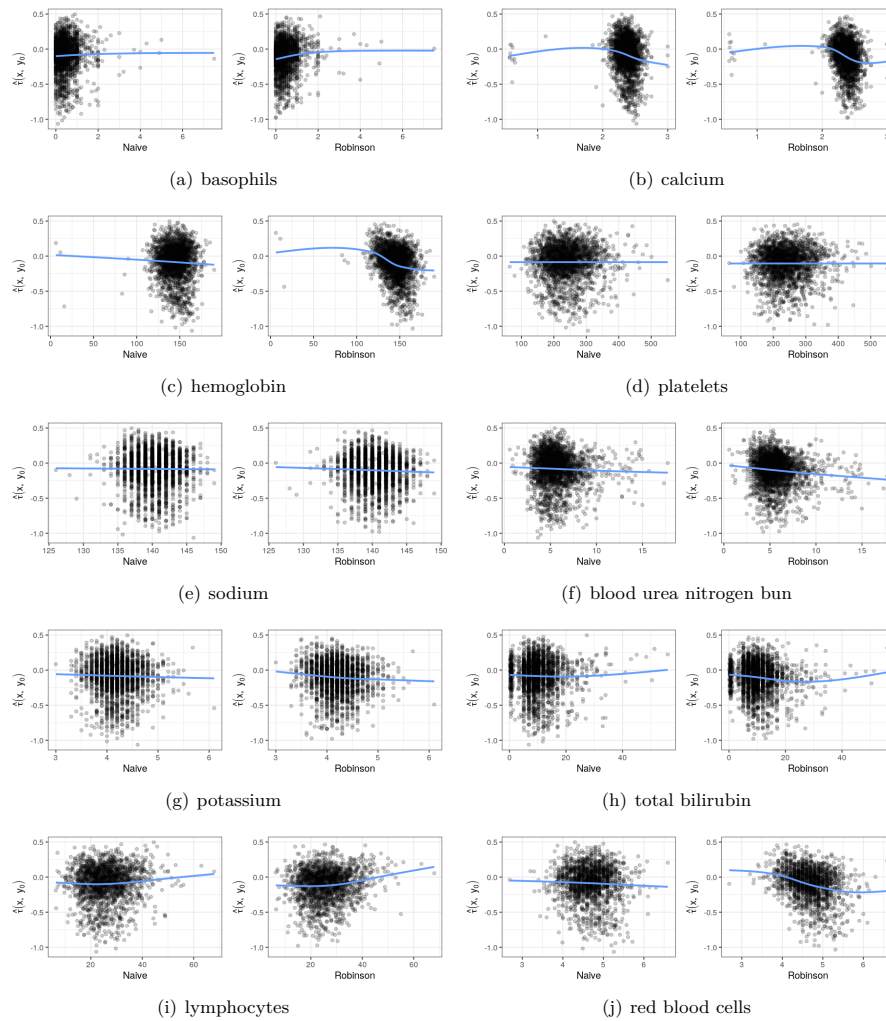


Figure S. 9: Handwriting ability score: dependency plot of individual average treatment effects calculated by model-based forest without (left) and with Robinson centering (right). Blue lines and diamond points depict (smooth conditional) mean effects.

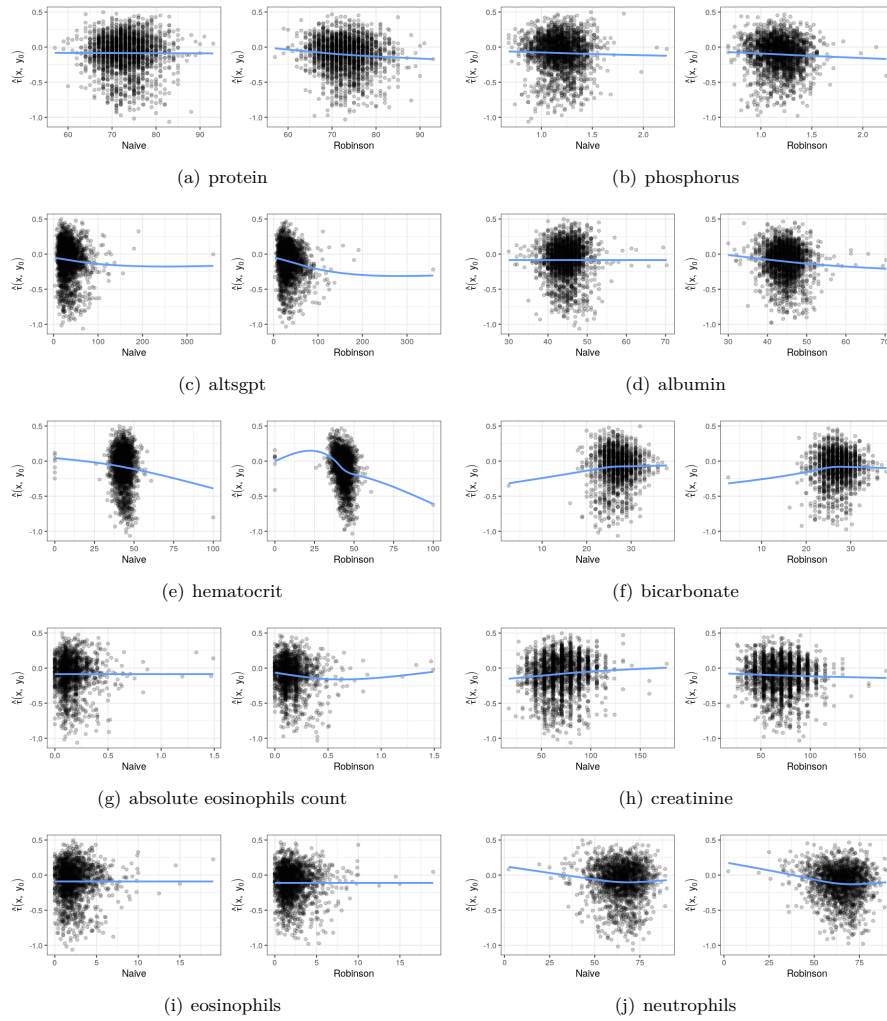


Figure S. 10: Handwriting ability score: dependency plot of individual average treatment effects calculated by model-based forest without (left) and with Robinson centering (right). Blue lines and diamond points depict (smooth conditional) mean effects.

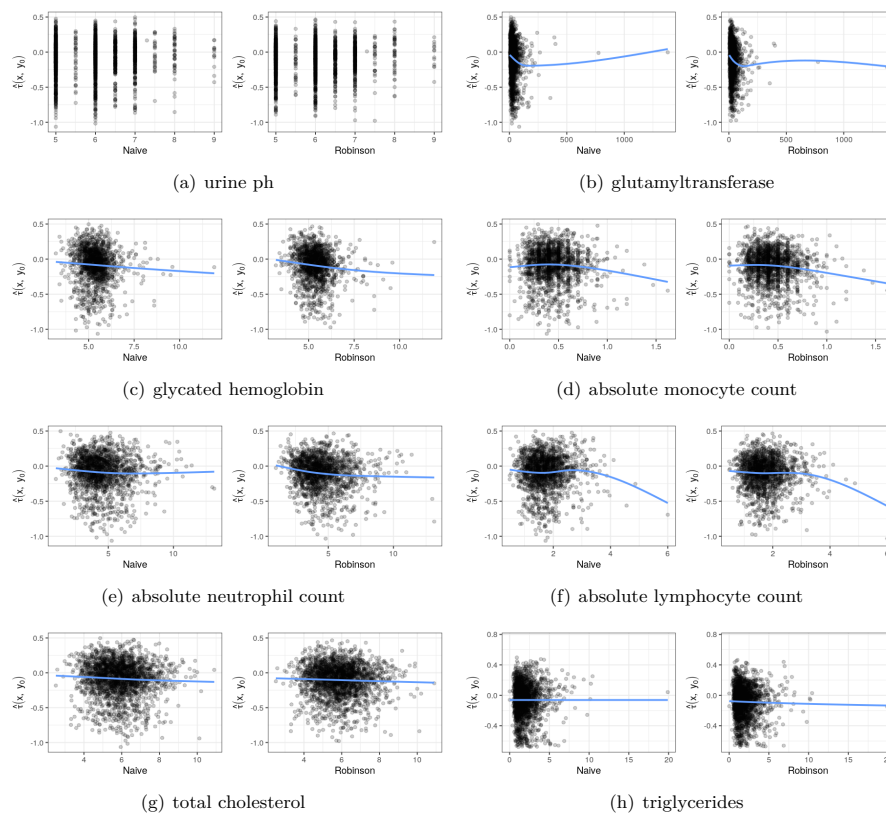


Figure S. 11: Handwriting ability score: dependency plot of individual average treatment effects calculated by model-based forest without (left) and with Robinson centering (right). Blue lines and diamond points depict (smooth conditional) mean effects.

Affiliation:

Susanne Dandl, Andreas Bender
Institut für Statistik, LMU München, Germany
Munich Center for Machine Learning (MCML), Germany

Torsten Hothorn
Institut für Epidemiologie, Biostatistik und Prävention, Universität Zürich
Hirschengraben 84, CH-8001 Zürich, Switzerland
E-mail: Torsten.Hothorn@R-project.org

7 General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models

Contributing Article

Molnar C, König G, Herbinger J, Freiesleben T, Dandl S, Scholbeck CA, Casalicchio G, Grosse-Wentrup M, Bischl B (2022). “General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models.” In A Holzinger, R Goebel, R Fong, T Moon, KR Müller, W Samek (eds.), *xxAI - Beyond Explainable AI*, volume 13200 of *Lecture Notes in Artificial Intelligence*, pp. 39–68. Springer, Cham. doi: 10.1007/978-3-031-04083-2_4

Replication Code

The code for replicating the examples and experiments in this paper is available under https://github.com/slds-lmu/code_pitfalls_uml.

Declaration of Contributions







Susanne Dandl wrote the chapter “Ignoring Multiple Comparison Problem” and parts of the chapter “Ignoring the Rashomon Effect”. She also provided feedback to other chapters, proofread, and revised the paper.

Contributions of Co-authors

Christoph Molnar initiated and coordinated the project. Christoph Molnar, Gunnar König, Julia Herbinger, Timo Freiesleben, Christian A. Scholbeck, and Giuseppe Casalicchio authored at least one chapter. All co-authors provided valuable input, proofread, and revised the paper.



General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models

Christoph Molnar^{1,7} , Gunnar König^{1,4} , Julia Herbinger¹ ,
Timo Freiesleben^{2,3} , Susanne Dandl¹ , Christian A. Scholbeck¹ ,
Giuseppe Casalicchio¹ , Moritz Grosse-Wentrup^{4,5,6} , and Bernd Bischl¹ 

¹ Department of Statistics, LMU Munich, Munich, Germany
christoph.molnar.ai@gmail.com

² Munich Center for Mathematical Philosophy, LMU Munich, Munich, Germany

³ Graduate School of Systemic Neurosciences, LMU Munich, Munich, Germany

⁴ Research Group Neuroinformatics, Faculty for Computer Science,
University of Vienna, Vienna, Austria

⁵ Research Platform Data Science @ Uni Vienna, Vienna, Austria

⁶ Vienna Cognitive Science Hub, Vienna, Austria

⁷ Leibniz Institute for Prevention Research and Epidemiology - BIPS GmbH,
Bremen, Germany

Abstract. An increasing number of model-agnostic interpretation techniques for machine learning (ML) models such as partial dependence plots (PDP), permutation feature importance (PFI) and Shapley values provide insightful model interpretations, but can lead to wrong conclusions if applied incorrectly. We highlight many general pitfalls of ML model interpretation, such as using interpretation techniques in the wrong context, interpreting models that do not generalize well, ignoring feature dependencies, interactions, uncertainty estimates and issues in high-dimensional settings, or making unjustified causal interpretations, and illustrate them with examples. We focus on pitfalls for global methods that describe the average model behavior, but many pitfalls also apply to local methods that explain individual predictions. Our paper addresses ML practitioners by raising awareness of pitfalls and identifying solutions for correct model interpretation, but also addresses ML researchers by discussing open issues for further research.

Keywords: Interpretable machine learning · Explainable AI

This work is funded by the Bavarian State Ministry of Science and the Arts (coordinated by the Bavarian Research Institute for Digital Transformation (bidt)), by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A, by the German Research Foundation (DFG), Emmy Noether Grant 437611051, and by the Graduate School of Systemic Neurosciences (GSN) Munich. The authors of this work take full responsibilities for its content.

© The Author(s) 2022

A. Holzinger et al. (Eds.): xxAI 2020, LNAI 13200, pp. 39–68, 2022.

https://doi.org/10.1007/978-3-031-04083-2_4

1 Introduction

In recent years, both industry and academia have increasingly shifted away from parametric models, such as generalized linear models, and towards non-parametric and non-linear machine learning (ML) models such as random forests, gradient boosting, or neural networks. The major driving force behind this development has been a considerable outperformance of ML over traditional models on many prediction tasks [32]. In part, this is because most ML models handle interactions and non-linear effects automatically. While classical statistical models – such as generalized additive models (GAMs) – also support the inclusion of interactions and non-linear effects, they come with the increased cost of having to (manually) specify and evaluate these modeling options. The benefits of many ML models are partly offset by their lack of interpretability, which is of major importance in many applications. For certain model classes (e.g. linear models), feature effects or importance scores can be directly inferred from the learned parameters and the model structure. In contrast, it is more difficult to extract such information from complex non-linear ML models that, for instance, do not have intelligible parameters and are hence often considered black boxes. However, model-agnostic interpretation methods allow us to harness the predictive power of ML models while gaining insights into the black-box model. These interpretation methods are already applied in many different fields. Applications of interpretable machine learning (IML) include understanding pre-emption decision-making [124] with partial dependence plots [36], inferring behavior from smartphone usage [105, 106] with the help of permutation feature importance [107] and accumulated local effect plots [3], or understanding the relation between critical illness and health records [70] using Shapley additive explanations (SHAP) [78]. Given the widespread application of interpretable machine learning, it is crucial to highlight potential pitfalls, that, in the worst case, can produce incorrect conclusions.

This paper focuses on pitfalls for model-agnostic IML methods, i.e. methods that can be applied to any predictive model. Model-specific methods, in contrast, are tied to a certain model class (e.g. saliency maps [57] for gradient-based models, such as neural networks), and are mainly considered out-of-scope for this work. We focus on pitfalls for global interpretation methods, which describe the expected behavior of the entire model with respect to the whole data distribution. However, many of the pitfalls also apply to local explanation methods, which explain individual predictions or classifications. Global methods include the partial dependence plot (PDP) [36], partial importance (PI) [19], accumulated local effects (ALE) [3], or the permutation feature importance (PFI) [12, 19, 33]. Local methods include the individual conditional expectation (ICE) curves [38], individual conditional importance (ICI) [19], local interpretable model-agnostic explanations (LIME) [94], Shapley values [108] and SHapley Additive exPlanations (SHAP) [77, 78] or counterfactual explanations [26, 115]. Furthermore, we distinguish between feature effect and feature importance methods. A feature effect indicates the direction and magnitude of a change in predicted outcome due to changes in feature values. Effect methods include

		Local	Global
Feature	Effects	ICE LIME Counterfactuals Shapley Values SHAP	PDP ALE
	Importance	ICI	PI PFI SAGE

Fig. 1. Selection of popular model-agnostic interpretation techniques, classified as local or global, and as effect or importance methods.

Shapley values, SHAP, LIME, ICE, PDP, or ALE. Feature importance methods quantify the contribution of a feature to the model performance (e.g. via a loss function) or to the variance of the prediction function. Importance methods include the PFI, ICI, PI, or SAGE. See Fig. 1 for a visual summary.

The interpretation of ML models can have subtle pitfalls. Since many of the interpretation methods work by similar principles of manipulating data and “probing” the model [100], they also share many pitfalls. The sources of these pitfalls can be broadly divided into three categories: (1) application of an unsuitable ML model which does not reflect the underlying data generating process very well, (2) inherent limitations of the applied IML method, and (3) wrong application of an IML method. Typical pitfalls for (1) are bad model generalization or the unnecessary use of complex ML models. Applying an IML method in a wrong way (3) often results from the users’ lack of knowledge of the inherent limitations of the chosen IML method (2). For example, if feature dependencies and interactions are present, potential extrapolations might lead to misleading interpretations for perturbation-based IML methods (inherent limitation). In such cases, methods like PFI might be a wrong choice to quantify feature importance.

Table 1. Categorization of the pitfalls by source.

Sources of pitfall	Sections
Unsuitable ML model	3, 4
Limitation of IML method	5.1, 6.1, 6.2, 9.1, 9.2
Wrong application of IML method	2, 5.2, 5.3, 7, 8, 9.3, 10

Contributions: We uncover and review general pitfalls of model-agnostic interpretation techniques. The categorization of these pitfalls into different sources is provided in Table 1. Each section describes and illustrates a pitfall, reviews possible solutions for practitioners to circumvent the pitfall, and discusses open issues that require further research. The pitfalls are accompanied by illustrative

examples for which the code can be found in this repository: https://github.com/compstat-lmu/code_pitfalls_uml.git. In addition to reproducing our examples, we invite readers to use this code as a starting point for their own experiments and explorations.

Related Work: Rudin et al. [96] present principles for interpretability and discuss challenges for model interpretation with a focus on inherently interpretable models. Das et al. [27] survey methods for explainable AI and discuss challenges with a focus on saliency maps for neural networks. A general warning about using and explaining ML models for high stakes decisions has been brought forward by Rudin [95], in which the author argues against model-agnostic techniques in favor of inherently interpretable models. Krishnan [64] criticizes the general conceptual foundation of interpretability, but does not dispute the usefulness of available methods. Likewise, Lipton [73] criticizes interpretable ML for its lack of causal conclusions, trust, and insights, but the author does not discuss any pitfalls in detail. Specific pitfalls due to dependent features are discussed by Hooker [54] for PDPs and functional ANOVA as well as by Hooker and Mentch [55] for feature importance computations. Hall [47] discusses recommendations for the application of particular interpretation methods but does not address general pitfalls.

2 Assuming One-Fits-All Interpretability

Pitfall: Assuming that a single IML method fits in all interpretation contexts can lead to dangerous misinterpretation. IML methods condense the complexity of ML models into human-intelligible descriptions that only provide insight into specific aspects of the model and data. The vast number of interpretation methods make it difficult for practitioners to choose an interpretation method that can answer their question. Due to the wide range of goals that are pursued under the umbrella term “interpretability”, the methods differ in which aspects of the model and data they describe.

For example, there are several ways to quantify or rank the features according to their relevance. The relevance measured by PFI can be very different from the relevance measured by the SHAP importance. If a practitioner aims to gain insight into the relevance of a feature regarding the model’s generalization error, a loss-based method (on unseen test data) such as PFI should be used. If we aim to expose which features the model relies on for its prediction or classification – irrespective of whether they aid the model’s generalization performance – PFI on test data is misleading. In such scenarios, one should quantify the relevance of a feature regarding the model’s prediction (and not the model’s generalization error) using methods like the SHAP importance [76].

We illustrate the difference in Fig. 2. We simulated a data-generating process where the target is completely independent of all features. Hence, the features are just noise and should not contribute to the model’s generalization error. Consequently, the features are not considered relevant by PFI on test data.

However, the model mechanistically relies on a number of spuriously correlated features. This reliance is exposed by marginal global SHAP importance.

As the example demonstrates, it would be misleading to view the PFI computed on test data or global SHAP as one-fits-all feature importance techniques. Like any IML method, they can only provide insight into certain aspects of model and data.

Many pitfalls in this paper arise from situations where an IML method that was designed for one purpose is applied in an unsuitable context. For example, extrapolation (Sect. 5.1) can be problematic when we aim to study how the model behaves under realistic data but simultaneously can be the correct choice if we want to study the sensitivity to a feature outside the data distribution.

For some IML techniques – especially local methods – even the same method can provide very different explanations, depending on the choice of hyperparameters: For counterfactuals, explanation goals are encoded in their optimization metrics [26, 34] such as sparsity and data faithfulness; The scope and meaning of LIME explanations depend on the kernel width and the notion of complexity [8, 37].

Solution: The suitability of an IML method cannot be evaluated with respect to one-fits-all interpretability but must be motivated and assessed with respect to well-defined interpretation goals. Similarly, practitioners must tailor the choice of the IML method and its respective hyperparameters to the interpretation context. This implies that these goals need to be clearly stated in a detailed manner *before* any analysis – which is still often not the case.

Open Issues: Since IML methods themselves are subject to interpretation, practitioners must be informed about which conclusions can or cannot be drawn given different choices of IML technique. In general, there are three aspects to be considered: (a) an intuitively understandable and plausible algorithmic construction of the IML method to achieve an explanation; (b) a clear mathematical axiomatization of interpretation goals and properties, which are linked by proofs and theoretical considerations to IML methods, and properties of models and data characteristics; (c) a practical translation for practitioners of the axioms from (b) in terms of what an IML method provides and what not, ideally with implementable guidelines and diagnostic checks for violated assumptions to guarantee correct interpretations. While (a) is nearly always given for any published method, much work remains for (b) and (c).

3 Bad Model Generalization

Pitfall: Under- or overfitting models can result in misleading interpretations with respect to the true feature effects and importance scores, as the model does not match the underlying data-generating process well [39]. Formally, most IML methods are designed to interpret the model instead of drawing inferences about

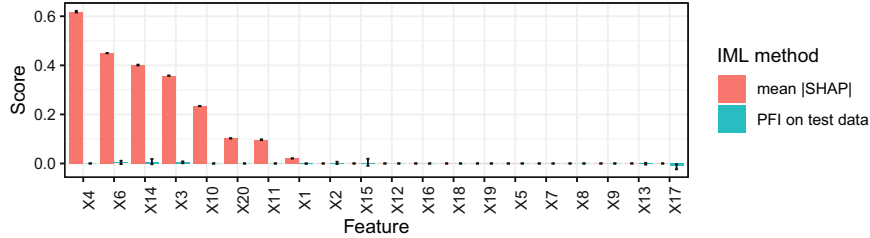


Fig. 2. Assuming one-fits-all interpretability. A default `xgboost` regression model that minimizes the mean squared error (MSE) was fitted on 20 independently and uniformly distributed features to predict another independent, uniformly sampled target. In this setting, predicting the (unconditional) mean $\mathbb{E}[Y]$ in a constant model is optimal. The learner overfits due to a small training data size. Mean marginal SHAP (red, error bars indicate 0.05 and 0.95 quantiles) exposes all mechanistically used features. In contrast, PFI on test data (blue, error bars indicate 0.05 and 0.95 quantiles) considers all features to be irrelevant, since no feature contributes to the generalization performance.

the data-generating process. In practice, however, the latter is often the goal of the analysis, and then an interpretation can only be as good as its underlying model. If a model approximates the data-generating process well enough, its interpretation should reveal insights into the underlying process.

Solution: In-sample evaluation (i.e. on training data) should not be used to assess the performance of ML models due to the risk of overfitting on the training data, which will lead to overly optimistic performance estimates. We must resort to out-of-sample validation based on resampling procedures such as hold-out for larger datasets or cross-validation, or even repeated cross-validation for small sample size scenarios. These resampling procedures are readily available in software [67, 89], and well-studied in theory as well as practice [4, 11, 104], although rigorous analysis of cross-validation is still considered an open problem [103]. Nested resampling is necessary, when computational model selection and hyperparameter tuning are involved [10]. This is important, as the Bayes error for most practical situations is unknown, and we cannot make absolute statements about whether a model already optimally fits the data.

Figure 3 shows the mean squared errors for a simulated example on both training and test data for a support vector machine (SVM), a random forest, and a linear model. Additionally, PDPs for all models are displayed, which show to what extent each model’s effect estimates deviate from the ground truth. The linear model is unable to represent the non-linear relationship, which is reflected in a high error on both test and training data and the linear PDPs. In contrast, the random forest has a low training error but a much higher test error, which indicates overfitting. Also, the PDPs for the random forest display overfitting behavior, as the curves are quite noisy, especially at the lower and upper value

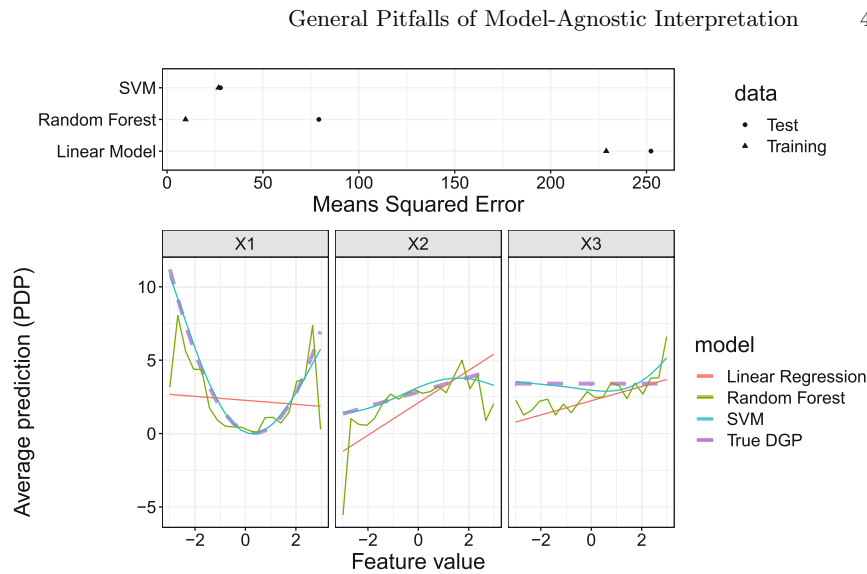


Fig. 3. Bad model generalization. **Top:** Performance estimates on training and test data for a linear regression model (underfitting), a random forest (overfitting) and a support vector machine with radial basis kernel (good fit). The three features are drawn from a uniform distribution, and the target was generated as $Y = X_1^2 + X_2 - 5X_1X_2 + \epsilon$, with $\epsilon \sim N(0, 5)$. **Bottom:** PDPs for the data-generating process (DGP) – which is the ground truth – and for the three models.

ranges of each feature. The SVM with both low training and test error comes closest to the true PDPs.

4 Unnecessary Use of Complex Models

Pitfall: A common mistake is to use an opaque, complex ML model when an interpretable model would have been sufficient, i.e. when the performance of interpretable models is only negligibly worse – or maybe the same or even better – than that of the ML model. Although model-agnostic methods can shed light on the behavior of complex ML models, inherently interpretable models still offer a higher degree of transparency [95] and considering them increases the chance of discovering the true data-generating function [23]. What constitutes an interpretable model is highly dependent on the situation and target audience, as even a linear model might be difficult to interpret when many features and interactions are involved.

It is commonly believed that complex ML models always outperform more interpretable models in terms of accuracy and should thus be preferred. However, there are several examples where interpretable models have proven to be serious competitors: More than 15 years ago, Hand [49] demonstrated that simple models often achieve more than 90% of the predictive power of potentially highly complex models across the UCI benchmark data repository and concluded that such

models often should be preferred due to their inherent interpretability; Makridakis et al. [79] systematically compared various ML models (including long-short-term-memory models and multi-layer neural networks) to statistical models (e.g. damped exponential smoothing and the Theta method) in time series forecasting tasks and found that the latter consistently show greater predictive accuracy; Kuhle et al. [65] found that random forests, gradient boosting and neural networks did not outperform logistic regression in predicting fetal growth abnormalities; Similarly, Wu et al. [120] have shown that a logistic regression model performs as well as AdaBoost and even better than an SVM in predicting heart disease from electronic health record data; Baesens et al. [7] showed that simple interpretable classifiers perform competitively for credit scoring, and in an update to the study the authors note that “the complexity and/or recency of a classifier are misleading indicators of its prediction performance” [71].

Solution: We recommend starting with simple, interpretable models such as linear regression models and decision trees. Generalized additive models (GAM) [50] can serve as a gradual transition between simple linear models and more complex machine learning models. GAMs have the desirable property that they can additively model smooth, non-linear effects and provide PDPs out-of-the-box, but without the potential pitfall of masking interactions (see Sect. 6). The additive model structure of a GAM is specified before fitting the model so that only the pre-specified feature or interaction effects are estimated. Interactions between features can be added manually or algorithmically (e.g. via a forward greedy search) [18]. GAMs can be fitted with component-wise boosting [99]. The boosting approach allows to smoothly increase model complexity, from sparse linear models to more complex GAMs with non-linear effects and interactions. This smooth transition provides insight into the tradeoffs between model simplicity and performance gains. Furthermore, component-wise boosting has an in-built feature selection mechanism as the model is build incrementally, which is especially useful in high-dimensional settings (see Sect. 9.1). The predictive performance of models of different complexity should be carefully measured and compared. Complex models should only be favored if the additional performance gain is both significant and relevant – a judgment call that the practitioner must ultimately make. Starting with simple models is considered best practice in data science, independent of the question of interpretability [23]. The comparison of predictive performance between model classes of different complexity can add further insights for interpretation.

Open Issues: Measures of model complexity allow quantifying the trade-off between complexity and performance and to automatically optimize for multiple objectives beyond performance. Some steps have been made towards quantifying model complexity, such as using functional decomposition and quantifying the complexity of the components [82] or measuring the stability of predictions [92]. However, further research is required, as there is no single perfect definition of interpretability, but rather multiple depending on the context [30, 95].

5 Ignoring Feature Dependence

5.1 Interpretation with Extrapolation

Pitfall: When features are dependent, perturbation-based IML methods such as PFI, PDP, LIME, and Shapley values extrapolate in areas where the model was trained with little or no training data, which can cause misleading interpretations [55]. This is especially true if the ML model relies on feature interactions [45] – which is often the case. Perturbations produce artificial data points that are used for model predictions, which in turn are aggregated to produce global or local interpretations [100]. Feature values can be perturbed by replacing original values with values from an equidistant grid of that feature, with permuted or randomly subsampled values [19], or with quantiles. We highlight two major issues: First, if features are dependent, all three perturbation approaches produce unrealistic data points, i.e. the new data points are located outside of the multivariate joint distribution of the data (see Fig. 4). Second, even if features are independent, using an equidistant grid can produce unrealistic values for the feature of interest. Consider a feature that follows a skewed distribution with outliers. An equidistant grid would generate many values between outliers and non-outliers. In contrast to the grid-based approach, the other two approaches maintain the marginal distribution of the feature of interest.

Both issues can result in misleading interpretations (illustrative examples are given in [55,84]), since the model is evaluated in areas of the feature space with few or no observed real data points, where model uncertainty can be expected to be very high. This issue is aggravated if interpretation methods integrate over such points with the same weight and confidence as for much more realistic samples with high model confidence.

Solution: Before applying interpretation methods, practitioners should check for dependencies between features in the data, e.g. via descriptive statistics or measures of dependence (see Sect. 5.2). When it is unavoidable to include dependent features in the model (which is usually the case in ML scenarios), additional information regarding the strength and shape of the dependence structure should be provided. Sometimes, alternative interpretation methods can be used as a workaround or to provide additional information. Accumulated local effect plots (ALE) [3] can be applied when features are dependent, but can produce non-intuitive effect plots for simple linear models with interactions [45]. For other methods such as the PFI, conditional variants exist [17,84,107]. In the case of LIME, it was suggested to focus in sampling on realistic (i.e. close to the data manifold) [97] and relevant areas (e.g. close to the decision boundary) [69]. Note, however, that conditional interpretations are often different and should not be used as a substitute for unconditional interpretations (see Sect. 5.3). Furthermore, dependent features should not be interpreted separately but rather jointly. This can be achieved by visualizing e.g. a 2-dimensional ALE plot of two dependent features, which, admittedly, only works for very low-dimensional combinations. Especially in high-dimensional settings where dependent features

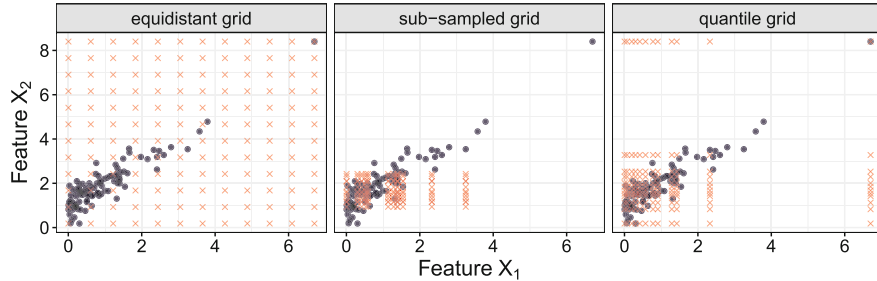


Fig. 4. Interpretation with extrapolation. Illustration of artificial data points generated by three different perturbation approaches. The black dots refer to observed data points and the red crosses to the artificial data points.

can be grouped in a meaningful way, grouped interpretation methods might be more reasonable (see Sect. 9.1).

We recommend using quantiles or randomly subsampled values over equidistant grids. By default, many implementations of interpretability methods use an equidistant grid to perturb feature values [41,81,89], although some also allow using user-defined values.

Open Issues: A comprehensive comparison of strategies addressing extrapolation and how they affect an interpretation method is currently missing. This also includes studying interpretation methods and their conditional variants when they are applied to data with different dependence structures.

5.2 Confusing Linear Correlation with General Dependence

Pitfall: Features with a Pearson correlation coefficient (PCC) close to zero can still be dependent and cause misleading model interpretations (see Fig. 5). While independence between two features implies that the PCC is zero, the converse is generally false. The PCC, which is often used to analyze dependence, only tracks linear correlations and has other shortcomings such as sensitivity to outliers [113]. Any type of dependence between features can have a strong impact on the interpretation of the results of IML methods (see Sect. 5.1). Thus, knowledge about the (possibly non-linear) dependencies between features is crucial for an informed use of IML methods.

Solution: Low-dimensional data can be visualized to detect dependence (e.g. scatter plots) [80]. For high-dimensional data, several other measures of dependence in addition to PCC can be used. If dependence is monotonic, Spearman’s rank correlation coefficient [72] can be a simple, robust alternative to PCC. For categorical or mixed features, separate dependence measures have been proposed, such as Kendall’s rank correlation coefficient for ordinal features, or the phi coefficient and Goodman & Kruskal’s lambda for nominal features [59].

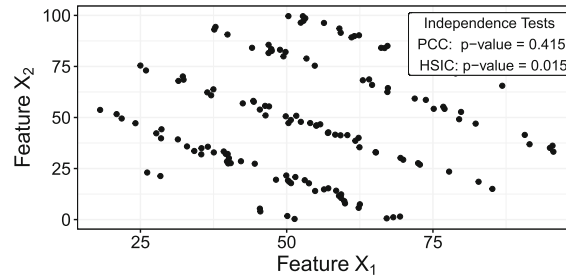


Fig. 5. Confusing linear correlation with dependence. Highly dependent features X_1 and X_2 that have a correlation close to zero. A test (H_0 : Features are independent) using Pearson correlation is not significant, but for HSIC, the H_0 -hypothesis gets rejected. Data from [80].

Studying non-linear dependencies is more difficult since a vast variety of possible associations have to be checked. Nevertheless, several non-linear association measures with sound statistical properties exist. Kernel-based measures, such as kernel canonical correlation analysis (KCCA) [6] or the Hilbert-Schmidt independence criterion (HSIC) [44], are commonly used. They have a solid theoretical foundation, are computationally feasible, and robust [113]. In addition, there are information-theoretical measures, such as (conditional) mutual information [24] or the maximal information coefficient (MIC) [93], that can however be difficult to estimate [9, 116]. Other important measures are e.g. the distance correlation [111], the randomized dependence coefficient (RDC) [74], or the alternating conditional expectations (ACE) algorithm [14]. In addition to using PCC, we recommend using at least one measure that detects non-linear dependencies (e.g. HSIC).

5.3 Misunderstanding Conditional Interpretation

Pitfall: Conditional variants of interpretation techniques avoid extrapolation but require a different interpretation. Interpretation methods that perturb features independently of others will extrapolate under dependent features but provide insight into the model’s mechanism [56, 61]. Therefore, these methods are said to be true to the model but not true to the data [21].

For feature effect methods such as the PDP, the plot can be interpreted as the isolated, average effect the feature has on the prediction. For the PFI, the importance can be interpreted as the drop in performance when the feature’s information is “destroyed” (by perturbing it). Marginal SHAP value functions [78] quantify a feature’s contribution to a specific prediction, and marginal SAGE value functions [25] quantify a feature’s contribution to the overall prediction performance. All the aforementioned methods extrapolate under dependent features (see also Sect. 5.1), but satisfy sensitivity, i.e. are zero if a feature is not used by the model [25, 56, 61, 110].

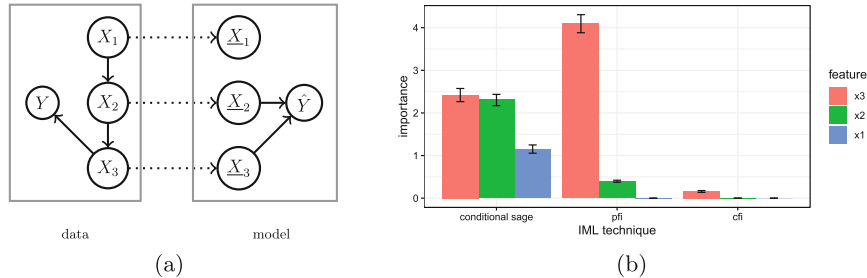


Fig. 6. Misunderstanding conditional interpretation. A linear model was fitted on the data-generating process modeled using a linear Gaussian structural causal model. The entailed directed acyclic graph is depicted on the left. For illustrative purposes, the original model coefficients were updated such that not only feature X_3 , but also feature X_2 is used by the model. PFI on test data considers both X_3 and X_2 to be relevant. In contrast, conditional feature importance variants either only consider X_3 to be relevant (CFI) or consider all features to be relevant (conditional SAGE value function).

Conditional variants of these interpretation methods do not replace feature values independently of other features, but in such a way that they conform to the conditional distribution. This changes the interpretation as the effects of all dependent features become entangled. Depending on the method, conditional sampling leads to a more or less restrictive notion of relevance.

For example, for dependent features, the Conditional Feature Importance (CFI) [17, 84, 107, 117] answers the question: “How much does the model performance drop if we permute a feature, *but given that we know the values of the other features?*” [63, 84, 107].¹ Two highly dependent features might be individually important (based on the unconditional PFI), but have a very low conditional importance score because the information of one feature is contained in the other and vice versa.

In contrast, the conditional variant of PDP, called marginal plot or M-plot [3], violates sensitivity, i.e. may even show an effect for features that are not used by the model. This is because for M-plots, the feature of interest is not sampled conditionally on the remaining features, but rather the remaining features are sampled conditionally on the feature of interest. As a consequence, the distribution of dependent covariates varies with the value of the feature of interest. Similarly, conditional SAGE and conditional SHAP value functions sample the remaining features conditional on the feature of interest and therefore violate sensitivity [25, 56, 61, 109].

We demonstrate the difference between PFI, CFI, and conditional SAGE value functions on a simulated example (Fig. 6) where the data-generating mech-

¹ While for CFI the conditional independence of the feature of interest X_j with the target Y given the remaining features X_{-j} ($Y \perp X_j | X_{-j}$) is already a sufficient condition for zero importance, the corresponding PFI may still be nonzero [63].

anism is known. While PFI only considers features to be relevant if they are actually used by the model, SAGE value functions may also consider a feature to be important that is not directly used by the model if it contains information that the model exploits. CFI only considers a feature to be relevant if it is both mechanistically used by the model and contributes unique information about Y .

Solution: When features are highly dependent and conditional effects and importance scores are used, the practitioner must be aware of the distinct interpretation. Recent work formalizes the implications of marginal and conditional interpretation techniques [21, 25, 56, 61, 63]. While marginal methods provide insight into the model’s mechanism but are not true to the data, their conditional variants are not true to the model but provide insight into the associations in the data.

If joint insight into model and data is required, designated methods must be used. ALE plots [3] provide interval-wise unconditional interpretations that are true to the data. They have been criticized to produce non-intuitive results for certain data-generating mechanisms [45]. Molnar et al. [84] propose a subgroup-based conditional sampling technique that allows for group-wise marginal interpretations that are true to model and data and that can be applied to feature importance and feature effects methods such as conditional PDPs and CFI. For feature importance, the DEDACT framework [61] allows to decompose conditional importance measures such as SAGE value functions into their marginal contributions and vice versa, thereby allowing global insight into both: the sources of prediction-relevant information in the data as well as into the feature pathways by which the information enters the model.

Open Issues: The quality of conditional IML techniques depends on the goodness of the conditional sampler. Especially in continuous, high-dimensional settings, conditional sampling is challenging. More research on the robustness of interpretation techniques regarding the quality of the sample is required.

6 Misleading Interpretations Due to Feature Interactions

6.1 Misleading Feature Effects Due to Aggregation

Pitfall: Global interpretation methods, such as PDP or ALE plots, visualize the average effect of a feature on a model’s prediction. However, they can produce misleading interpretations when features interact. Figure 7 A and B show the marginal effect of features X_1 and X_2 of the below-stated simulation example. While the PDP of the non-interacting feature X_1 seems to capture the true underlying effect of X_1 on the target quite well (A), the global aggregated effect of the interacting feature X_2 (B) shows almost no influence on the target, although an effect is clearly there by construction.

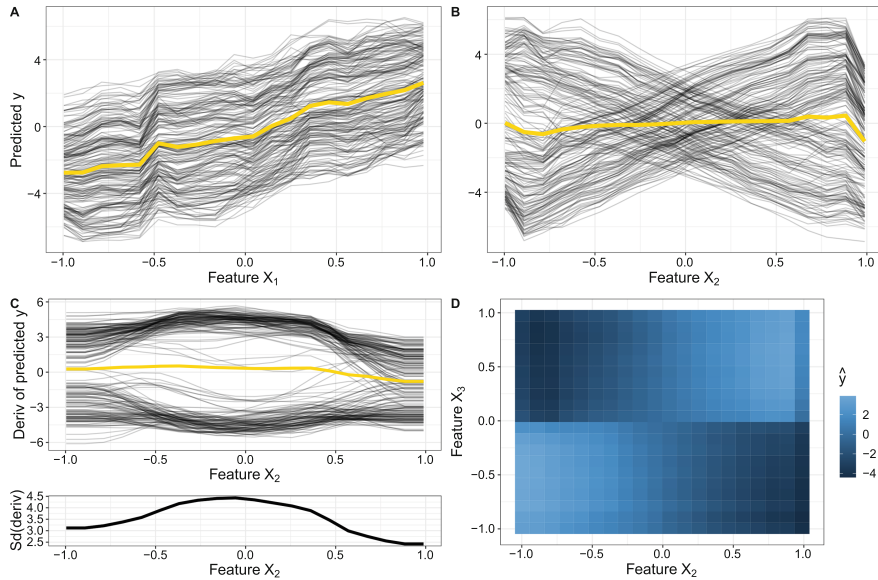


Fig. 7. Misleading effect due to interactions. Simulation example with interactions: $Y = 3X_1 - 6X_2 + 12X_2\mathbb{1}_{(X_3 \geq 0)} + \epsilon$ with $X_1, X_2, X_3 \stackrel{i.i.d.}{\sim} U[-1, 1]$ and $\epsilon \stackrel{i.i.d.}{\sim} N(0, 0.3)$. A random forest with 500 trees is fitted on 1000 observations. Effects are calculated on 200 randomly sampled (training) observations. **A, B:** PDP (yellow) and ICE curves of X_1 and X_2 ; **C:** Derivative ICE curves and their standard deviation of X_2 ; **D:** 2-dimensional PDP of X_2 and X_3 .

Solution: For the PDP, we recommend to additionally consider the corresponding ICE curves [38]. While PDP and ALE average out interaction effects, ICE curves directly show the heterogeneity between individual predictions. Figure 7 A illustrates that the individual marginal effect curves all follow an upward trend with only small variations. Hence, by aggregating these ICE curves to a global marginal effect curve such as the PDP, we do not lose much information. However, when the regarded feature interacts with other features, such as feature X_2 with feature X_3 in this example, then marginal effect curves of different observations might not show similar effects on the target. Hence, ICE curves become very heterogeneous, as shown in Fig. 7 B. In this case, the influence of feature X_2 is not well represented by the global average marginal effect. Particularly for continuous interactions where ICE curves start at different intercepts, we recommend the use of derivative or centered ICE curves, which eliminate differences in intercepts and leave only differences due to interactions [38]. Derivative ICE curves also point out the regions of highest interaction with other features. For example, Fig. 7 C indicates that predictions for X_2 taking values close to 0 strongly depend on other features' values. While these methods show that interactions are present with regards to the feature of interest but do not reveal other

features with which it interacts, the 2-dimensional PDP or ALE plot are options to visualize 2-way interaction effects. The 2-dimensional PDP in Fig. 7 D shows that predictions with regards to feature X_2 highly depend on the feature values of feature X_3 .

Other methods that aim to gain more insights into these visualizations are based on clustering homogeneous ICE curves, such as visual interaction effects (VINE) [16] or [122]. As an example, in Fig. 7 B, it would be more meaningful to average over the upward and downward proceeding ICE curves separately and hence show that the average influence of feature X_2 on the target depends on an interacting feature (here: X_3). Work by Zon et al. [125] followed a similar idea by proposing an interactive visualization tool to group Shapley values with regards to interacting features that need to be defined by the user.

Open Issues: The introduced visualization methods are not able to illustrate the type of the underlying interaction and most of them are also not applicable to higher-order interactions.

6.2 Failing to Separate Main from Interaction Effects

Pitfall: Many interpretation methods that quantify a feature's importance or effect cannot separate an interaction from main effects. The PFI, for example, includes both the importance of a feature and the importance of all its interactions with other features [19]. Also local explanation methods such as LIME and Shapley values only provide additive explanations without separation of main effects and interactions [40].

Solution: Functional ANOVA introduced by [53] is probably the most popular approach to decompose the joint distribution into main and interaction effects. Using the same idea, the H-Statistic [35] quantifies the interaction strength between two features or between one feature and all others by decomposing the 2-dimensional PDP into its univariate components. The H-Statistic is based on the fact that, in the case of non-interacting features, the 2-dimensional partial dependence function equals the sum of the two underlying univariate partial dependence functions. Another similar interaction score based on partial dependencies is defined by [42]. Instead of decomposing the partial dependence function, [87] uses the predictive performance to measure interaction strength. Based on Shapley values, Lundberg et al. [77] proposed SHAP interaction values, and Casalicchio et al. [19] proposed a fair attribution of the importance of interactions to the individual features.

Furthermore, Hooker [54] considers dependent features and decomposes the predictions in main and interaction effects. A way to identify higher-order interactions is shown in [53].

Open Issues: Most methods that quantify interactions are not able to identify higher-order interactions and interactions of dependent features. Furthermore,

the presented solutions usually lack automatic detection and ranking of all interactions of a model. Identifying a suitable shape or form of the modeled interaction is not straightforward as interactions can be very different and complex, e.g., they can be a simple product of features (multiplicative interaction) or can have a complex joint non-linear effect such as smooth spline surface.

7 Ignoring Model and Approximation Uncertainty

Pitfall: Many interpretation methods only provide a mean estimate but do not quantify uncertainty. Both the model training and the computation of interpretation are subject to uncertainty. The model is trained on (random) data, and therefore should be regarded as a random variable. Similarly, LIME’s surrogate model relies on perturbed and reweighted samples of the data to approximate the prediction function locally [94]. Other interpretation methods are often defined in terms of expectations over the data (PFI, PDP, Shapley values, ...), but are approximated using Monte Carlo integration. Ignoring uncertainty can result in the interpretation of noise and non-robust results. The true effect of a feature may be flat, but – purely by chance, especially on smaller datasets – the Shapley value might show an effect. This effect could cancel out once averaged over multiple model fits.

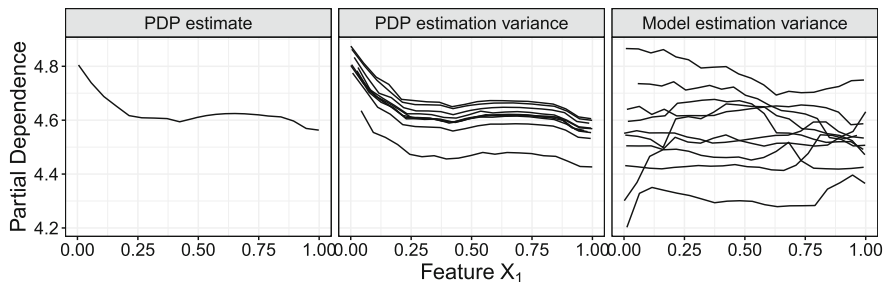


Fig. 8. Ignoring model and approximation uncertainty. PDP for X_1 with $Y = 0 \cdot X_1 + \sum_{j=2}^{10} X_j + \epsilon_i$ with $X_1, \dots, X_{10} \sim U[0, 1]$ and $\epsilon_i \sim N(0, 0.9)$. **Left:** PDP for X_1 of a random forest trained on 100 data points. **Middle:** Multiple PDPs (10x) for the model from left plots, but with different samples (each $n=100$) for PDP estimation. **Right:** Repeated (10x) data samples of $n=100$ and newly fitted random forest.

Figure 8 shows that a single PDP (first plot) can be misleading because it does not show the variance due to PDP estimation (second plot) and model fitting (third plot). If we are not interested in learning about a specific model, but rather about the relationship between feature X_1 and the target (in this case), we should consider the model variance.

Solution: By repeatedly computing PDP and PFI with a given model, but with different permutations or bootstrap samples, the uncertainty of the estimate can be quantified, for example in the form of confidence intervals. For PFI, frameworks for confidence intervals and hypothesis tests exist [2, 117], but they assume a fixed model. If the practitioner wants to condition the analysis on the modeling process and capture the process' variance instead of conditioning on a fixed model, PDP and PFI should be computed on multiple model fits [83].

Open Issues: While Moosbauer et al. [85] derived confidence bands for PDPs for probabilistic ML models that cover the model's uncertainty, a general model-agnostic uncertainty measure for feature effect methods such as ALE [3] and PDP [36] has (to the best of our knowledge) not been introduced yet.

8 Ignoring the Rashomon Effect

Pitfall: Sometimes different models explain the data-generating process equally well, but contradict each other. This phenomenon is called the Rashomon effect, named after the movie "Rashomon" from the year 1950. Breiman formalized it for predictive models in 2001 [13]: Different prediction models might perform equally well (Rashomon set), but construct the prediction function in a different way (e.g. relying on different features). This can result in conflicting interpretations and conclusions about the data. Even small differences in the training data can cause one model to be preferred over another.

For example, Dong and Rudin [29] identified a Rashomon set of equally well performing models for the COMPAS dataset. They showed that the models differed greatly in the importance they put on certain features. Specifically, if criminal history was identified as less important, race was more important and vice versa. Cherry-picking one model and its underlying explanation might not be sufficient to draw conclusions about the data-generating process. As Hancox-Li [48] states "just because race happens to be an unimportant variable in that one explanation does not mean that it is objectively an unimportant variable".

The Rashomon effect can also occur at the level of the interpretation method itself. Differing hyperparameters or interpretation goals can be one reason (see Sect. 2). But even if the hyperparameters are fixed, we could still obtain contradicting explanations by an interpretation method, e.g., due to a different data sample or initial seed.

A concrete example of the Rashomon effect is counterfactual explanations. Different counterfactuals may all alter the prediction in the desired way, but point to different feature changes required for that change. If a person is deemed uncreditworthy, one corresponding counterfactual explaining this decision may point to a scenario in which the person had asked for a shorter loan duration and amount, while another counterfactual may point to a scenario in which the person had a higher income and more stable job. Focusing on only one counterfactual explanation in such cases strongly limits the possible epistemic access.

Solution: If multiple, equally good models exist, their interpretations should be compared. Variable importance clouds [29] is a method for exploring variable importance scores for equally good models within one model class. If the interpretations are in conflict, conclusions must be drawn carefully. Domain experts or further constraints (e.g. fairness or sparsity) could help to pick a suitable model. Semenova et al. [102] also hypothesized that a large Rashomon set could contain simpler or more interpretable models, which should be preferred according to Sect. 4.

In the case of counterfactual explanations, multiple, equally good explanations exist. Here, methods that return a set of explanations rather than a single one should be used – for example, the method by Dandl et al. [26] or Mothilal et al. [86].

Open Issues: Numerous very different counterfactual explanations are overwhelming for users. Methods for aggregating or combining explanations are still a matter of future research.

9 Failure to Scale to High-Dimensional Settings

9.1 Human-Intelligibility of High-Dimensional IML Output

Pitfall: Applying IML methods naively to high-dimensional datasets (e.g. visualizing feature effects or computing importance scores on feature level) leads to an overwhelming and high-dimensional IML output, which impedes human analysis. Especially interpretation methods that are based on visualizations make it difficult for practitioners in high-dimensional settings to focus on the most important insights.

Solution: A natural approach is to reduce the dimensionality before applying any IML methods. Whether this facilitates understanding or not depends on the possible semantic interpretability of the resulting, reduced feature space – as features can either be selected or dimensionality can be reduced by linear or non-linear transformations. Assuming that users would like to interpret in the original feature space, many feature selection techniques can be used [46], resulting in much sparser and consequently easier to interpret models. Wrapper selection approaches are model-agnostic and algorithms like greedy forward selection or subset selection procedures [5, 60], which start from an empty model and iteratively add relevant (subsets of) features if needed, even allow to measure the relevance of features for predictive performance. An alternative is to directly use models that implicitly perform feature selection such as LASSO [112] or component-wise boosting [99] as they can produce sparse models with fewer features. In the case of LIME or other interpretation methods based on surrogate models, the aforementioned techniques could be applied to the surrogate model.

When features can be meaningfully grouped in a data-driven or knowledge-driven way [51], applying IML methods directly to grouped features instead of

single features is usually more time-efficient to compute and often leads to more appropriate interpretations. Examples where features can naturally be grouped include the grouping of sensor data [20], time-lagged features [75], or one-hot-encoded categorical features and interaction terms [43]. Before a model is fitted, groupings could already be exploited for dimensionality reduction, for example by selecting groups of features by the group LASSO [121].

For model interpretation, various papers extended feature importance methods from single features to groups of features [5, 43, 114, 119]. In the case of grouped PFI, this means that we perturb the entire group of features at once and measure the performance drop compared to the unperturbed dataset. Compared to standard PFI, the grouped PFI does not break the association to the other features of the group, but to features of other groups and the target. This is especially useful when features within the same group are highly correlated (e.g. time-lagged features), but between-group dependencies are rather low. Hence, this might also be a possible solution for the extrapolation pitfall described in Sect. 5.1.

We consider the PhoneStudy in [106] as an illustration. The PhoneStudy dataset contains 1821 features to analyze the link between human behavior based on smartphone data and participants' personalities. Interpreting the results in this use case seems to be challenging since features were dependent and single feature effects were either small or non-linear [106]. The features have been grouped in behavior-specific categories such as app-usage, music consumption, or overall phone usage. Au et al. [5] calculated various grouped importance scores on the feature groups to measure their influence on a specific personality trait (e.g. conscientiousness). Furthermore, the authors applied a greedy forward subset selection procedure via repeated subsampling on the feature groups and showed that combining app-usage features and overall phone usage features were most of the times sufficient for the given prediction task.

Open Issues: The quality of a grouping-based interpretation strongly depends on the human intelligibility and meaningfulness of the grouping. If the grouping structure is not naturally given, then data-driven methods can be used. However, if feature groups are not meaningful (e.g. if they cannot be described by a super-feature such as app-usage), then subsequent interpretations of these groups are purposeless. One solution could be to combine feature selection strategies with interpretation methods. For example, LIME's surrogate model could be a LASSO model. However, beyond surrogate models, the integration of feature selection strategies remains an open issue that requires further research.

Existing research on grouped interpretation methods mainly focused on quantifying grouped feature importance, but the question of "how a group of features influences a model's prediction" remains almost unanswered. Only recently, [5, 15, 101] attempted to answer this question by using dimension-reduction techniques (such as PCA) before applying the interpretation method. However, this is also a matter of further research.

9.2 Computational Effort

Pitfall: Some interpretation methods do not scale linearly with the number of features. For example, for the computation of exact Shapley values the number of possible coalitions [25,78], or for a (full) functional ANOVA decomposition the number of components (main effects plus all interactions) scales with $\mathcal{O}(2^p)$ [54].²

Solution: For the functional ANOVA, a common solution is to keep the analysis to the main effects and selected 2-way interactions (similar for PDP and ALE). Interesting 2-way interactions can be selected by another method such as the H-statistic [35]. However, the selection of 2-way interactions requires additional computational effort. Interaction strength usually decreases quickly with increasing interaction size, and one should only consider d -way interactions when all their $(d-1)$ -way interactions were significant [53]. For Shapley-based methods, an efficient approximation exists that is based on randomly sampling and evaluating feature orderings until the estimates converge. The variance of the estimates reduces in $\mathcal{O}(\frac{1}{m})$, where m is the number of evaluated orderings [25,78].

9.3 Ignoring Multiple Comparison Problem

Pitfall: Simultaneously testing the importance of multiple features will result in false-positive interpretations if the multiple comparisons problem (MCP) is ignored. The MCP is well known in significance tests for linear models and exists similarly in testing for feature importance in ML. For example, suppose we simultaneously test the importance of 50 features (with the H_0 -hypothesis of zero importance) at the significance level $\alpha = 0.05$. Even if all features are unimportant, the probability of observing that at least one feature is significantly important is $1 - \mathbb{P}(\text{'no feature important'}) = 1 - (1 - 0.05)^{50} \approx 0.923$. Multiple comparisons become even more problematic the higher the dimension of the dataset.

Solution: Methods such as Model-X knockoffs [17] directly control for the false discovery rate (FDR). For all other methods that provide p-values or confidence intervals, such as PIMP (Permutation IMPortance) [2], which is a testing approach for PFI, MCP is often ignored in practice to the best of our knowledge, with some exceptions [105,117]. One of the most popular MCP adjustment methods is the Bonferroni correction [31], which rejects a null hypothesis if its p-value is smaller than α/p , with p as the number of tests. It has the disadvantage that it increases the probability of false negatives [90]. Since MCP is well known in statistics, we refer the practitioner to [28] for an overview and discussion of alternative adjustment methods, such as the Bonferroni-Holm method [52].

² Similar to the PDP or ALE plots, the functional ANOVA components describe individual feature effects and interactions.

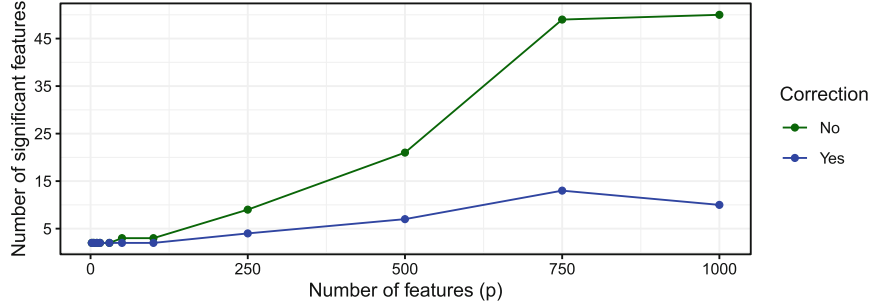


Fig. 9. Failure to scale to high-dimensional settings. Comparison of the number of features with significant importance - once with and once without Bonferroni-corrected significance levels for a varying number of added noise variables. Datasets were sampled from $Y = 2X_1 + 2X_2^2 + \epsilon$ with $X_1, X_2, \epsilon \sim N(0, 1)$. $X_3, X_4, \dots, X_p \sim N(0, 1)$ are additional noise variables with p ranging between 2 and 1000. For each p , we sampled two datasets from this data-generating process - one to train a random forest with 500 trees on and one to test whether feature importances differed from 0 using PIMP. In all experiments, X_1 and X_2 were correctly identified as important.

As an example, in Fig. 9 we compare the number of features with significant importance measured by PIMP once with and once without Bonferroni-adjusted significance levels ($\alpha = 0.05$ vs. $\alpha = 0.05/p$). Without correcting for multiple comparisons, the number of features mistakenly evaluated as important grows considerably with increasing dimension, whereas Bonferroni correction results in only a modest increase.

10 Unjustified Causal Interpretation

Pitfall: Practitioners are often interested in causal insights into the underlying data-generating mechanisms, which IML methods do not generally provide. Common causal questions include the identification of causes and effects, predicting the effects of interventions, and answering counterfactual questions [88]. For example, a medical researcher might want to identify risk factors or predict average and individual treatment effects [66]. In search of answers, a researcher can therefore be tempted to interpret the result of IML methods from a causal perspective.

However, a causal interpretation of predictive models is often not possible. Standard supervised ML models are not designed to model causal relationships but to merely exploit associations. A model may therefore rely on causes and effects of the target variable as well as on variables that help to reconstruct unobserved influences on Y , e.g. causes of effects [118]. Consequently, the question of whether a variable is relevant to a predictive model (indicated e.g. by $\text{PFI} > 0$) does not directly indicate whether a variable is a cause, an effect, or does not stand in any causal relation to the target variable. Furthermore,

even if a model would rely solely on direct causes for the prediction, the causal structure between features must be taken into account. Intervening on a variable in the real world may affect not only Y but also other variables in the feature set. Without assumptions about the underlying causal structure, IML methods cannot account for these adaptations and guide action [58,62].

As an example, we constructed a dataset by sampling from a structural causal model (SCM), for which the corresponding causal graph is depicted in Fig. 10. All relationships are linear Gaussian with variance 1 and coefficients 1. For a linear model fitted on the dataset, all features were considered to be relevant based on the model coefficients ($\hat{y} = 0.329x_1 + 0.323x_2 - 0.327x_3 + 0.342x_4 + 0.334x_5$, $R^2 = 0.943$), although x_3 , x_4 and x_5 do not cause Y .

Solution: The practitioner must carefully assess whether sufficient assumptions can be made about the underlying data-generating process, the learned model, and the interpretation technique. If these assumptions are met, a causal interpretation may be possible. The PDP between a feature and the target can be interpreted as the respective average causal effect if the model performs well and the set of remaining variables is a valid adjustment set [123]. When it is known whether a model is deployed in a causal or anti-causal setting – i.e. whether the model attempts to predict an effect from its causes or the other way round – a partial identification of the causal roles based on feature relevance is possible (under strong and non-testable assumptions) [118]. Designated tools and approaches are available for causal discovery and inference [91].

Open Issues: The challenge of causal discovery and inference remains an open key issue in the field of ML. Careful research is required to make explicit under which assumptions what insight about the underlying data-generating mechanism can be gained by interpreting an ML model.

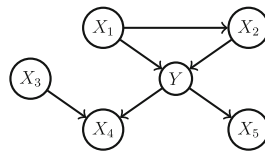


Fig. 10. Causal graph

11 Discussion

In this paper, we have reviewed numerous pitfalls of local and global model-agnostic interpretation techniques, e.g. in the case of bad model generalization, dependent features, interactions between features, or causal interpretations. We have not attempted to provide an exhaustive list of all potential pitfalls in ML

model interpretation, but have instead focused on common pitfalls that apply to various model-agnostic IML methods and pose a particularly high risk.

We have omitted pitfalls that are more specific to one IML method type: For local methods, the vague notions of neighborhood and distance can lead to misinterpretations [68,69], and common distance metrics (such as the Euclidean distance) are prone to the curse of dimensionality [1]; Surrogate methods such as LIME may not be entirely faithful to the original model they replace in interpretation. Moreover, we have not addressed pitfalls associated with certain data types (like the definition of superpixels in image data [98]), nor those related to human cognitive biases (e.g. the illusion of model understanding [22]).

Many pitfalls in the paper are strongly linked with axioms that encode desiderata of model interpretation. For example, pitfall Sect. 5.3 (misunderstanding conditional interpretations) is related to violations of sensitivity [56,110]. As such, axioms can help to make the strengths and limitations of methods explicit. Therefore, we encourage an axiomatic evaluation of interpretation methods.

We hope to promote a more cautious approach when interpreting ML models in practice, to point practitioners to already (partially) available solutions, and to stimulate further research on these issues. The stakes are high: ML algorithms are increasingly used for socially relevant decisions, and model interpretations play an important role in every empirical science. Therefore, we believe that users can benefit from concrete guidance on properties, dangers, and problems of IML techniques – especially as the field is advancing at high speed. We need to strive towards a recommended, well-understood set of tools, which will in turn require much more careful research. This especially concerns the meta-issues of comparisons of IML techniques, IML diagnostic tools to warn against misleading interpretations, and tools for analyzing multiple dependent or interacting features.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 420–434. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44503-X_27
2. Altmann, A., Tološi, L., Sander, O., Lengauer, T.: Permutation importance: a corrected feature importance measure. *Bioinformatics* **26**(10), 1340–1347 (2010). <https://doi.org/10.1093/bioinformatics/btq134>
3. Apley, D.W., Zhu, J.: Visualizing the effects of predictor variables in black box supervised learning models. *J. R. Stat. Soc.: Ser. B (Stat. Methodol.)* **82**(4), 1059–1086 (2020). <https://doi.org/10.1111/rssb.12377>
4. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. *Statist. Surv.* **4**, 40–79 (2010). <https://doi.org/10.1214/09-SS054>
5. Au, Q., Herbinger, J., Stachl, C., Bischl, B., Casalicchio, G.: Grouped feature importance and combined features effect plot. arXiv preprint [arXiv:2104.11688](https://arxiv.org/abs/2104.11688) (2021)
6. Bach, F.R., Jordan, M.I.: Kernel independent component analysis. *J. Mach. Learn. Res.* **3**(Jul), 1–48 (2002)

7. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. *J. Oper. Res. Soc.* **54**(6), 627–635 (2003). <https://doi.org/10.1057/palgrave.jors.2601545>
8. Bansal, N., Agarwal, C., Nguyen, A.: SAM: the sensitivity of attribution methods to hyperparameters. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8673–8683 (2020)
9. Belghazi, M.I., et al.: Mutual information neural estimation. In: *International Conference on Machine Learning*, pp. 531–540 (2018)
10. Bischl, B., et al.: Hyperparameter optimization: foundations, algorithms, best practices and open challenges. *arXiv preprint arXiv:2107.05847* (2021)
11. Bischl, B., Mersmann, O., Trautmann, H., Weihs, C.: Resampling methods for meta-model validation with recommendations for evolutionary computation. *Evol. Comput.* **20**(2), 249–275 (2012). <https://doi.org/10.1162/EVCO.a.00069>
12. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
13. Breiman, L.: Statistical modeling: the two cultures (with comments and a rejoinder by the author). *Stat. Sci.* **16**(3), 199–231 (2001). <https://doi.org/10.1214/ss/1009213726>
14. Breiman, L., Friedman, J.H.: Estimating optimal transformations for multiple regression and correlation. *J. Am. Stat. Assoc.* **80**(391), 580–598 (1985). <https://doi.org/10.1080/01621459.1985.10478157>
15. Brenning, A.: Transforming feature space to interpret machine learning models. *arXiv:2104.04295* (2021)
16. Britton, M.: Vine: visualizing statistical interactions in black box models. *arXiv preprint arXiv:1904.00561* (2019)
17. Candès, E., Fan, Y., Janson, L., Lv, J.: Panning for gold: ‘model-x’ knockoffs for high dimensional controlled variable selection. *J. R. Stat. Soc.: Ser. B (Stat. Methodol.)* **80**(3), 551–577 (2018). <https://doi.org/10.1111/rssb.12265>
18. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N.: Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1721–1730 (2015). <https://doi.org/10.1145/2783258.2788613>
19. Casalicchio, G., Molnar, C., Bischl, B.: Visualizing the feature importance for black box models. In: *Berlingiero, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G. (eds.) ECML PKDD 2018. LNCS (LNAI)*, vol. 11051, pp. 655–670. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10925-7_40
20. Chakraborty, D., Pal, N.R.: Selecting useful groups of features in a connectionist framework. *IEEE Trans. Neural Netw.* **19**(3), 381–396 (2008). <https://doi.org/10.1109/TNN.2007.910730>
21. Chen, H., Janizek, J.D., Lundberg, S., Lee, S.I.: True to the model or true to the data? *arXiv preprint arXiv:2006.16234* (2020)
22. Chromik, M., Eiband, M., Buchner, F., Krüger, A., Butz, A.: I think I get your point, AI! the illusion of explanatory depth in explainable AI. In: *26th International Conference on Intelligent User Interfaces, IUI 2021*, pp. 307–317. Association for Computing Machinery, New York (2021). <https://doi.org/10.1145/3397481.3450644>
23. Claeskens, G., Hjort, N.L., et al.: *Model Selection and Model Averaging*. Cambridge Books (2008). <https://doi.org/10.1017/CBO9780511790485>

24. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley (2012). <https://doi.org/10.1002/047174882X>
25. Covert, I., Lundberg, S.M., Lee, S.I.: Understanding global feature contributions with additive importance measures. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 17212–17223. Curran Associates, Inc. (2020)
26. Dandl, S., Molnar, C., Binder, M., Bischl, B.: Multi-objective counterfactual explanations. In: Bäck, T., et al. (eds.) PPSN 2020. LNCS, vol. 12269, pp. 448–469. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58112-1_31
27. Das, A., Rad, P.: Opportunities and challenges in explainable artificial intelligence (XAI): a survey. arXiv preprint [arXiv:2006.11371](https://arxiv.org/abs/2006.11371) (2020)
28. Dickhaus, T.: Simultaneous Statistical Inference. Springer, Heidelberg (2014). <https://doi.org/10.1007/978-3-642-45182-9>
29. Dong, J., Rudin, C.: Exploring the cloud of variable importance for the set of all good models. Nat. Mach. Intell. **2**(12), 810–824 (2020). <https://doi.org/10.1038/s42256-020-00264-0>
30. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint [arXiv:1702.08608](https://arxiv.org/abs/1702.08608) (2017)
31. Dunn, O.J.: Multiple comparisons among means. J. Am. Stat. Assoc. **56**(293), 52–64 (1961). <https://doi.org/10.1080/01621459.1961.10482090>
32. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems. J. Mach. Learn. Res. **15**(1), 3133–3181 (2014). <https://doi.org/10.5555/2627435.2697065>
33. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: learning a variable’s importance by studying an entire class of prediction models simultaneously. J. Mach. Learn. Res. **20**(177), 1–81 (2019)
34. Freiesleben, T.: Counterfactual explanations & adversarial examples-common grounds, essential differences, and potential transfers. arXiv preprint [arXiv:2009.05487](https://arxiv.org/abs/2009.05487) (2020)
35. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. Ann. Appl. Stat. **2**(3), 916–954 (2008). <https://doi.org/10.1214/07-AOAS148>
36. Friedman, J.H., et al.: Multivariate adaptive regression splines. Ann. Stat. **19**(1), 1–67 (1991). <https://doi.org/10.1214/aos/1176347963>
37. Garreau, D., von Luxburg, U.: Looking deeper into tabular lime. arXiv preprint [arXiv:2008.11092](https://arxiv.org/abs/2008.11092) (2020)
38. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. J. Comput. Graph. Stat. **24**(1), 44–65 (2015). <https://doi.org/10.1080/10618600.2014.907095>
39. Good, P.I., Hardin, J.W.: Common Errors in Statistics (and How to Avoid Them). Wiley (2012). <https://doi.org/10.1002/9781118360125>
40. Gosiewska, A., Biecek, P.: Do not trust additive explanations. arXiv preprint [arXiv:1903.11420](https://arxiv.org/abs/1903.11420) (2019)
41. Greenwell, B.M.: PDP: an R package for constructing partial dependence plots. R J. **9**(1), 421–436 (2017). <https://doi.org/10.32614/RJ-2017-016>
42. Greenwell, B.M., Boehmke, B.C., McCarthy, A.J.: A simple and effective model-based variable importance measure. [arXiv:1805.04755](https://arxiv.org/abs/1805.04755) (2018)
43. Gregorutti, B., Michel, B., Saint-Pierre, P.: Grouped variable importance with random forests and application to multiple functional data analysis. Comput. Stat. Data Anal. **90**, 15–35 (2015). <https://doi.org/10.1016/j.csda.2015.04.002>

44. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with Hilbert-Schmidt norms. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 63–77. Springer, Heidelberg (2005). https://doi.org/10.1007/11564089_7
45. Grömping, U.: Model-agnostic effects plots for interpreting machine learning models. Reports in Mathematics, Physics and Chemistry Report 1/2020 (2020)
46. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**(Mar), 1157–1182 (2003)
47. Hall, P.: On the art and science of machine learning explanations. arXiv preprint [arXiv:1810.02909](https://arxiv.org/abs/1810.02909) (2018)
48. Hancox-Li, L.: Robustness in machine learning explanations: does it matter? In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT* 2020, pp. 640–647. Association for Computing Machinery, New York (2020). <https://doi.org/10.1145/3351095.3372836>
49. Hand, D.J.: Classifier technology and the illusion of progress. *Stat. Sci.* **21**(1), 1–14 (2006). <https://doi.org/10.1214/088342306000000060>
50. Hastie, T., Tibshirani, R.: Generalized additive models. *Stat. Sci.* **1**(3), 297–310 (1986). <https://doi.org/10.1214/ss/1177013604>
51. He, Z., Yu, W.: Stable feature selection for biomarker discovery. *Comput. Biol. Chem.* **34**(4), 215–225 (2010). <https://doi.org/10.1016/j.compbiolchem.2010.07.002>
52. Holm, S.: A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **6**(2), 65–70 (1979)
53. Hooker, G.: Discovering additive structure in black box functions. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2004, pp. 575–580. Association for Computing Machinery, New York (2004). <https://doi.org/10.1145/1014052.1014122>
54. Hooker, G.: Generalized functional ANOVA diagnostics for high-dimensional functions of dependent variables. *J. Comput. Graph. Stat.* **16**(3), 709–732 (2007). <https://doi.org/10.1198/106186007X237892>
55. Hooker, G., Mentch, L.: Please stop permuting features: an explanation and alternatives. arXiv preprint [arXiv:1905.03151](https://arxiv.org/abs/1905.03151) (2019)
56. Janzing, D., Minorics, L., Blöbaum, P.: Feature relevance quantification in explainable AI: a causality problem. arXiv preprint [arXiv:1910.13413](https://arxiv.org/abs/1910.13413) (2019)
57. Kadir, T., Brady, M.: Saliency, scale and image description. *Int. J. Comput. Vis.* **45**(2), 83–105 (2001). <https://doi.org/10.1023/A:1012460413855>
58. Karimi, A.H., Schölkopf, B., Valera, I.: Algorithmic recourse: from counterfactual explanations to interventions. arXiv:2002.06278 (2020)
59. Khamis, H.: Measures of association: how to choose? *J. Diagn. Med. Sonography* **24**(3), 155–162 (2008). <https://doi.org/10.1177/8756479308317006>
60. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artif. Intell.* **97**(1–2), 273–324 (1997)
61. König, G., Freiesleben, T., Bischl, B., Casalicchio, G., Grosse-Wentrup, M.: Decomposition of global feature importance into direct and associative components (DEDACT). arXiv preprint [arXiv:2106.08086](https://arxiv.org/abs/2106.08086) (2021)
62. König, G., Freiesleben, T., Grosse-Wentrup, M.: A causal perspective on meaningful and robust algorithmic recourse. arXiv preprint [arXiv:2107.07853](https://arxiv.org/abs/2107.07853) (2021)
63. König, G., Molnar, C., Bischl, B., Grosse-Wentrup, M.: Relative feature importance. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 9318–9325. IEEE (2021). <https://doi.org/10.1109/ICPR48806.2021.9413090>

64. Krishnan, M.: Against interpretability: a critical examination of the interpretability problem in machine learning. *Philos. Technol.* **33**(3), 487–502 (2019). <https://doi.org/10.1007/s13347-019-00372-9>
65. Kuhle, S., et al.: Comparison of logistic regression with machine learning methods for the prediction of fetal growth abnormalities: a retrospective cohort study. *BMC Pregnancy Childbirth* **18**(1), 1–9 (2018). <https://doi.org/10.1186/s12884-018-1971-2>
66. König, G., Grosse-Wentrup, M.: A Causal Perspective on Challenges for AI in Precision Medicine (2019)
67. Lang, M., et al.: MLR3: a modern object-oriented machine learning framework in R. *J. Open Source Softw.* (2019). <https://doi.org/10.21105/joss.01903>
68. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: The dangers of post-hoc interpretability: unjustified counterfactual explanations. In: Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, pp. 2801–2807. International Joint Conferences on Artificial Intelligence Organization (2019)
69. Laugel, T., Renard, X., Lesot, M.J., Marsala, C., Detyniecki, M.: Defining locality for surrogates in post-hoc interpretability. arXiv preprint [arXiv:1806.07498](https://arxiv.org/abs/1806.07498) (2018)
70. Lauritsen, S.M., et al.: Explainable artificial intelligence model to predict acute critical illness from electronic health records. *Nat. Commun.* **11**(1), 1–11 (2020). <https://doi.org/10.1038/s41467-020-17431-x>
71. Lessmann, S., Baesens, B., Seow, H.V., Thomas, L.C.: Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. *Eur. J. Oper. Res.* **247**(1), 124–136 (2015). <https://doi.org/10.1016/j.ejor.2015.05.030>
72. Liebetrau, A.: Measures of Association. No. Bd. 32; Bd. 1983 in 07, SAGE Publications (1983)
73. Lipton, Z.C.: The mythos of model interpretability. *Queue* **16**(3), 31–57 (2018). <https://doi.org/10.1145/3236386.3241340>
74. Lopez-Paz, D., Hennig, P., Schölkopf, B.: The randomized dependence coefficient. In: *Advances in Neural Information Processing Systems*, pp. 1–9 (2013). <https://doi.org/10.5555/2999611.2999612>
75. Lozano, A.C., Abe, N., Liu, Y., Rosset, S.: Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics* **25**(12), i110–i118 (2009). <https://doi.org/10.1093/bioinformatics/btp199>
76. Lundberg, S.M., et al.: From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2**(1), 56–67 (2020). <https://doi.org/10.1038/s42256-019-0138-9>
77. Lundberg, S.M., Erion, G.G., Lee, S.I.: Consistent individualized feature attribution for tree ensembles. arXiv preprint [arXiv:1802.03888](https://arxiv.org/abs/1802.03888) (2018)
78. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *NIPS*, vol. 30, pp. 4765–4774. Curran Associates, Inc. (2017). <https://doi.org/10.5555/3295222.3295230>
79. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: concerns and ways forward. *PloS One* **13**(3) (2018). <https://doi.org/10.1371/journal.pone.0194889>
80. Matejka, J., Fitzmaurice, G.: Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 1290–1294 (2017). <https://doi.org/10.1145/3025453.3025912>

81. Molnar, C., Casalicchio, G., Bischl, B.: IML: an R package for interpretable machine learning. *J. Open Source Softw.* **3**(26), 786 (2018). <https://doi.org/10.21105/joss.00786>
82. Molnar, C., Casalicchio, G., Bischl, B.: Quantifying model complexity via functional decomposition for better post-hoc interpretability. In: Cellier, P., Driessens, K. (eds.) *ECML PKDD 2019. CCIS*, vol. 1167, pp. 193–204. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43823-4_17
83. Molnar, C., Freiesleben, T., König, G., Casalicchio, G., Wright, M.N., Bischl, B.: Relating the partial dependence plot and permutation feature importance to the data generating process. arXiv preprint [arXiv:2109.01433](https://arxiv.org/abs/2109.01433) (2021)
84. Molnar, C., König, G., Bischl, B., Casalicchio, G.: Model-agnostic feature importance and effects with dependent features—a conditional subgroup approach. arXiv preprint [arXiv:2006.04628](https://arxiv.org/abs/2006.04628) (2020)
85. Moosbauer, J., Herbinger, J., Casalicchio, G., Lindauer, M., Bischl, B.: Towards explaining hyperparameter optimization via partial dependence plots. In: *8th ICML Workshop on Automated Machine Learning (AutoML)* (2020)
86. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. *CoRR* abs/1905.07697 (2019). <http://arxiv.org/abs/1905.07697>
87. Oh, S.: Feature interaction in terms of prediction performance. *Appl. Sci.* **9**(23) (2019). <https://doi.org/10.3390/app9235191>
88. Pearl, J., Mackenzie, D.: *The Ladder of Causation. The Book of Why: The New Science of Cause and Effect*, pp. 23–52. Basic Books, New York (2018). <https://doi.org/10.1080/14697688.2019.1655928>
89. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011). <https://doi.org/10.5555/1953048.2078195>
90. Perneger, T.V.: What’s wrong with Bonferroni adjustments. *BMJ* **316**(7139), 1236–1238 (1998). <https://doi.org/10.1136/bmj.316.7139.1236>
91. Peters, J., Janzing, D., Scholkopf, B.: *Elements of Causal Inference - Foundations and Learning Algorithms*. The MIT Press (2017). <https://doi.org/10.5555/3202377>
92. Philipp, M., Rusch, T., Hornik, K., Strobl, C.: Measuring the stability of results from supervised statistical learning. *J. Comput. Graph. Stat.* **27**(4), 685–700 (2018). <https://doi.org/10.1080/10618600.2018.1473779>
93. Reshef, D.N., et al.: Detecting novel associations in large data sets. *Science* **334**(6062), 1518–1524 (2011). <https://doi.org/10.1126/science.1205438>
94. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you?: explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM (2016). <https://doi.org/10.1145/2939672.2939778>
95. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **1**(5), 206–215 (2019). <https://doi.org/10.1038/s42256-019-0048-x>
96. Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: fundamental principles and 10 grand challenges. arXiv preprint [arXiv:2103.11251](https://arxiv.org/abs/2103.11251) (2021)
97. Saito, S., Chua, E., Capel, N., Hu, R.: Improving lime robustness with smarter locality sampling. arXiv preprint [arXiv:2006.12302](https://arxiv.org/abs/2006.12302) (2020)
98. Schallner, L., Rabold, J., Scholz, O., Schmid, U.: Effect of superpixel aggregation on explanations in lime—a case study with biological data. arXiv preprint [arXiv:1910.07856](https://arxiv.org/abs/1910.07856) (2019)

99. Schmid, M., Hothorn, T.: Boosting additive models using component-wise p-splines. *Comput. Stat. Data Anal.* **53**(2), 298–311 (2008). <https://doi.org/10.1016/j.csda.2008.09.009>
100. Scholbeck, C.A., Molnar, C., Heumann, C., Bischl, B., Casalicchio, G.: Sampling, intervention, prediction, aggregation: a generalized framework for model-agnostic interpretations. In: Cellier, P., Driessens, K. (eds.) *ECML PKDD 2019. CCIS*, vol. 1167, pp. 205–216. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43823-4_18
101. Seedorff, N., Brown, G.: Totalvis: a principal components approach to visualizing total effects in black box models. *SN Comput. Sci.* **2**(3), 1–12 (2021). <https://doi.org/10.1007/s42979-021-00560-5>
102. Semenova, L., Rudin, C., Parr, R.: A study in Rashomon curves and volumes: a new perspective on generalization and model simplicity in machine learning. arXiv preprint [arXiv:1908.01755](https://arxiv.org/abs/1908.01755) (2021)
103. Shalev-Shwartz, S., Ben-David, S.: *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, Cambridge (2014)
104. Simon, R.: Resampling strategies for model assessment and selection. In: Dubitzky, W., Granzow, M., Berrar, D. (eds.) *Fundamentals of Data Mining in Genomics and Proteomics*, pp. 173–186. Springer, Cham (2007). https://doi.org/10.1007/978-0-387-47509-7_8
105. Stachl, C., et al.: Behavioral patterns in smartphone usage predict big five personality traits. *PsyArXiv* (2019). <https://doi.org/10.31234/osf.io/ks4vd>
106. Stachl, C., et al.: Predicting personality from patterns of behavior collected with smartphones. *Proc. Natl. Acad. Sci.* (2020). <https://doi.org/10.1073/pnas.1920484117>
107. Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A.: Conditional variable importance for random forests. *BMC Bioinform.* **9**(1), 307 (2008). <https://doi.org/10.1186/1471-2105-9-307>
108. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.* **41**(3), 647–665 (2013). <https://doi.org/10.1007/s10115-013-0679-x>
109. Sundararajan, M., Najmi, A.: The many Shapley values for model explanation. arXiv preprint [arXiv:1908.08474](https://arxiv.org/abs/1908.08474) (2019)
110. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: *International Conference on Machine Learning*, pp. 3319–3328. PMLR (2017)
111. Székely, G.J., Rizzo, M.L., Bakirov, N.K., et al.: Measuring and testing dependence by correlation of distances. *Ann. Stat.* **35**(6), 2769–2794 (2007). <https://doi.org/10.1214/009053607000000505>
112. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **58**(1), 267–288 (1996). <https://doi.org/10.1111/j.1467-9868.2011.00771.x>
113. Tjostheim, D., Otneim, H., Støve, B.: Statistical dependence: beyond pearson’s ρ . arXiv preprint [arXiv:1809.10455](https://arxiv.org/abs/1809.10455) (2018)
114. Valentin, S., Harkotte, M., Popov, T.: Interpreting neural decoding models using grouped model reliance. *PLoS Comput. Biol.* **16**(1), e1007148 (2020). <https://doi.org/10.1371/journal.pcbi.1007148>
115. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. JL Tech.* **31**, 841 (2017). <https://doi.org/10.2139/ssrn.3063289>

116. Walters-Williams, J., Li, Y.: Estimation of mutual information: a survey. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS (LNAI), vol. 5589, pp. 389–396. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02962-2_49
117. Watson, D.S., Wright, M.N.: Testing conditional independence in supervised learning algorithms. arXiv preprint [arXiv:1901.09917](https://arxiv.org/abs/1901.09917) (2019)
118. Weichwald, S., Meyer, T., Özdenizci, O., Schölkopf, B., Ball, T., Grosse-Wentrup, M.: Causal interpretation rules for encoding and decoding models in neuroimaging. *Neuroimage* **110**, 48–59 (2015). <https://doi.org/10.1016/j.neuroimage.2015.01.036>
119. Williamson, B.D., Gilbert, P.B., Simon, N.R., Carone, M.: A unified approach for inference on algorithm-agnostic variable importance. [arXiv:2004.03683](https://arxiv.org/abs/2004.03683) (2020)
120. Wu, J., Roy, J., Stewart, W.F.: Prediction modeling using EHR data: challenges, strategies, and a comparison of machine learning approaches. *Med. Care* S106–S113 (2010). <https://doi.org/10.1097/MLR.0b013e3181de9e17>
121. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc.: Ser. B (Statistical Methodology)* **68**(1), 49–67 (2006). <https://doi.org/10.1111/j.1467-9868.2005.00532.x>
122. Zhang, X., Wang, Y., Li, Z.: Interpreting the black box of supervised learning models: visualizing the impacts of features on prediction. *Appl. Intell.* **51**(10), 7151–7165 (2021). <https://doi.org/10.1007/s10489-021-02255-z>
123. Zhao, Q., Hastie, T.: Causal interpretations of black-box models. *J. Bus. Econ. Stat.* 1–10 (2019). <https://doi.org/10.1080/07350015.2019.1624293>
124. Zhao, X., Lovreglio, R., Nilsson, D.: Modelling and interpreting pre-evacuation decision-making using machine learning. *Autom. Constr.* **113**, 103140 (2020). <https://doi.org/10.1016/j.autcon.2020.103140>
125. van der Zon, S.B., Duivesteijn, W., van Ipenburg, W., Veldsink, J., Pechenizkiy, M.: ICIE 1.0: a novel tool for interactive contextual interaction explanations. In: Alzate, C., et al. (eds.) MIDAS/PAP -2018. LNCS (LNAI), vol. 11054, pp. 81–94. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-13463-1_6

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



8 Multi-Objective Counterfactual Explanations

Contributing Article

Dandl S, Molnar C, Binder M, Bischl B (2020). “Multi-Objective Counterfactual Explanations.” In T Bäck, M Preuss, A Deutz, H Wang, C Doerr, M Emmerich, H Trautmann (eds.), *Parallel Problem Solving from Nature – PPSN XVI*, pp. 448–469. Springer International Publishing, Cham. doi : 10.1007/978-3-030-58112-1_31

Replication Code

The results and experiments of this manuscript can be replicated using the code available at <https://github.com/dandls/moc>.

Declaration of Contributions

Susanne Dandl was responsible for implementing the method, for the benchmark and application in R. She executed the experiment and aggregated and visualized the results. She also wrote the majority of the manuscript.

Contributions of Co-authors

Christoph Molnar supervised and provided guidance throughout the whole process. He also implemented the benchmarks for DiCE and Recourse in Python and transferred the results to R. All co-authors provided input in the methodology and study design, and participated in proofreading and revising the paper.

Note

The publication builds upon the master thesis of Susanne Dandl, which was supervised by Christoph Molnar and Bernd Bischl. In the master thesis, the first version of the method was developed, including the basic implementation. For the publication, the master thesis was heavily extended in multiple directions:

- a fourth objective was added, which reflects a counterfactual’s adherence to the data manifold
- the returned counterfactual set was enriched by returning all nondominated counterfactuals found over the generations
- a novel mutator was proposed based on conditional density trees
- a method was proposed to reduce the size of the counterfactual set based on the hypervolume contribution

- the benchmark study comprised 10 instead of 7 data sets and two additional machine learning algorithms
- the method was compared against four additional methods
- three additional evaluation criteria were considered
- two additional visualization methods were proposed and implemented
- the code to generate counterfactuals with the proposed method was transferred into its own R package, the starting point for the contribution of Chapter 10



Multi-Objective Counterfactual Explanations

Susanne Dandl^(*), Christoph Molnar, Martin Binder, and Bernd Bischl

Department of Statistics, LMU Munich, Ludwigstr. 33, 80539 Munich, Germany
`susanne.dandl@stat.uni-muenchen.de`

Abstract. Counterfactual explanations are one of the most popular methods to make predictions of black box machine learning models interpretable by providing explanations in the form of ‘what-if scenarios’. Most current approaches optimize a collapsed, weighted sum of multiple objectives, which are naturally difficult to balance a-priori. We propose the Multi-Objective Counterfactuals (MOC) method, which translates the counterfactual search into a multi-objective optimization problem. Our approach not only returns a diverse set of counterfactuals with different trade-offs between the proposed objectives, but also maintains diversity in feature space. This enables a more detailed post-hoc analysis to facilitate better understanding and also more options for actionable user responses to change the predicted outcome. Our approach is also model-agnostic and works for numerical and categorical input features. We show the usefulness of MOC in concrete cases and compare our approach with state-of-the-art methods for counterfactual explanations.

Keywords: Interpretability · Interpretable machine learning · Counterfactual explanations · Multi-objective optimization · NSGA-II

1 Introduction

Interpretable machine learning methods have become very important in recent years to explain the behavior of black box machine learning (ML) models. A useful method for explaining *single* predictions of a model are counterfactual explanations. ML credit risk prediction is a common motivation for counterfactuals. For people whose credit applications have been rejected, it is valuable to know why they have not been accepted, either to understand the decision making process or to assess their actionable options to change the outcome. Counterfactuals provide these explanations in the form of “if these features had different values, your credit application would have been accepted”. For such explanations to be plausible, they should only suggest small changes in a few features.

This work has been partially supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A and by the Bavarian State Ministry of Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B). The authors of this work take full responsibility for its content.

© The Author(s) 2020
T. Bäck et al. (Eds.): PPSN 2020, LNCS 12269, pp. 448–469, 2020.
https://doi.org/10.1007/978-3-030-58112-1_31

Therefore, counterfactuals can be defined as close neighbors of an actual data point, but their predictions have to be sufficiently close to a (usually quite different) desired outcome. Counterfactuals explain why a certain outcome was not reached, can offer potential reasons to object against an unfair outcome and give guidance on how the desired prediction could be reached in the future [35]. Note that counterfactuals are also valuable for predictive modelers on a more technical level to investigate the pointwise robustness and the pointwise bias of their model.

2 Related Work

Counterfactuals are closely related to adversarial perturbations. These have the aim to deceive ML models instead of making the models interpretable [30]. Attribution methods such as Local Interpretable Model-agnostic Explanations (LIME) [27] and Shapley Values [22] explain a prediction by determining how much each feature contributed to it. Counterfactual explanations differ from feature attributions since they generate data points with a different, desired prediction instead of attributing a prediction to the features.

Counterfactual methods can be model-agnostic or model-specific. The latter usually exploit the internal structure of the underlying ML model, such as the trained weights of a neural network, while the former are based on general principles which work for arbitrary ML models - often by only assuming access to the prediction function of an already fitted model. Several model-agnostic counterfactual methods have been proposed [8, 11, 16, 18, 25, 29, 37]. Apart from Grath et al. [11], these approaches are limited to classification. Unlike the other methods, the method of Poyiadzi et al. [25] can obtain plausible counterfactuals by constructing feasible paths between data points with opposite predictions.

A model-specific approach was proposed by Wachter et al. [35], who also introduced and formalized the concept of counterfactuals in predictive modeling. Like many model-specific methods [15, 20, 24, 28, 33] their approach is limited to differentiable models. The approach of Tolomei et al. [32] generates explanations for tree-based ensemble binary classifiers. As with [35] and [20], it only returns a single counterfactual per run.

3 Contributions

In this paper, we introduce Multi-Objective Counterfactuals (MOC), which to the best of our knowledge is the first method to formalize the counterfactual search as a multi-objective optimization problem. We argue that the mathematical problem behind the search for counterfactuals should be naturally addressed as multi-objective. Most of the above methods optimize a collapsed, weighted sum of multiple objectives to find counterfactuals, which are naturally difficult to balance a-priori. They carry the risk of arbitrarily reducing the solution set to a single candidate without the option to discuss inherent trade-offs – which

should be especially relevant for model interpretation that is by design very hard to precisely capture in a (single) mathematical formulation.

Compared to Wachter et al. [35], we use a distance metric for mixed feature spaces and two additional objectives: one that measures the number of feature changes to obtain sparse and therefore more interpretable counterfactuals, and one that measures the closeness to the nearest observed data points for more plausible counterfactuals. MOC returns a Pareto set of counterfactuals that represents different trade-offs between our proposed objectives, and which are constructed to be diverse in feature space. This seems preferable because changes to different features can lead to a desired counterfactual prediction¹ and it is more likely that some counterfactuals meet the (hidden) preferences of a user. A single counterfactual might even suggest a strategy that is interpretable but not actionable (e.g., ‘reduce your number of pregnancies’) or counterproductive in more general contexts (e.g., ‘increase your age to reduce the risk of diabetes’). In addition, if multiple otherwise quite different counterfactuals suggest changes to the same feature, the user may have more confidence that the feature is an important lever to achieve the desired outcome. We refer the reader to Appendix A for two concrete examples illustrating the above.

Compared to other counterfactual methods, MOC is model-agnostic and handles classification, regression and mixed feature spaces, which furthermore increases its practical usefulness in general applications. Together with [16], our paper also includes one of the first benchmark studies that compares multiple counterfactual methods on multiple, heterogeneous datasets.

4 Methodology

[35] loosely define counterfactuals as:

“You were denied a loan because your annual income was £30,000. If your income had been £45,000, you would have been offered a loan. Here the statement of decision is followed by a counterfactual, or statement of how the world would have to be different for a desirable outcome to occur. Multiple counterfactuals are possible, as multiple desirable outcomes can exist, and there may be several ways to achieve any of these outcomes.”

We now formalize this statement by stating four objectives, which a counterfactual should adhere to. In the subsequent section we provide detailed definitions of these objectives and tie them together as a multi-objective optimization problem in order to generate a diverse set of different trade-off solutions.

4.1 Multi-Objective Counterfactuals

Definition 1 (Counterfactual Explanation). *Let $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$ be a prediction function, \mathcal{X} the feature space and $Y' \subset \mathbb{R}$ a set of desired outcomes. The latter*

¹ Rashomon effect [5].

can either be a single value or an interval of values. We define a counterfactual explanation \mathbf{x}' for an observation \mathbf{x}^* as a data point fulfilling the following: (1) its prediction $f(\mathbf{x}')$ is close to the desired outcome set Y' , (2) it is close to \mathbf{x}^* in the \mathcal{X} space, (3) it differs from \mathbf{x}^* only in a few features, and (4) it is a plausible data point according to the probability distribution $\mathbb{P}_{\mathcal{X}}$. For classification models, we assume that \hat{f} returns the probability for a user-selected class and Y' has to be the desired probability (range).

This can be translated into a multi-objective minimization task:

$$\min_{\mathbf{x}} \mathbf{o}(\mathbf{x}) := \min_{\mathbf{x}} (o_1(\hat{f}(\mathbf{x}), Y'), o_2(\mathbf{x}, \mathbf{x}^*), o_3(\mathbf{x}, \mathbf{x}^*), o_4(\mathbf{x}, \mathbf{X}^{obs})), \quad (1)$$

with $\mathbf{o} : \mathcal{X} \rightarrow \mathbb{R}^4$ and \mathbf{X}^{obs} as the observed (i.e. training) data. The first component o_1 quantifies the distance between $\hat{f}(\mathbf{x})$ and Y' . We define it as:²

$$o_1(\hat{f}(\mathbf{x}), Y') = \begin{cases} 0 & \text{if } \hat{f}(\mathbf{x}) \in Y' \\ \inf_{y' \in Y'} |\hat{f}(\mathbf{x}) - y'| & \text{else} \end{cases}.$$

The second component o_2 quantifies the distance between \mathbf{x}^* and \mathbf{x} using the Gower distance to account for mixed features [10]:

$$o_2(\mathbf{x}, \mathbf{x}^*) = \frac{1}{p} \sum_{j=1}^p \delta_G(x_j, x_j^*) \in [0, 1]$$

with p being the number of features. The value of δ_G depends on the feature type:

$$\delta_G(x_j, x_j^*) = \begin{cases} \frac{1}{\widehat{R}_j} |x_j - x_j^*| & \text{if } x_j \text{ is numerical} \\ \mathbb{I}_{x_j \neq x_j^*} & \text{if } x_j \text{ is categorical} \end{cases}$$

with \widehat{R}_j as the value range of feature j , extracted from the observed dataset.

Since the Gower distance does not take into account how many features have been changed, we introduce objective o_3 , which counts the number of changed features using the L_0 norm:

$$o_3(\mathbf{x}, \mathbf{x}^*) = \|\mathbf{x} - \mathbf{x}^*\|_0 = \sum_{j=1}^p \mathbb{I}_{x_j \neq x_j^*}.$$

The fourth objective o_4 measures the weighted average Gower distance between \mathbf{x} and the k nearest observed data points $\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[k]} \in \mathbf{X}^{obs}$ as an empirical approximation of how likely \mathbf{x} originates from the distribution of \mathcal{X} :

$$o_4(\mathbf{x}, \mathbf{X}^{obs}) = \sum_{i=1}^k w^{[i]} \frac{1}{p} \sum_{j=1}^p \delta_G(x_j, x_j^{[i]}) \in [0, 1] \text{ where } \sum_{i=1}^k w^{[i]} = 1.$$

² We chose the L_1 norm over the L_2 norm for a natural interpretation. Its non-differentiability is negligible for evolutionary optimization.

Throughout this paper, we set k to 1. Further procedures to increase the plausibility of the counterfactuals are integrated into the optimization algorithm and are described in Sect. 4.3.

Balancing the four objectives is difficult since the objectives contradict each other. For example, minimizing the distance between counterfactual outcome and desired outcome Y' (o_1) becomes more difficult when we require counterfactual feature values close to \mathbf{x}^* (o_2 and o_3) and to the observed data (o_4).

4.2 Counterfactual Search

Our proposed method MOC uses the *Nondominated Sorting Genetic Algorithm II* (NSGA-II) [7] with modifications specific to the problem considered. First, unlike the original NSGA-II, it uses *mixed integer evolutionary strategies* (MIES) [19] to work with the mixed discrete and continuous search space. Furthermore, a different crowding distance sorting algorithm is used, and we propose some optional adjustments tailored to the counterfactual search in the upcoming section.

For MOC, each candidate is described by its feature vector (the ‘genes’) and the objective values of the candidates are evaluated by Eq. (1). Features of candidates are recombined and mutated with predefined probabilities – some of the control parameters of MOC. Numerical features are recombined by the simulated binary crossover recombinator [6], all other feature types by the uniform crossover recombinator [31]. Based on [19], numerical features are mutated by the scaled Gaussian mutator. Categorical features are altered by uniformly sampling from their admissible levels, while binary and logical features are simply flipped. After recombination and mutation, some feature values are randomly set to the values of \mathbf{x}^* with a given (low) probability – another control parameter – to prevent all features from deviating from \mathbf{x}^* .

Contrary to NSGA-II, the crowding distance is computed not only in the objective space \mathbb{R}^4 (L_1 norm) but also in the feature space \mathcal{X} (Gower distance), and the distances are summed up with equal weighting. As a result, candidates are more likely kept if they differ greatly from another candidate in their feature values although they are similar in the objective values. Diversity in \mathcal{X} is desired because the chances of obtaining counterfactuals that meet the (hidden) preferences of users are higher. This approach is based on Avila et al. [2].

MOC stops if either a predefined number of generations is reached (default) or the performance no longer improves for a given number of successive generations.

4.3 Further Modifications

Initialization. Naively, we could initialize a population by uniformly sampling some feature values from their full range of possible values, while randomly setting other features to the values of \mathbf{x}^* to induce sparsity. However, if a feature has a large influence on the prediction, it should be more likely that the counterfactual values differ from \mathbf{x}^* . The importance of a feature for an entire dataset can

be measured as the standard deviation of the partial dependence plot [12]. Analogously, we propose to measure the feature importance for a single prediction with the standard deviation of the Individual Conditional Expectation (ICE) curve of \mathbf{x}^* . ICE curves show for one observation and for one feature how the prediction changes when the feature is changed, while other features are fixed to the values of the considered observation [9]. The greater the standard deviation of the ICE curve, the higher we set the probability that the feature value is initialized with a different value than the one of \mathbf{x}^* . Therefore, the standard deviation σ_j^{ICE} of each feature x_j is transformed into probabilities within $[p_{min}, p_{max}] \cdot 100\%$:

$$P(\text{value differs}) = \frac{(\sigma_j^{ICE} - \min(\sigma^{ICE})) \cdot (p_{max} - p_{min})}{\max(\sigma^{ICE}) - \min(\sigma^{ICE})} + p_{min}$$

with $\sigma^{ICE} := (\sigma_1^{ICE}, \dots, \sigma_p^{ICE})$. p_{min} and p_{max} are control parameters with default values 0.01 and 0.99.

Actionability. To get more actionable counterfactuals, extreme values of numerical features outside a predefined range are capped to the upper or lower bound after recombination and mutation. The ranges can either be derived from the minimum and maximum values of the features in the observed dataset or users can define these ranges. In addition, users can identify non-actionable features such as the country of birth or gender. The values of these features are permanently set to the values of \mathbf{x}^* for all candidates within MOC.

Penalization. Furthermore, candidates whose predictions are further away from the target than a predefined distance $\epsilon \in \mathbb{R}$ can be penalized. After the candidates have been sorted into fronts F_1 to F_K using nondominated sorting, the candidate that violates the constraint least will be reassigned to front F_{K+1} , the candidate with the second smallest violation to F_{K+2} , and so on. The concept is based on Deb et al. [7]. Since the constraint violators are in the last fronts, they are less likely to be selected for the next generation.

Mutation. Since the aforementioned mutators do not take the data distribution into account and can potentially generate unlikely new candidates, we suggest a conditional mutator. It generates plausible feature values conditional on the values of the other features. For each input feature, we trained a transformation tree [14] on X^{obs} , which is then used to sample values from the conditional distribution. We mutate the feature in randomized order since a feature mutation now depends on the previous changes.

How our proposed strategies for initialization and mutation affect MOC is later examined in a benchmark study (Sects. 6 and 7).

4.4 Evaluation Metric

We use the popular hypervolume indicator (HV) [38] to evaluate the quality of our estimated Pareto front, with reference point $\mathbf{s} = (\inf_{y' \in Y'} |\hat{f}(\mathbf{x}^*) - y'|, 1, p, 1)$, representing the maximal values of the objectives. We compute the HV always over the complete archive of evaluated solutions.

4.5 Tuning of Parameters

We also use HV, when we tune MOC’s control parameters – population size, the probabilities for recombining and mutating a feature of a candidate – with iterated F-racing [21]. Furthermore, we let iterated F-racing decide whether our proposed strategies for initialization and mutation of Sect. 4.3 are preferable. Tuning is performed on six binary classification datasets from OpenML [34] – which were not used in the benchmark. A summary of the tuning setup and results can be found in Table 5 in Appendix B. Iterated F-racing found both our initialization and mutation strategy to be advantageous. The tuned parameters were used for the credit data application and the benchmark study.

5 Credit Data Application

This section demonstrates the usefulness of MOC to explain the prediction of credit risk using the German credit dataset [13]. The dataset has 522 complete observations and nine features containing credit and customer information. Categories with few case numbers were combined. The binary target indicates whether a customer has a ‘good’ or ‘bad’ credit risk. We chose the first observation of the dataset as \mathbf{x}^* with the following feature values:

Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose
22	Female	2	Own	Little	Moderate	5951	48	Radio/TV

We tuned a support vector machine (with radial-basis (RBF) kernel) on the remaining data with the same tuning setup as for the benchmark (Appendix C). To obtain a single numerical outcome, only the predicted probability for the class ‘good’ credit risk was returned. We obtained an accuracy of 0.64 for the model using two nested cross-validations (CV) (5-fold CV in outer and inner loop) and a predicted probability for ‘good’ credit risk of 0.41 for \mathbf{x}^* .

We set the desired outcome interval to $Y' = [0.5, 1]$, which indicates a change to a ‘good’ credit risk. We generated counterfactuals using MOC with the parameter setting selected by iterated F-racing. Candidates with a prediction below 0.5 were penalized.

A total of 136 counterfactuals were found by MOC. In the following, we focus upon the 82 of them with predictions within $[0.5, 1]$. Credit *duration* was changed

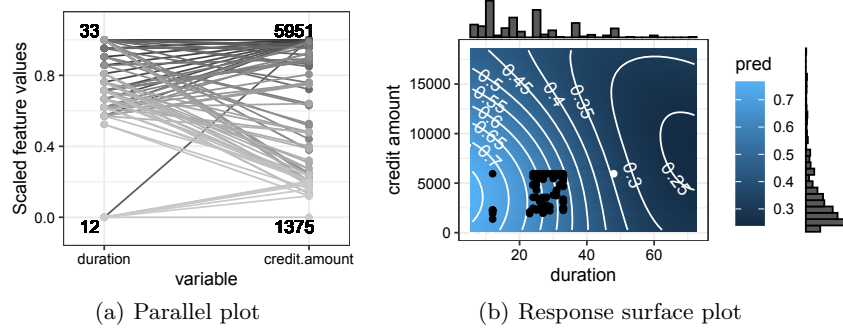


Fig. 1. Visualization of counterfactuals for the first data point \mathbf{x}^* of the credit dataset. (a) Feature values of the counterfactuals. Only changed features are shown. The given numbers indicate the minimum and maximum feature values of the counterfactuals. (b) Response surface plot for the model prediction along features duration and credit amount, holding other feature values constant at the value of \mathbf{x}^* . Colors and contour lines indicate the predicted value. The white point is \mathbf{x}^* and the black points are the counterfactuals that only proposed changes in duration and/or credit amount. The histograms show the marginal distributions of the features in the observed dataset.

for all counterfactuals, followed by *credit amount* (86%). Since a user might not want to investigate all returned counterfactuals individually (in feature space), we provide a visual summary of the Pareto set in Fig. 1, either as a parallel coordinate plot or a response surface plot³ along two features. All counterfactuals had values equal to or smaller than the values of \mathbf{x}^* for *duration* and *credit amount*. The response surface plot illustrates why these feature changes were recommended. The color gradient and contour lines indicate that either *duration* or both *credit amount* and *duration* must be decreased to reach the desired outcome. Due to the fourth objective and the conditional mutator, we obtained counterfactuals in high density areas (indicated by histograms). Counterfactuals in the lower left corner seem to be in a less favorable region far from \mathbf{x}^* , but they are close to the training data.

6 Experimental Setup

In this section, the performance of MOC is evaluated in a benchmark study for binary classification. The datasets are from the OpenML platform [34] and are briefly described in Table 1. We selected datasets with no missing values, with up to 3500 observations and a maximum of 40 features. We randomly selected ten observed data points per dataset as \mathbf{x}^* and excluded them from the training data. For each dataset, we tuned and trained the following models: logistic regression, random forest, xgboost, RBF support vector machine and a

³ This is equivalent to a 2-D ICE-curve through \mathbf{x}^* [9]. We refer to Sect. 4.3 for a general definition of ICE curves.

Table 1. Description of benchmark datasets. Legend: *task*: OpenML task id; *Obs*: Number of rows; *Cont/Cat*: Number of continuous/categorical features.

Task	Name	Obs	Cont	Cat
3718	boston	506	12	1
3846	cmc	1473	2	7
145976	diabetes	768	8	0
9971	ilpd	583	9	1
3913	kc2	522	21	0
3	kr-vs-kp	3196	0	36
3749	no2	500	7	0
3918	pc1	1109	21	0
3778	plasma_retinol	315	10	3
145804	tic-tac-toe	958	0	9

Table 2. MOC’s coverage rate of methods to be compared per dataset averaged over all models. The number of nondominated counterfactuals for each method are given in parentheses. Higher values of coverage indicate that MOC dominates the other method. The * indicates that the binomial test with $H_0 : p < 0.5$ that a counterfactual is covered by MOC is significant at the 0.05 level.

	DiCE	Recourse	Tweaking
boston	1* (36)	0.92* (24)	0.9* (10)
cmc	1* (17)		0.75 (8)
diabetes	1* (64)	0.45 (40)	1 (3)
ilpd	1* (26)	1* (37)	0.83 (6)
kc2	1* (53)	0.31 (55)	1 (2)
kr-vs-kp	1* (8)		0.2 (10)
no2	1* (58)	0.5 (12)	0.9* (10)
pc1	1* (60)	0.66* (38)	
plasma_retinol	1* (7)		0.89* (9)
tic-tac-toe	1* (20)		0.75 (8)

one-hidden-layer neural network. The tuning parameter set and the performance using nested resampling are in Table 8 in Appendix C. Each model returned only the probability for one class. The desired target for each \mathbf{x}^* was set to the opposite of the predicted class:

$$Y' = \begin{cases}]0.5, 1] & \text{if } \hat{f}(\mathbf{x}^*) \leq 0.5 \\ [0, 0.5] & \text{else} \end{cases}.$$

The benchmark study aimed to answer two research questions:

- Q1) How does MOC perform compared to other state-of-the-art methods for counterfactuals?
 Q2) How do our proposed strategies for initialization and mutation of Sect. 4.3 influence the performance of MOC?

For the first one, we compared MOC – once with and once without our proposed strategies for initialization and mutation – with ‘DiCE’ by Mothilal et al. [24], ‘Recourse’ by Ustun et al. [33] and ‘Tweaking’ by Tolomei et al. [32]. We chose DiCE, Recourse and Tweaking because they are implemented in general open source code libraries.⁴ The methods are only applicable to certain models: DiCE can handle neural networks and logistic regressions, Recourse can handle logistic regressions and Tweaking can handle random forests. Since Recourse can only process binary and numerical features, we did not train logistic regression on cmc, tic-tac-toe, kr-vs-kp and plasma_retinol. As a baseline, we selected the

⁴ Most other counterfactual methods are implemented for specific examples, but cannot be easily used for other datasets.

closest observed data point to \mathbf{x}^* (according to the Gower distance) that has a prediction equal to our desired outcome. Since this approach is part of the *What-If Tool* [36], we call this approach ‘Whatif’.

The parameters of DiCE, Recourse and Tweaking were set to the default values recommended by the authors (Appendix D). To allow for a fair comparison, we initialized MOC with the parameters of iterated F-racing which were tuned on other binary classification datasets (Appendix B). While MOC can potentially return several hundreds of counterfactuals, the other methods are designed to either return one or a few. We have therefore limited the maximum number of counterfactuals to ten for all approaches.⁵ Tweaking and Whatif generated only one counterfactual by design. For MOC we reduced the number of counterfactuals by preferring the ones that achieved the target prediction Y' and/or the highest HV contribution.

For all methods, only nondominated counterfactuals were considered for the evaluation. Since we are interested in a diverse set of counterfactuals, we evaluate the methods based on the size of their counterfactual set, its objective values, and the coverage rate derived from the coverage indicator by Zitzler and Thiele [38]. The coverage rate is the relative frequency with which counterfactuals of a method are dominated by MOC’s counterfactuals for a certain model and \mathbf{x}^* . A counterfactual covers another counterfactual if it dominates it, and it does not cover the other if both have the same objective values or the other has lower values in at least one objective. A coverage rate of 1 implies that for each generated counterfactual of a method MOC generated at least one dominating counterfactual. We only computed the coverage rate over counterfactuals that met the desired target Y' .

To answer the second research question, we compared the dominated HV over the generations of MOC with and without our proposed strategies for initialization and mutation. As a baseline, we used a random search approach that has the same population size (20) and number of generations (175) as MOC. In each generation, some feature values were uniformly sampled from their set of possible values derived from the observed data and \mathbf{x}^* , while other features were set to the values of \mathbf{x}^* . The HV for one generation was computed over the newly generated candidates combined with the candidates of the previous generations.

7 Results

Q1) MOC vs. State-of-the-Art Counterfactual Methods

Table 2 shows the coverage rate of each method (to be compared) by the tuned MOC per dataset. Some fields are empty because Recourse could not process features with more than two classes and Tweaking never achieved the desired outcome for pc1. MOC’s counterfactuals dominated all counterfactuals of DiCE for all datasets. The same holds for Tweaking except for kr-vs-kp and tic-tac-toe because the counterfactuals of Tweaking had the same objective values as

⁵ Note that this artificially penalizes our approach in the benchmark comparison.

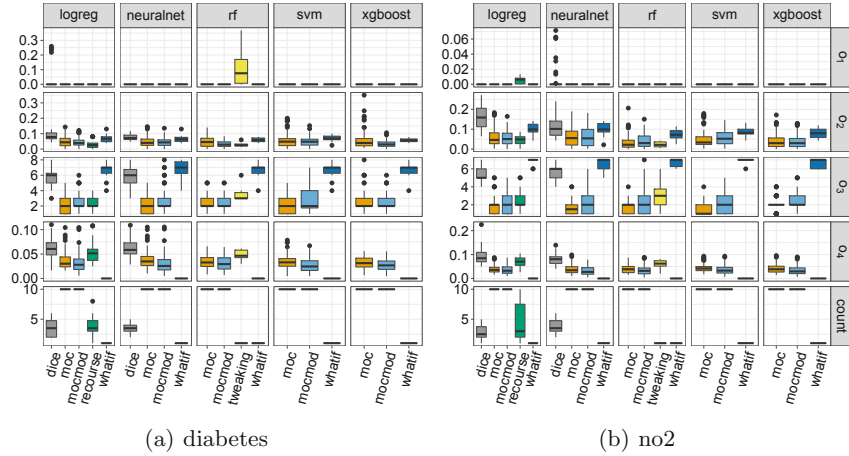


Fig. 2. Boxplots of the objective values and number of nondominated counterfactuals (*count*) per model for MOC with our proposed strategies for initialization and mutation (*mocmod*), MOC without these modifications, Whatif, DiCE, Recourse and Tweaking for the datasets diabetes and no2. Lower values are better except for *count*.

the ones of MOC. MOC's coverage rate of Recourse only exceeded 90% for boston and ilpd since Recourse's counterfactuals often deviated less from \mathbf{x}^* (but performed worse in other objectives).

Figure 2 compares MOC (with (*mocmod*) and without (*moc*) our proposed strategies for initialization and mutation) with the other methods for the datasets diabetes and no2 and for each model separately. The resulting boxplots for all other datasets are shown in Figs. 4 and 5 in the Appendix. They agree with the results shown here. Compared to the other methods, both versions of MOC found the most nondominated solutions, which met the target and changed the least features. DiCE performed worse than MOC in all objectives. Tweaking's counterfactuals were often closer to \mathbf{x}^* , but they were further away from the nearest training data point and more features were changed. Tweaking's counterfactuals often did not reach the desired outcome because they stayed too close to \mathbf{x}^* . The MOC with our proposed modifications found counterfactuals closer to \mathbf{x}^* and the observed data, but required more feature changes compared to MOC without the modifications.

Q2) MOC Strategies for Initialization and Mutation

Figure 3 shows the ranks of the dominated HVs for MOC without modifications, for each modification of MOC and random search. Ranks were calculated per dataset, model, \mathbf{x}^* and generation, and were averaged over all datasets, models and \mathbf{x}^* . We transformed HVs to ranks because the HVs are not comparable across \mathbf{x}^* . It can be seen that the MOC with our proposed modifications clearly

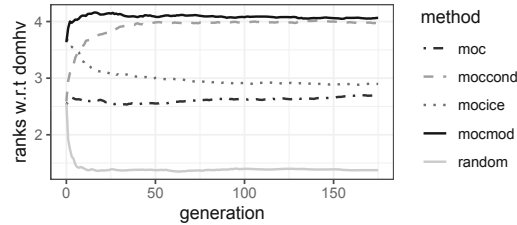


Fig. 3. Comparison of the ranks w.r.t. the dominated HV (*domhv*) per generation averaged over all models and datasets. For each approach, the population size of each generation was 20. A higher HV and therefore a higher rank is better. Legend: *moc*: MOC without our proposed modifications; *moccond*: MOC with the conditional mutator; *mocice*: MOC with the ICE curve variance initialization; *mocmod*: MOC with both modifications; *random*: random search.

outperforms the MOC without these modifications. The ranks of the initial population were higher when the ICE curve variance was used to initialize the candidates. The use of the conditional mutator led to higher dominated HVs over the generations. We received the best performance over the generations when both modifications were used. At each generation, all versions of MOC outperformed random search. Figure 6 in the Appendix shows the ranks over the generations for each dataset separately. They largely agree with the results shown here. The performance gains of MOC compared to random search were particularly evident for higher-dimensional datasets.

8 Conclusion and Outlook

In this paper, we introduced Multi-Objective Counterfactuals (MOC), which to the best of our knowledge is the first method to formalize the counterfactual search as a multi-objective optimization problem. Compared to state-of-the-art approaches, MOC returns a diverse set of counterfactuals with different trade-offs between our proposed objectives. Furthermore, MOC is model-agnostic and suited for classification, regression and mixed feature spaces. We demonstrated the usefulness of MOC to explain a prediction on the German credit dataset and showed in a benchmark study that MOC finds more counterfactuals than other counterfactual methods that are closer to the training data and required fewer feature changes. Our proposed initialization strategy (based on ICE curve variances) and our conditional mutator resulted in higher performance in fewer evaluations and in counterfactuals that were closer to the data point we were interested in and to the observed data.

MOC has only been evaluated on binary classification, and only with respect to the dominated HV and the individual objectives. It is an open question how to let users select the counterfactuals that meet their – a-priori unknown – trade-off between the objectives. We leave these investigations to future research.

9 Electronic Submission

The complete code of the algorithm and the code to reproduce the experiments and results of this paper are available at <https://github.com/susanne-207/moc>. The implementation of MOC is based on our implementation of [19], which we also used for [3]. We will provide an open source R library with our implementation of the method based on the `iml` package [23].

A Illustration of MOC’s Benefits

This section illustrates the benefits of having a *diverse set* of counterfactuals using the diabetes dataset of the benchmark study (Sect. 6). We will compare the counterfactuals returned by MOC with the ones of Recourse [33] and Tweaking [32]. Due to space constraints, we only show the six counterfactuals of MOC with the highest HV contribution for both examples.

Table 3. Counterfactuals and corresponding objective values of MOC and Recourse for the prediction of a logistic regression for observation 741 of the diabetes dataset. Shaded fields indicate values that differ from the value of observation 741 in brackets.

Feature (\mathbf{x}^*)	MOC ₁	MOC ₂	MOC ₃	MOC ₄	MOC ₅	MOC ₆	Recourse ₁	Recourse ₂	Recourse ₃
preg (11)	11.00	6.35	11.00	11.00	11.00	6.35	11.00	11.00	10.92
plas (120)	27.78	3.29	79.75	94.85	79.75	3.18	57.00	57.00	57.00
pres (80)	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00
skin (37)	37.00	37.00	37.00	37.00	37.00	37.00	37.00	36.81	37.00
insu (150)	150.00	150.00	17.13	150.00	40.61	150.00	150.00	150.00	150.00
mass (42.3)	42.30	42.30	29.17	15.36	29.17	42.30	42.30	42.30	42.30
pedi (0.78)	0.78	0.78	0.31	0.78	0.17	0.78	0.78	0.78	0.78
age (48)	48.00	41.61	44.42	48.00	48.00	48.00	28.36	28.36	28.36
o_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
o_2	0.06	0.12	0.10	0.07	0.10	0.11	0.08	0.08	0.08
o_3	1.00	3.00	5.00	2.00	4.00	2.00	2.00	3.00	3.00
o_4	0.10	0.05	0.03	0.07	0.04	0.07	0.09	0.09	0.09

Table 3 contrasts MOC’s counterfactuals with the three counterfactuals of Recourse for the prediction of observation 741. A logistic regression predicted a probability of having diabetes of 0.89 for this observation. The desired target is a prediction of less than 0.5, which indicates having no diabetes. All counterfactuals of Recourse suggest the same reduction in *age* and plasma concentration (*plas*), with two counterfactuals additionally suggesting a minimal reduction in the number of pregnancies (*preg*) or the skin fold thickness (*skin*).⁶ Apart from that a reduction in *age* or *preg* is impossible, they do not offer many options

⁶ By reclassifying *age* and *preg* as integers (instead of decimals), integer changes would be recommended by MOC, Recourse and Tweaking.

Table 4. Counterfactuals and corresponding objective values given by MOC and Tweaking for the prediction of a random forest for observation 268 of the cmc dataset. Shaded fields indicate values that differ from the value of observation 268 in brackets.

Feature (\mathbf{x}^*)	MOC ₁	MOC ₂	MOC ₃	MOC ₄	MOC ₅	MOC ₆	Tweaking ₁
preg (2)	2.00	2.00	2.00	2.00	2.00	2.00	1.53
plas (128)	121.50	90.21	126.83	128.00	88.44	120.64	119.71
pres (64)	64.00	64.00	64.00	64.00	64.00	64.00	64.00
skin (42)	42.00	42.00	42.00	42.00	42.00	42.00	42.00
insu (0)	0.00	0.00	0.00	0.00	0.00	90.93	0.00
mass (40)	40.00	40.00	40.00	40.00	40.00	40.00	40.00
pedi (1.1)	1.10	0.48	1.10	0.17	0.46	1.10	1.10
age (24)	24.00	24.00	24.00	24.00	25.85	24.00	28.29
o_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
o_2	0.00	0.06	0.00	0.05	0.06	0.02	0.02
o_3	1.00	2.00	1.00	1.00	3.00	2.00	3.00
o_4	0.05	0.02	0.05	0.04	0.01	0.03	0.06

for users. Instead, MOC returned a larger set of counterfactuals that provide more options for actionable user responses and are closer to the observed data than Recourse’s counterfactuals (o_4). Counterfactual MOC₁ has overall lower objective values than all counterfactuals of Recourse. MOC₃ suggested changes to five features so that it is especially close to the nearest training data point (o_4).

Table 4 compares the set of counterfactuals found by MOC with the single counterfactual found by Tweaking for the prediction of observation 268. A random forest classifier predicted a probability of having diabetes of 0.62 for this observation. Again, the desired target is a prediction of less than 0.5. Tweaking suggested reducing the number of children and plasma glucose concentration (*plas*) while increasing the *age* so that the probability of diabetes decreases. This is contradictory and not plausible. In contrast, MOC’s counterfactuals suggest various strategies, e.g., only a decrease of *plas*, which is easier to realize. In addition, MOC₁, MOC₃ and MOC₆ dominate the counterfactual of Tweaking. Since five of six counterfactuals suggest changes to *plas*, the user may have more confidence that *plas* is an important lever to achieve the desired outcome.

B Iterated F-racing

We used iterated F-racing (irace) [21] to tune the parameters of MOC for binary classification. The parameters and considered ranges are given in Table 5. The number of generations was not part of the parameter set because it would be always tuned to the upper bound. Instead, the number of generations was determined after the other parameters were tuned with irace. Irace was initialized with a maximum budget of 3000 evaluations equal to 3000 runs of MOC. In every step, irace randomly selected one of 300 instances. Each instance consisted of a trained model, a randomly selected data point from the observed data as \mathbf{x}^*

Table 5. Parameter space investigated with iterated F-racing, as well as the resulting optimized configuration (*Result*).

Name	Description	Range	Result
M	Population size	[20, 100]	20
initialization	Initialization strategy	[Random, ICE curve]	ICE curve
conditional	Whether to use the conditional mutator	[TRUE, FALSE]	TRUE
p.rec	Probability a pair of parents is chosen to recombine	[0.3, 1]	0.57
p.rec.gen	Probability a feature is recombined	[0.3, 1]	0.85
p.rec.use.orig	Probability the indicator for feature changes is recombined	[0.3, 1]	0.88
p.mut	Probability a child is chosen to be mutated	[0.05, 0.8]	0.79
p.mut.gen	Probability one feature is mutated	[0.05, 0.8]	0.56
p.mut.use.orig	Probability indicator for a feature change is flipped	[0.05, 0.5]	0.32

and a desired outcome. The desired target for each \mathbf{x}^* was the opposite of the predicted class:

$$Y' = \begin{cases}]0.5, 1] & \text{if } \hat{f}(\mathbf{x}^*) \leq 0.5 \\ [0, 0.5] & \text{else} \end{cases}.$$

The trained model was either logistic regression, random forest, xgboost, RBF support vector machine or a two-hidden-layer neural network. Each model estimated only the probability for one class. The models were trained on datasets obtained from the OpenML platform [34] (without the sampled \mathbf{x}^*) and are briefly described in Table 7. While these datasets were not used in the benchmark study (Sect. 6), the same preprocessing steps were conducted and the models were tuned with the same setup (see Sect. C for details).

In each step of irace, parameter configurations were evaluated by running MOC on the same selected instance. MOC stopped after evaluating 8000 candidates with Eq. (1), which should be enough to ensure convergence of the HV in most cases. The integral of the first order spline approximation of the dominated HV over the evaluations was the performance criterion as recommended by [26]. The integral takes into account not only the extent but also the rate of convergence of the dominated HV. A Friedman test was used to discard less promising configurations. The first Friedman test was conducted after initial configurations were evaluated on 15 instances; afterward, the test was conducted after evaluating the remaining configurations on a single instance to accelerate the exclusion process. The best configuration returned is given in Table 5.

To obtain a default parameter for the number of generations for the benchmark study, we determined for the 300 instances after how many generations of the tuned MOC the dominated HV has not increased for 10 generations. We chose the maximum of 175 generations as a default for the study.

Table 6. Tuning search space per model. The hyperparameters *ntrees* and *nrounds* were log-transformed.

Model	Hyperparameter	Range
randomforest	ntrees	[0, 1000]
xgboost	nrounds	[0, 1000]
svm	cost	[0.01, 1]
logreg	lr	[0.0005, 0.1]
neuralnet	lr	[0.0005, 0.1]
	layer_size	[1, 6]

Table 7. Description of datasets for tuning with iterated F-racing. Legend: *Task*: OpenML task id; *Obs*: Number of rows; *Cont/Cat*: Number of continuous/categorical features.

Task	Name	Obs	Cont	Cat
3818	tae	151	3	2
3917	kc1	2109	21	0
52945	breastTumor	277	0	6
3483	mammography	11183	6	0
3822	nursery	12960	0	8
3586	abalone	4177	7	1

C Model Hyperparameters for the Benchmark Study

We used random search (with 200 iterations for neural networks and 100 iterations for all other models) and 5-fold CV (with misclassification error as performance measure) to tune the hyperparameters of the models on the training data. The tuning search space was the same as for iterated F-racing and is shown in Table 6. Numerical features were scaled (standardization (Z-score) for random forest, min-max-scaling (0–1-range) for all other models) and categorical features were one-hot encoded. For neural network and logistic regression, ADAM [17] was the optimizer, the batch size was 32 with a 1/3 validation split and early stopping was conducted after 5 patience steps. Logistic regression needed these configurations because we constructed the model as a zero-hidden-layer neural network. For all other hyperparameters of the models, we chose the default values of the `mlr` [4] and `keras` [1] R packages. Table 8 shows the accuracies of the trained models using nested resampling (5-fold CV in outer and inner loop).

Table 8. Accuracy using nested resampling per benchmark dataset and model. Legend: *Name*: OpenML task name; *rf*: random forest. Logistic regression (*logreg*) was only trained on datasets with numerical or binary features.

Name	rf	xgboost	svm	logreg	neuralnet
boston	0.90	0.89	0.87	0.86	0.87
cmc	0.70	0.72	0.67		0.68
diabetes	0.76	0.74	0.75	0.63	0.68
ilpd	0.69	0.67	0.65	0.53	0.58
kc2	0.81	0.80	0.79	0.75	0.72
kr-vs-kp	0.99	0.99	0.97		0.99
no2	0.63	0.59	0.58	0.55	0.54
pc1	0.93	0.93	0.91	0.91	0.88
plasma_retinol	0.53	0.52	0.58		0.55
tic-tac-toe	0.99	0.99	0.98		0.97

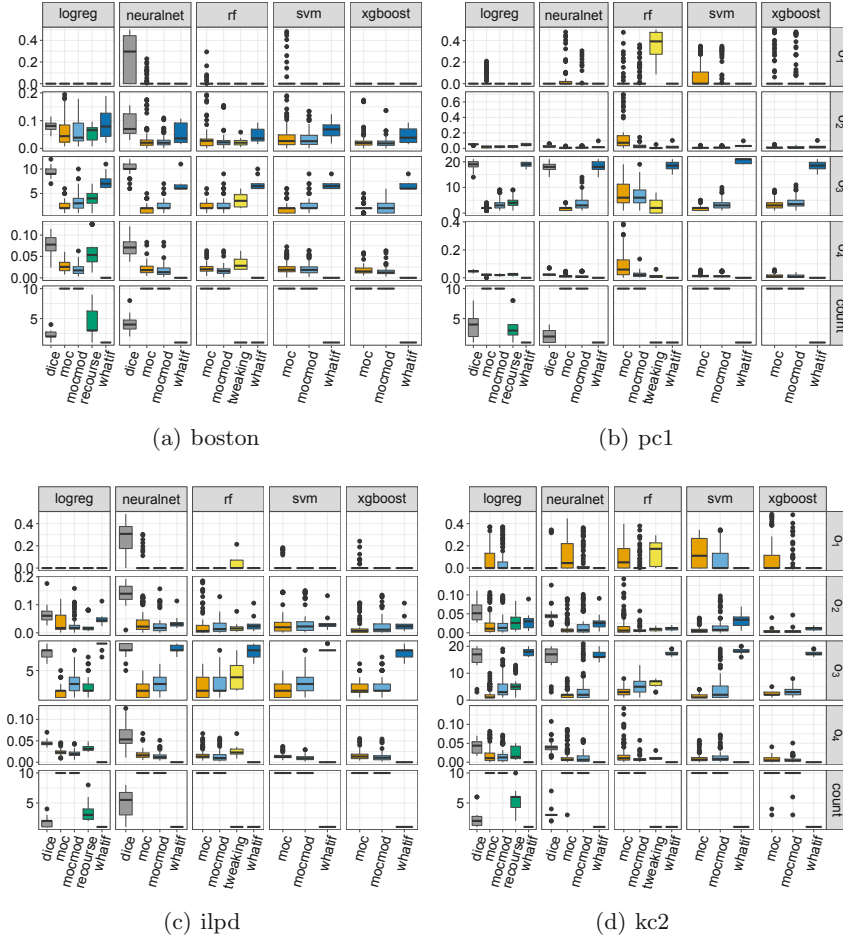


Fig. 4. Boxplots of the objective values and number of nondominated counterfactuals (*count*) per dataset and model for MOC with our proposed strategies for initialization and mutation (*mocmod*), MOC without these modifications, Whatif, DiCE, Recourse and Tweaking. Lower values are better except for *count*.

D Control Parameters of Counterfactual Methods

For Tweaking [32], we only changed ϵ , a positive threshold that limits the tweaking of each feature. It was set to 0.5 because it obtained better results for the authors on their data example on Ad Quality in comparison to the default value 0.1. We used the R implementation of Tweaking on Github: <https://github.com/katokohaku/featureTweakR> (commit 6f3e614). For Recourse [33], we left all parameters at their default settings. We used the Python implementation of Recourse on Github: <https://github.com/ustunb/actionable-recourse> (com-

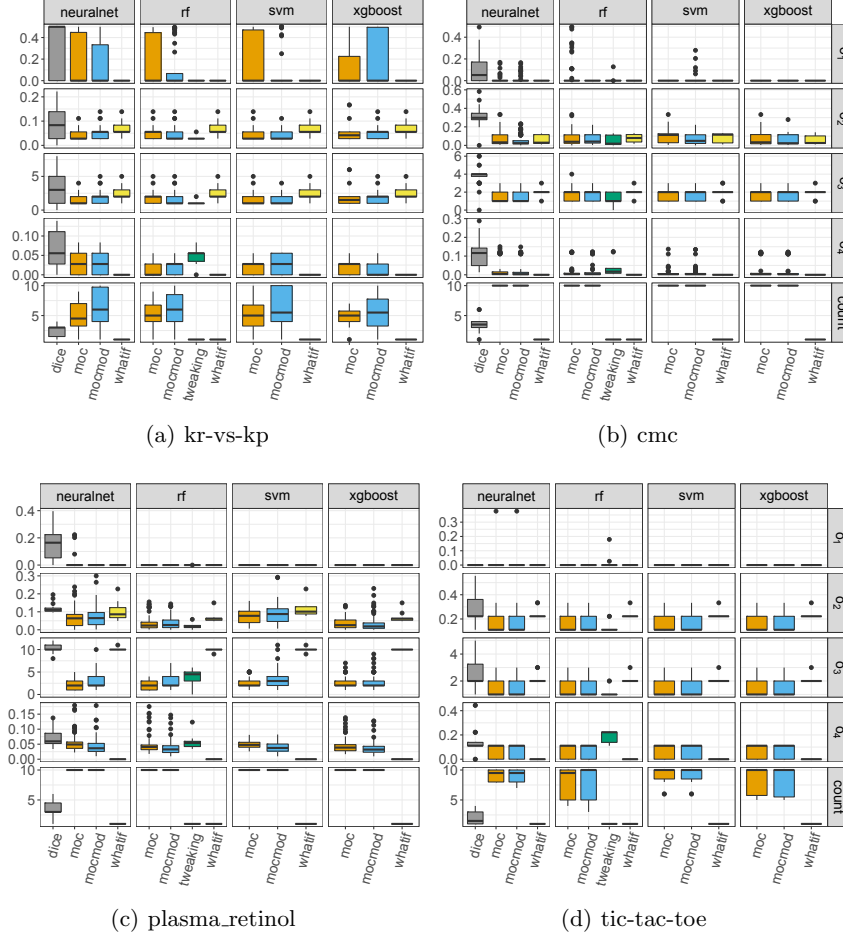


Fig. 5. Boxplots of the objective values and number of nondominated counterfactuals (*count*) per dataset and model for MOC with our proposed strategies for initialization and mutation (*mocmod*), MOC without these modifications, Whatif, DiCE, Recurse and Tweaking. Lower values are better except for *count*.

mit `aaae8fa`). For DiCE [24], we used the ‘DiverseCF’ version proposed by the authors [24] and left the control parameters at their defaults. We used the inverse mean absolute deviation for the feature weights. For datasets where the mean absolute deviation of a feature was zero, we set the feature weight to 10. We used the Python implementation of DiCE available on Github: <https://github.com/microsoft/DiCE> (commit `fed9d27`).

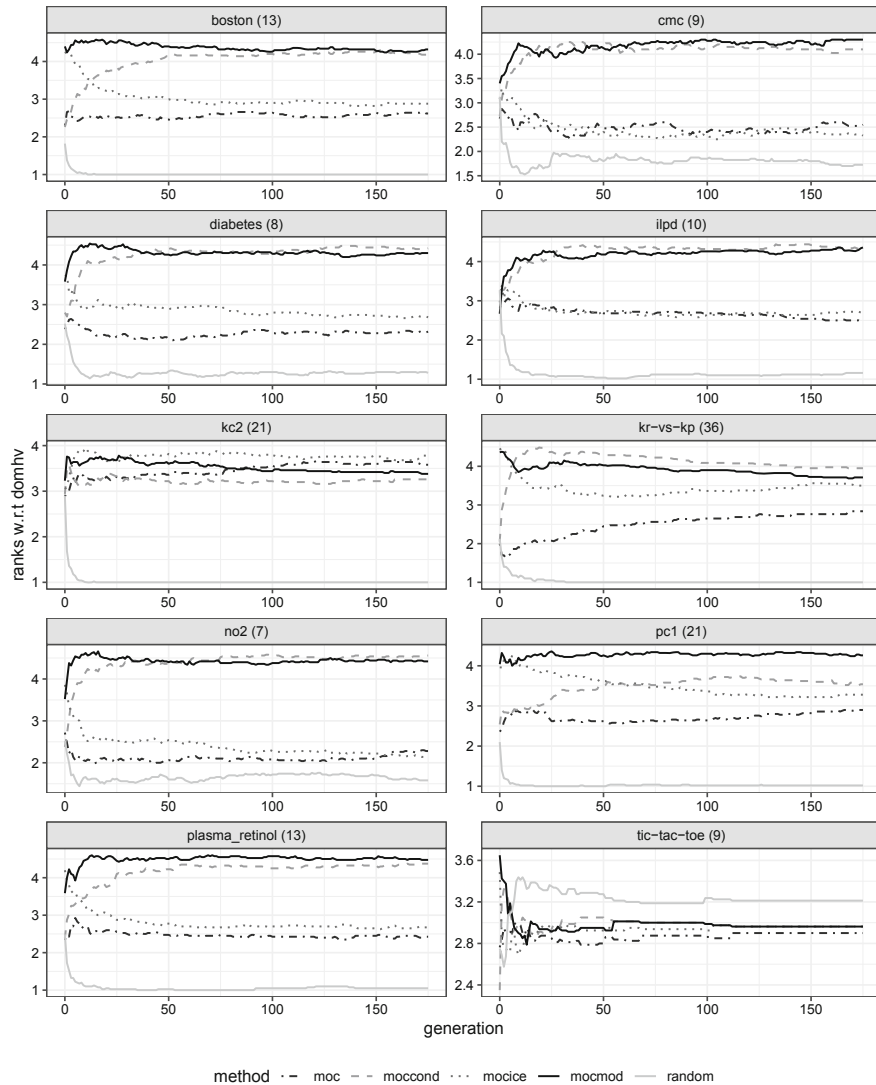


Fig. 6. Comparison of the ranks w.r.t. the dominated HV ($domhv$) per generation and per benchmark dataset averaged over all models. The numbers in parentheses indicate the number of features. For each approach, the population size of each generation was 20. Higher ranks are better. Legend: *moc*: MOC without modifications; *moccond*: MOC with the conditional mutator; *mocice*: MOC with the ICE curve variance initialization; *mocmod*: MOC with both modifications; *random*: random search.

References

1. Allaire, J., Chollet, F.: keras: R Interface to ‘Keras’ (2019). <https://keras.rstudio.com>, R package version 2.3.0
2. Avila, S.L., Krähenbühl, L., Sareni, B.: A multi-niching multi-objective genetic algorithm for solving complex multimodal problems. In: OIPE. Sorrento, Italy (2006). <https://hal.archives-ouvertes.fr/hal-00398660>
3. Binder, M., Moosbauer, J., Thomas, J., Bischl, B.: Multi-Objective Hyperparameter Tuning and Feature Selection using Filter Ensembles (2019). Accepted at GECCO 2020
4. Bischl, B., et al.: mlr: Machine Learning in R. *J. Mach. Learn. Res.* **17**(170), 1–5 (2016). <http://jmlr.org/papers/v17/15-066.html>, R package version 2.17
5. Breiman, L.: Statistical modeling: the two cultures. *Stat. Sci.* **16**(3), 199–231 (2001). <https://doi.org/10.1214/ss/1009213726>
6. Deb, K., Agarwal, R.B.: Simulated binary crossover for continuous search space. *Complex Syst.* **9**, 115–148 (1995)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002). <https://doi.org/10.1109/4235.996017>
8. Dhurandhar, A., Pedapati, T., Balakrishnan, A., Chen, P., Shanmugam, K., Puri, R.: Model Agnostic Contrastive Explanations for Structured Data. *CoRR abs/1906.00117* (2019). <http://arxiv.org/abs/1906.00117>
9. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat.* **24**(1), 44–65 (2015). <https://doi.org/10.1080/10618600.2014.907095>
10. Gower, J.C.: A general coefficient of similarity and some of its properties. *Biometrics* **27**(4), 857–871 (1971)
11. Grath, R.M., et al.: Interpretable Credit Application Predictions With Counterfactual Explanations. *CoRR (abs/1811.05245)* (2018). <http://arxiv.org/abs/1811.05245>
12. Greenwell, B.M., Boehmke, B.C., McCarthy, A.J.: A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755* (2018)
13. Hofmann, H.: German Credit Risk (2016). <https://www.kaggle.com/uciml/german-credit>. Accessed 25 Jan 2020
14. Hothorn, T., Zeileis, A.: Transformation Forests (2017)
15. Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., Ghosh, J.: Towards Realistic Individual Recourse and Actionable Explanations in black-box decision making systems. *CoRR abs/1907.09615* (2019). <http://arxiv.org/abs/1907.09615>
16. Karimi, A., Barthe, G., Balle, B., Valera, I.: Model-Agnostic Counterfactual Explanations for Consequential Decisions. *CoRR (abs/1905.11190)* (2019). <http://arxiv.org/abs/1905.11190>
17. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations, December 2014
18. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: Comparison-Based Inverse Classification for Interpretability in Machine Learning. *CoRR (abs/1712.08443)* (2017). <http://arxiv.org/abs/1712.08443>
19. Li, R., et al.: Mixed integer evolution strategies for parameter optimization. *Evol. Comput.* **21**(1), 29–64 (2013)

20. Looveren, A.V., Klaise, J.: Interpretable Counterfactual Explanations Guided by Prototypes. CoRR abs/1907.02584 (2019). <http://arxiv.org/abs/1907.02584>
21. López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T.: The irace package: iterated racing for automatic algorithm configuration. Oper. Res. Perspect. **3**, 43–58 (2016). <https://doi.org/10.1016/j.orp.2016.09.002>, <http://www.sciencedirect.com/science/article/pii/S2214716015300270>, R package version 3.4.1
22. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, pp. 4765–4774 (2017)
23. Molnar, C., Bischl, B., Casalicchio, G.: iml: an R package for interpretable machine learning. JOSS **3**(26), 786 (2018). <https://doi.org/10.21105/joss.00786>
24. Mothilal, R.K., Sharma, A., Tan, C.: Explaining Machine Learning Classifiers through Diverse Counterfactual explanations. CoRR (abs/1905.07697) (2019). <http://arxiv.org/abs/1905.07697>
25. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., Bie, T.D., Flach, P.: FACE: Feasible and Actionable Counterfactual Explanations (2019)
26. Radulescu, A., López-Ibáñez, M., Stützle, T.: Automatically improving the anytime behaviour of multiobjective evolutionary algorithms. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 825–840. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37140-0_61
27. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why should i trust you?” Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)
28. Russell, C.: Efficient Search for Diverse Coherent Explanations. CoRR (abs/1901.04909) (2019). <http://arxiv.org/abs/1901.04909>
29. Sharma, S., Henderson, J., Ghosh, J.: CERTIFAI: Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models. CoRR abs/1905.07857 (2019). <http://arxiv.org/abs/1905.07857>
30. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput. **23**, 828–841 (2017)
31. Syswerda, G.: Uniform crossover in genetic algorithms. In: Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 2–9. Morgan Kaufmann Publishers Inc., San Francisco (1989)
32. Tolomei, G., Silvestri, F., Haines, A., Lalmas, M.: Interpretable predictions of tree-based ensembles via actionable feature tweaking. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2017, pp. 465–474. ACM, New York (2017). <https://doi.org/10.1145/3097983.3098039>
33. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, pp. 10–19. ACM, New York (2019). <https://doi.org/10.1145/3287560.3287566>
34. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: OpenML: networked science in machine learning. SIGKDD Explor. **15**(2), 49–60 (2013). <https://doi.org/10.1145/2641190.2641198>
35. Wachter, S., Mittelstadt, B.D., Russell, C.: Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. CoRR (abs/1711.00399) (2017). <http://arxiv.org/abs/1711.00399>
36. Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viégas, F.B., Wilson, J.: The What- If Tool: Interactive Probing of Machine Learning Models. CoRR abs/1907.04135 (2019). <http://arxiv.org/abs/1907.04135>

37. White, A., d'Avila Garcez, A.: Measurable Counterfactual Local Explanations for Any Classifier (2019)
38. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms—a comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0056872>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



9 Multi-Objective Counterfactual Fairness

Contributing Article

Dandl S, Pfisterer F, Bischl B (2022b). “Multi-Objective Counterfactual Fairness.” In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '22, p. 328–331. Association for Computing Machinery, New York, NY, USA. doi : 10.1145/3520304.3528779

Declaration of Contributions

Susanne Dandl and Florian Pfisterer made equal contributions to this project. While Florian Pfisterer had the initial idea for the project, the general framework heavily builds upon Susanne Dandl’s previous work – the contribution of Chapter 8. They jointly implemented the proposed method and ran the experiments. They wrote the manuscript together and made equal improvements and revisions.

Contributions of Co-authors

Bernd Bischl consistently provided guidance and valuable input throughout the entire process.



Multi-Objective Counterfactual Fairness

Susanne Dandl*
LMU Munich
Munich, Germany

Florian Pfisterer*
LMU Munich
Munich, Germany

Bernd Bischl
LMU Munich
Munich, Germany

ABSTRACT

When machine learning is used to automate judgments, e.g. in areas like lending or crime prediction, incorrect decisions can lead to adverse effects for affected individuals. This occurs, e.g., if the data used to train these models is based on prior decisions that are unfairly skewed against specific subpopulations. If models should automate decision-making, they must account for these biases to prevent perpetuating or creating discriminatory practices. Counterfactual fairness audits models with respect to a notion of fairness that asks for equal outcomes between a decision made in the real world and a counterfactual world where the individual subject to a decision comes from a different protected demographic group. In this work, we propose a method to conduct such audits without access to the underlying causal structure of the data generating process by framing it as a multi-objective optimization task that can be efficiently solved using a genetic algorithm.

CCS CONCEPTS

• Computing methodologies → Supervised learning by classification; • Mathematics of computing;

KEYWORDS

machine learning, fairness, counterfactuals, multi-objective

ACM Reference Format:

Susanne Dandl, Florian Pfisterer, and Bernd Bischl. 2022. Multi-Objective Counterfactual Fairness. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion)*, July 9–13, 2022, Boston, MA, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3520304.3528779>

1 INTRODUCTION

Machine learning (ML) is increasingly used to automate judgments in areas like lending, hiring, or predictive policing. Decisions made by such systems cannot only lead to adverse effects for affected individuals, but also shape future data that are collected (or not collected) [1], e.g., by not collecting data on individuals denied a loan. Such adverse effects are ethically or legally problematic when they disproportionately affect protected subgroups, e.g., based on race, gender, or sexual orientation. Several reasons lead to unfair predictions, such as a lack of representative data or differences in data quality between subgroups. We focus on a scenario where the labels used to train machine learning models are biased on prior

*Both authors contributed equally to this research.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9268-6/22/07.
<https://doi.org/10.1145/3520304.3528779>

decisions which are unfairly skewed against a specific subpopulation. If such biases exist in the data, models must take them into account in order to prevent such injustices.

Several contributions have addressed this topic and have argued that a causal perspective is required to address the problem [9, 16]. This has resulted in a variety of (causal) fairness notions [15, 16, 23] that can be used to audit fairness algorithms. Counterfactuals [20] provide a causal, interpretable perspective to answer *what-if* questions about alternative (counterfactual) worlds. From a perspective of fairness, this allows us to answer questions such as: *Would the model's prediction change if the person had been male instead of female?* This requires access to the underlying (causal) mechanism generating the data, e.g., in the form of a *directed acyclic graph* (DAG, c.f. [20]), which are often ambiguous, especially in the context of high dimensional data.

Introductory Example In order to provide some intuition, we use the law school example from [16]. The directed acyclic graph for the postulated data generating process is shown in Figure 1a. Sex, race as well as a latent variable *knowledge* (K) influence the result in the law school admission test (LSAT), GPA and the first-year average grade (FYA). Instantiating a counterfactual instance x^* with, e.g., a changed variable Sex requires adapting the dependent variables *LSAT*, *GPA* and *FYA*. A ML model is now used to predict *FYA* from all other observed variables (Figure 1b). A fair model should now predict the same *FYA* regardless of x and x^* .

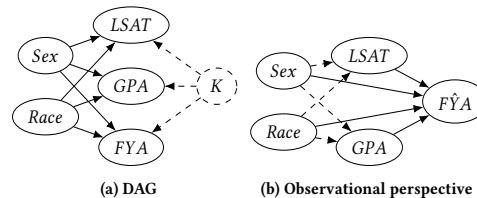


Figure 1: Law school example from [16].

Contributions: We propose a method to audit predictive models with respect to a fairness notion that relies on counterfactuals. Counterfactuals are found as solutions to a multi-objective optimization procedure, inspired by [7]. We argue that we can find realistic counterfactual examples by carefully crafting the objectives used for optimization. Due to the flexibility of the evolutionary algorithm used to tackle the resulting optimization problem, we can furthermore incorporate additional constraints in the optimization problem, allowing to attain more realistic and actionable counterfactuals. Unlike other methods, the multi-objective nature of our optimization problem allows us to return a Pareto-optimal set of diverse counterfactuals that can be used to assess fairness. Our

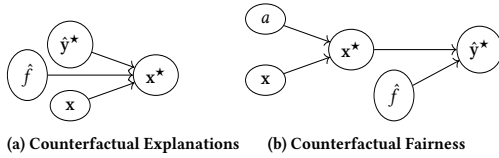


Figure 2: Generating counterfactuals x^* as explanations (CFE) (left) and for fairness (CFF) (right) for an observations x and predictor \hat{f} . The role of counterfactual prediction \hat{y}^* differs in both cases: While \hat{y}^* is incorporated into the generation of counterfactuals for CFEs, in CFF, the counterfactuals are first generated by striving for a different protected class a , and subsequently their counterfactual predictions are compared.

method does not require access to the underlying causal DAG and can therefore be used when such information is not available.

2 RELATED WORK

Fairness broadly asks that there is no *disproportionate* treatment between individuals depending on protected groups such as race, gender, or sexual orientation. A large body of work has previously studied differing notions of fairness [1, 18], often based on subgroup statistics in observational data [2, 4, 6, 13], while other notions of fairness argue to *treat similar persons similarly* [11] or argue for taking a causal perspective into account [5, 15, 16]. We follow the line of argumentation proposed in [16], which argues that the distribution over predictions should remain unchanged between the observed universe and a *counterfactual* universe in which an individual has different protected attributes. While [16] propose an algorithm that implements this definition, it requires access to the underlying DAG. One line of work implements notions similar to ours that do not require access, such as *FlipTest* [3], which uses a generative model approximating an optimal transport mapping to generate counterfactuals.

The notion of counterfactuals has been similarly used to improve *model interpretability*, answering which change in inputs would lead to a different model prediction [22]. These methods can generate potentially unrealistic out-of-distribution samples, which can jeopardize derived conclusions. For this reason, methods were proposed [7, 21] which focus on generating *plausible* counterfactuals. This is especially important in the context of algorithmic recourse. Karimi et al. [14] argue that explanations should be *actionable* but also *realistic* in the sense that they take into account the (causal) structure of the world from which they are obtained. This scenario differs from counterfactual fairness, since it aims at counterfactuals that lead to different model predictions. In contrast, counterfactual fairness notions observe the amount of change in a prediction from an instance to its counterfactual example. This difference is visualized in Figure 2.

Our method is heavily inspired by the MOC method described in [7], which was proposed in the context of finding multiple counterfactual explanations. In contrast, our method is used to find realistic counterfactual examples that allow auditing ML models with respect to counterfactual fairness for individual observations; when

applied to multiple observations, we could also obtain a global assessment. We similarly formulate a multi-objective optimization problem that can be efficiently solved using evolutionary algorithms. In order for our counterfactuals to be realistic and actionable, we carefully craft objectives and mutation operators used in the search.

3 METHODOLOGY

Let $\hat{f}(x) : \mathcal{X} \mapsto \mathbb{R}$ denote a model fitted to approximate the relationship between features x and a target variable of interest y , which are i.i.d. samples from a data generating distribution \mathbb{P}_{xy} . We assume that our data contain feature(s) A defining the protected class and define $Z \equiv X \setminus A$ as the set of all other observable features. For a data point x , we define a counterfactual observation as x^* with prediction $\hat{y}^* := \hat{f}(x^*)$. Counterfactuals that arise from intervention $A \leftarrow a$ could equivalently be denoted as $x_{A \leftarrow a}$ [20]. For ease of exposition, we restrict ourselves to classification models that predict probabilities throughout the manuscript. Extensions to regression models are straightforward once prediction thresholds are specified.

3.1 Counterfactual Fairness

We first restate the definition of counterfactual fairness from [16]. It assumes a causal model (U, X, F) , with U as a set of latent background variables not caused by any observed variables X , and F as a set of causal equations. \hat{Y} denotes a predictor that contrary to \hat{f} depends on X and U . The resulting \hat{Y} for intervention $A \leftarrow a$ is denoted as $\hat{Y}_{A \leftarrow a}(U)$.

DEFINITION 1 (COUNTERFACTUAL FAIRNESS [16]). *Predictor \hat{Y} is counterfactually fair if under arbitrary context $Z = z$ and $A = a$,*

$$P(\hat{Y}_{A \leftarrow a}(U) = y \mid Z = z, A = a) = P(\hat{Y}_{A \leftarrow a'}(U) = y \mid Z = z, A = a),$$

for all y and for any value a' attainable by A .

This suggests that changing A while keeping features that are not causally reliant on A constant has no effect on the distribution of Y . The computation of U and $\hat{Y}_{A \leftarrow a}$ is complex and requires access to the underlying DAG. We therefore state a similar criterion below that is practically applicable without access to the DAG. Note that the counterfactual instance is not necessarily *deterministic*, and the desired counterfactual can stem from a distribution of counterfactual instances.

3.2 A Practical Instantiation

In practical scenarios without access to the DAG, there is little chance to recover U . More realistically, our model uses x to predict the outcome of interest. Instead, we can therefore ask that the equality in Definition 1 holds between a data point x and its counterfactual x^* . We now state a version of counterfactual fairness that can be practically applied to observational data:

DEFINITION 2 (COUNTERFACTUAL FAIRNESS IN PRACTICE). *Predictor \hat{Y} is counterfactually fair if under any context $Z = z$ and $A = a$,*

$$P(\hat{f}(x_{A \leftarrow a}) = y \mid Z = z, A = a) = P(\hat{f}(x_{A \leftarrow a'}) = y \mid Z = z, A = a)$$

for all y and for any value a' attainable by A .

3.3 Generating Counterfactuals

The remaining task is now to generate counterfactuals $\mathbf{x}^* := \mathbf{x}_{A \leftarrow a'}$ which should fulfill the following requirements: (1) the counterfactual should be **valid**, such that it has high likelihood w.r.t. the distribution of the desired protected class $P_{X_{A=a'}}$; (2) the counterfactual should be **close** to the original observation; (3) the counterfactual should be **plausible** such that it lies in a high-density region w.r.t. the full dataset. Similar to [7], we translate our customized requirements into the following optimization problem:

$$\min_{\mathbf{x}^*} \mathbf{o}(\mathbf{x}^*) := \min_{\mathbf{x}^*} (o_{valid}(\mathbf{x}^*), o_{close}(\mathbf{x}^*, \mathbf{x}), o_{plaus}(\mathbf{x}^*, \mathbf{X}^{obs}))$$

with $\mathbf{o} : \mathcal{X} \rightarrow \mathbb{R}^3$ and \mathbf{X}^{obs} being the observed data.

The first objective o_{valid} quantifies whether \mathbf{x}^* truly stems from the desired protected group a' . We operationalize it for minimization using an additional predictor \hat{g} that is trained to predict whether a datapoint \mathbf{x}^* does not belong to the protected group a' .

$$o_{valid}(\mathbf{x}^*) = \hat{g}(\mathbf{x}^*)$$

The second and third objectives o_{close} and o_{plaus} are similar to the ones proposed by [7]. o_{close} quantifies the distance between the counterfactual \mathbf{x}^* and the original datapoint \mathbf{x} using an augmentation of the Gower distance (see c.f. [7]).

The third objective o_{plaus} quantifies the weighted average Gower distance between \mathbf{x}^* and the k nearest observed data points $\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[k]} \in \mathbf{X}^{obs}$ as an empirical approximation of how likely \mathbf{x}^* originates from the distribution of \mathcal{X} :

$$o_{plaus}(\mathbf{x}^*, \mathbf{X}^{obs}) = \sum_{i=1}^k w^{[i]} \frac{1}{p} \sum_{j=1}^p \delta_G(x_j^*, x_j^{[i]}) \in [0, 1]$$

where $\sum_{i=1}^k w^{[i]} = 1$. We optimize counterfactuals using an NSGA-II [8] variant adapted to the scenario of generating counterfactual instances proposed by [7], including their described modifications. The algorithm uses nature-inspired methods such as selection, mutation and recombination to steer a randomly initialized population towards the optimal solution (see Appendix A for details). This yields a set of Pareto-optimal counterfactuals that can be subsequently used to evaluate algorithms with respect to our practical notion of counterfactual fairness. The Pareto set can be interpreted as a distribution over counterfactuals (as defined by the objectives), reflecting the fact that *real* counterfactuals can be stochastic due to stochasticity in the data generating process as well as uncertainty in the estimation of required quantities.

Since we seek counterfactuals with a high likelihood of coming from the distribution of the desired protected class $P_{X_{A=a'}}$, we base the fairness notions of Section 4 on samples with high values of o_{valid} letting the user define a lower threshold for o_{valid} . We assume that this Pareto-optimal and valid subset approximates the distribution over counterfactuals for a single data point \mathbf{x} .

Actionable Counterfactuals. By defining additional customized operators or objectives (e.g., sparsity constraints), our method can be further adapted to more closely reflect the real-world data generating processes. This includes carefully designed mutation operators that constrain the allowable changes to features: values for non-actionable features (e.g., age) could be frozen, or monotonicity constraints could be considered such that an increase in one feature

leads to an increase or decrease in another feature [19]. Furthermore, we can accelerate the convergence to the Pareto front by initializing the first population of the NSGA-II with observations from \mathbf{X}^{obs} with $A = a'$. These observations per definition should have low values both for o_{valid} and o_{plaus} .

3.4 Evaluating for Counterfactual Fairness

A counterfactual generation procedure $gen : \mathcal{X} \rightarrow \mathcal{X}^*$ (such as the one proposed above) turns an instance \mathbf{x} into a set of counterfactual instances \mathbf{X}^* . We now define fairness criteria based on generated counterfactuals:

DEFINITION 3 (INSTANCE-WISE COUNTERFACTUAL UNFAIRNESS). For a single individual \mathbf{x} and a set of corresponding generated counterfactuals \mathbf{X}^* , we define unfairness as:

$$icuf(\mathbf{x}) = |\mathbb{E}_{\mathbf{x}^* \sim gen(\mathbf{x})} [\hat{f}(\mathbf{x}) - \hat{f}(\mathbf{x}^*)]|.$$

Computing the norm reflects the fact, that our notion does not differentiate between the direction of the unfairness (e.g., if \hat{f} favors or disadvantages the individual).

DEFINITION 4 (GLOBAL COUNTERFACTUAL UNFAIRNESS). For a distribution over datapoints \mathcal{X} and a set of sets of corresponding generated counterfactuals \mathcal{X}^* , we define a global notion of unfairness:

$$gcu(\mathcal{X}) = \mathbb{E}_{\mathbf{x} \sim \mathcal{X}} [icuf(\mathbf{x})].$$

Taking the expectation simultaneously reduces variance in the estimation and results in more robust estimates. Note that \hat{f} for our purposes can be a predicted probability. By thresholding predictions, we can simultaneously obtain *FlipSets* – the set of points for which the classification switches between the original instance and the counterfactual – and subsequently create *transparency reports* [3].

4 EMPIRICAL EVALUATION

Our goal is to create realistic counterfactuals. We therefore use the data generating process (DGP) of the *law school dataset* from [16] to generate data and *true* counterfactuals \mathbf{x}' , while we present results for another dataset in the supplementary material. We describe experimental details in Appendix B.

RQ1: Does our method generate realistic counterfactuals?

We present a visual comparison using t-SNE embeddings in Figure 3. Generated counterfactuals are found in high-density regions of the data and close to instances of the desired class. The true counterfactual is surrounded by generated counterfactuals. The average minimum Gower distance between \mathbf{x}^* and \mathbf{x}' is 0.069. We further quantify this in Table 1 by comparing our counterfactuals \mathbf{x}^* to two simple baselines: x^{nn} , the nearest neighbor of \mathbf{x} with desired protected attribute a' and x^{rnd} , a random observation. Distances between generated counterfactuals are typically lower than random points, while distances between an instance and the true counterfactual are comparatively high.

RQ2: How does fairness reported by our method compare to simple baselines?

To investigate the faithfulness of our method and several baselines, we calculate their *gucf* to the one of true counterfactuals. Individual values as well as further experiments are reported in the supplementary material. Table 2 reports *gucf* across several baselines and

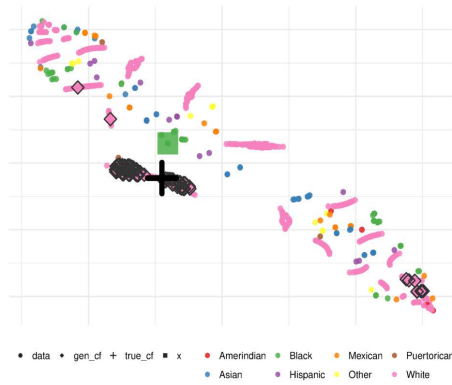


Figure 3: t-SNE plot for an instance of the law school DGP.

Table 1: Average Gower distances between x (original instance), x^* (generated counterfactual), x' (true counterfactual) and x^{rnd} (random point) and x^{nn} (nearest neighbor).

$d(x, x')$	$d(x, x^*)$	$d(x, x^{rnd})$	$d(x, x^{nn})$
0.16	0.07	0.192	0.008

Table 2: Mean $gcuf$ measured using true counterfactuals and different generation methods: The proposed method (ours) and two baselines: *flip*, flipping the protected attribute $A = a'$ in X , and *nn*, the nearest neighbors with $A = a'$.

$gcuf_{true}$	$gcuf_{ours}$	$gcuf_{flip}$	$gcuf_{nn}$
$0.277 \pm .003$	$0.278 \pm .003$	$0.265 \pm .004$	$0.318 \pm .004$

$gcuf$ obtained using true counterfactuals. Reported values using x^* are considerably closer to values estimated for true counterfactuals.

5 OUTLOOK

This manuscript proposes and evaluates a method for evaluating predictive models with respect to a counterfactual notion of individual and global fairness. Our method does not require access to the DAG generating the data, accounts for stochasticity by returning a Pareto-optimal set of counterfactuals, and is flexible enough for adoption to the needs of individual use cases. It is important to note that the validity of fairness auditing as proposed in our method heavily relies on the validity of generated counterfactuals, which is discussed in detail in Appendix C. In future work, we would like to improve the procedure used to find counterfactuals for a set of instances. The current procedure requires an inefficient loop across N observations that can hopefully be expedited by further tweaks to the optimization procedure. In a different line of work, we want to incorporate *path-based* notions of counterfactual fairness [5], which would allow for the definition of fair paths determined, e.g., due to principles such as business necessity (c.f. [12]).

ACKNOWLEDGMENTS

This work is funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

REFERENCES

- [1] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. *Fairness and Machine Learning*. fairmlbook.org. <http://www.fairmlbook.org>.
- [2] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2018. Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods & Research* (Aug. 2018), 42 pages. <https://doi.org/10.1177/0049124118782533> arXiv:1703.09207
- [3] Emily Black, Samuel Yeom, and Matt Fredrikson. 2020. Flptest: fairness testing via optimal transport. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 111–121.
- [4] Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21, 2 (2010), 277–292. <https://doi.org/10.1007/s10618-010-0190-x>
- [5] Silvia Chiappa. 2019. Path-specific counterfactual fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 7801–7808.
- [6] Alexandra Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (June 2017), 153–163. <https://doi.org/10.1089/big.2016.0047> arXiv:1703.00056
- [7] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. 2020. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature*. Springer, 448–469.
- [8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (April 2002), 182–197. <https://doi.org/10.1109/4235.996017>
- [9] Simon DeDeo. 2014. Wrong side of the tracks: Big data and protected categories. *arXiv preprint arXiv:1412.4643* (2014).
- [10] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml>
- [11] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, 214–226.
- [12] Susan S Grover. 1995. The business necessity defense in disparate impact discrimination cases. *Ga. L. Rev.* 30 (1995), 387.
- [13] Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems* 29 (Dec. 2016), 3323–3331. <https://doi.org/10.5555/3157382.3157469> arXiv:1610.02413
- [14] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. 2021. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 353–362.
- [15] Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. 2017. Avoiding discrimination through causal reasoning. *arXiv preprint arXiv:1706.02744* (2017).
- [16] Matt J Kusner, Joshua R Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *arXiv preprint arXiv:1703.06856* (2017).
- [17] Rui Li, Michael T.M. Emmerich, Jeroen Eggermont, Thomas Bäck, M. Schütz, J. Dijkstra, and J. H.C. Reiber. 2013. Mixed Integer Evolution Strategies for Parameter Optimization. *Evolutionary Computation* 21, 1 (2013), 29–64.
- [18] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. 54, 6, Article 115 (jul 2021), 35 pages. <https://doi.org/10.1145/3457607>
- [19] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) (FAT* '20). Association for Computing Machinery, New York, NY, USA, 607–617. <https://doi.org/10.1145/3351095.3372850>
- [20] Judea Pearl. 2009. Causal inference in statistics: An overview. *Statistics surveys* 3 (2009), 96–146.
- [21] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. *FACE: Feasible and Actionable Counterfactual Explanations*. Association for Computing Machinery, New York, NY, USA, 344–350. <https://doi.org/10.1145/3375627.3375850>
- [22] Sandra Wachter, Brent Daniel Mittelstadt, and Chris Russell. 2018. Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harvard Journal of Law and Technology* 31, 2 (2018), 841–887.
- [23] Junzhe Zhang and Elias Bareinboim. 2018. Equality of Opportunity in Classification: A Causal Approach. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2018/file/ff1418e8cc993fe8abcfce3ce2003e5c5-Paper.pdf>

A NSGA-II

NSGA-II [8] first initializes a random set of candidates (in our case counterfactual instances) which are evaluated by the proposed objectives. The best candidates are recombined in pairs and then slightly mutated to generate new candidates. Old and new candidates are ranked according to their objective values using non-dominated sorting and crowding distance sorting. The first aims at optimality, the second at diversity of the objective values. Based on this ranking, the best candidates are selected for the next generation. In subsequent generations, recombination, mutation and selection are repeated based on the updated population. In the end, the Pareto optimal set over all candidates is returned. Compared to the originally proposed NSGA-II, the method by Dandl et al. [7] uses mutation and recombination methods [17] to cover mixed (discrete and continuous) search spaces, and a crowding distance sorting that additionally considers diversity in the feature space.

B EXPERIMENTAL DETAILS

The goal of the experimental evaluation is two-fold: Since fairness metrics *icuf* (Definition 3) and *gcuf* (Definition 4) rely on the assumption that generated counterfactuals are realistic, we investigate this assumption in downstream experiments based on the *adult* dataset [10]. Simultaneously, our ultimate goal is to check for *instance-wise* or *global* unfairness, therefore, we also need to ascertain that our numeric estimates of unfairness correspond to the real unfairness. The latter can only be observed in scenarios where true counterfactuals are observable – which is not the case for the *adult* dataset. Therefore, we investigate our goals in a simulation scenario based on the law school example described in the introduction. The code to reproduce all experiments is available in a GitHub repository: <https://github.com/pfistfl/counterfactuals/tree/moccf/paper/experiments>. Optimization is generally run for ≤ 30 generations of the adapted NSGA-II algorithm. Generating counterfactuals for a single instance generally takes around 15 seconds for 30 generations.

Quality of generated counterfactuals

We generate the counterfactual for a given instance x and use t-SNE embeddings to visualize the generated counterfactuals $x^* \in X^*$. We visually judge the quality of generated counterfactuals using the following criteria:

- x^* should lie in high-density regions of the data.
- x^* should lie in high-density regions for samples of X with the desired protected status.
- x^* should be close to the original instance x .

Adult. We trained a random forest model on the first 1000 samples of the *adult* dataset [10]. As a preprocessing step, we combined categories of the protected attribute race with few observations such that we receive three categories (*White*, *Black* and *Other*). For an instance with race *Black*, we generated counterfactuals $x_{A \leftarrow White}$. Figure 5 of the Pareto front reveals that the three objectives contradict each other, e.g., counterfactuals with low values in o_{valid} or o_{close} have higher values in o_{plaus} . The t-SNE embeddings in

Figure 4 show that generated counterfactuals are found in high-density regions of the data and close to instances of the desired class.



Figure 4: t-SNE plot for the adult dataset after 175 generations.

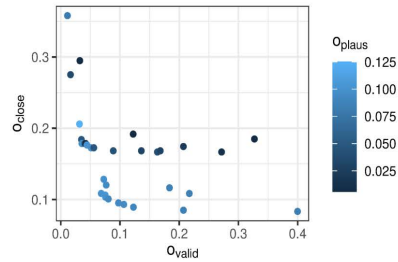


Figure 5: Plot of the Pareto front for the adult dataset after 175 generations.

Law School. We draw 1000 samples from the data generating process as described in [16] and detailed in the introduction. We then investigate the counterfactuals $x' := x_{A \leftarrow White}$ for all instances in X with race *Black*. We furthermore use the *FYA* variable in order to estimate a variable *PASS* (indicating whether a student will pass), where $PASS_{(i)} \sim Ber(\text{logit}(FYA_{(i)}))$ for each respective instance i . Given access to the *true* counterfactual x' , we can furthermore assess how close $x^* \in X^*$ lie to x' for example given the Gower distance. Results reported in Figure 3 are for a single instance, while distances reported in Table 1 are averaged across all instances with label *Black*. We did not include the protected attribute for calculating Gower distances.

Individual and global unfairness

We investigate global and individual level unfairness based on the law school example described in the introduction. We use the same experimental setup as described above. Since we have access to the data generating process in this simulated scenario, we can generate the true counterfactuals as well as counterfactuals generated using

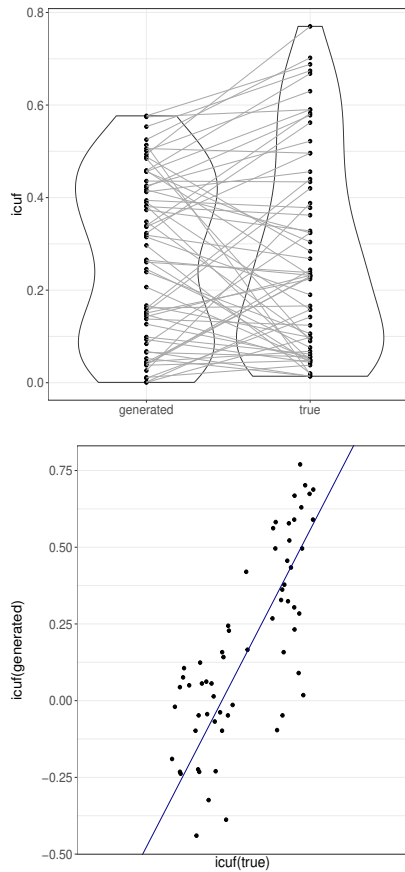


Figure 6: *Upper*: Comparison of *icuf* between generated counterfactuals (left) and true counterfactuals (right) for the law school example. The global *gcuf* is 0.268 and 0.319 respectively. *Lower*: Scatterplot of *icuf* for generated counterfactuals (x^*) and true counterfactuals (x') for the law school example.

the proposed method. The resulting *icuf* and *gcuf* for both true counterfactuals (right) and generated counterfactuals (left) are reported in Figure 6. While *icuf* is slightly underestimated, the global estimate of model unfairness (0.268) is reasonably close to the true one (0.319).

C ASSUMPTIONS AND VALIDITY OF GENERATED COUNTERFACTUALS

The goal of this work is to propose an alternative method for fairness auditing of machine learning models. In contrast to existing methods for observational data (cf. [13]), our method hopes to

generate *causally valid* counterfactuals. In the absence of an unambiguous DAG, there can be no guarantees that any generated counterfactual actually stems from the true distribution of counterfactuals – at best we can hope that we generate sufficiently similar datapoints given the specified objectives. Thus, we argue that our method (as well as other methods proposed in this context) should never be used in isolation, but as one additional perspective to detect potential biases in data. It is similarly important to consider fairness in its broader context, i.e., the actual outcomes that decisions based on ML models produce and their long-term effects, e.g., in the context of feedback loops. Furthermore, the question of whether a technical intervention in favor of possible other solutions is necessary for a given context needs to be thoroughly considered.

10 counterfactuals: An R Package for Counterfactual Explanation Methods

Contributing Article

Dandl S, Hofheinz A, Binder M, Bischl B, Casalicchio G (2023c). “**counterfactuals**: An R Package for Counterfactual Explanation Methods.” *arXiv 2304.06569 v2*, arXiv.org E-Print Archive. doi: 10.48550/arXiv.2304.06569

When this thesis was submitted, the article was under review by the *Journal of Statistical Software*.

Replication Code

The **counterfactuals** R package is available on CRAN and at <https://github.com/dandls/counterfactuals>. The code for replicating the benchmark study is available at https://github.com/slds-lmu/benchmark_2022_counterfactuals.


Declaration of Contributions

Susanne Dandl contributed essentially to the overall design of the R package. Her package on multi-objective counterfactual explanations (part of the contribution in Chapter 8) built the starting point for the counterfactuals package. She supervised the package development by Andreas Hofheinz and implemented some extensions. She also undertook the CRAN submission, including the required adaptations to the package. Susanne Dandl advised Andreas Hofheinz on designing the benchmark study and revised the study code. She executed the experiment, and implemented additional performance measures and visualization methods to report the results. Susanne conducted the included use cases and wrote the majority of the paper.

Contributions of Co-authors


Andreas Hofheinz implemented the majority of the package and large parts of the benchmark study code as part of his master thesis. The master thesis was supervised by Giuseppe Casalicchio, Susanne Dandl, and Martin Binder, who consistently provided guidance throughout the entire process. Andreas Hofheinz also wrote some parts of the manuscript. Giuseppe Casalicchio and Bernd Bischl provided valuable input to the design of the R package and benchmark study as senior authors. All co-authors helped to revise the manuscript.


counterfactuals: An R Package for Counterfactual Explanation Methods

Susanne Dandl 
LMU Munich
MCML

Andreas Hofheinz
LMU Munich

Martin Binder
LMU Munich
MCML

Bernd Bischl 
LMU Munich
MCML

Giuseppe Casalicchio 
LMU Munich
MCML

Abstract

Counterfactual explanation methods provide information on how feature values of individual observations must be changed to obtain a desired prediction. Despite the increasing amount of proposed methods in research, only a few implementations exist whose interfaces and requirements vary widely. In this work, we introduce the **counterfactuals** R package, which provides a modular and unified **R6**-based interface for counterfactual explanation methods. We implemented three existing counterfactual explanation methods and propose some optional methodological extensions to generalize these methods to different scenarios and to make them more comparable. We explain the structure and workflow of the package using real use cases and show how to integrate additional counterfactual explanation methods into the package. In addition, we compared the implemented methods for a variety of models and datasets with regard to the quality of their counterfactual explanations and their runtime behavior.

Keywords: counterfactual explanations, interpretable machine learning, R.

1. Introduction and related work

In recent years, counterfactual explanation methods have emerged as valuable techniques for explaining single predictions of black-box models. Denied loan applications serve as a common example; here, a counterfactual explanation (or *counterfactual* for short) could be: “You were denied a loan because your annual income was £30,000. If your income had been £45,000, you would have been offered a loan” (Wachter, Mittelstadt, and Russell 2018). More generally, counterfactuals address questions of the form: “For input \mathbf{x}^* , the model predicted y . What needs to be changed in \mathbf{x}^* so that the model predicts a desired outcome y' instead?”.

One advantage of counterfactuals is their human-friendly interpretability: as they simply suggest feature changes to obtain a desired outcome, they are comprehensible even to non-experts (Molnar 2022). In addition, counterfactual scenarios can help to detect biases of individual predictions (Wachter *et al.* 2018). There are several ways to change features to obtain a desired outcome, but not all of them are feasible. Therefore, counterfactual methods that

arXiv:2304.06569v2 [stat.ML] 15 Sep 2023

provide multiple (reasonable) counterfactuals and allow the user to assess their usefulness using domain knowledge are preferable (Dandl, Molnar, Binder, and Bischl 2020b). Counterfactual explanations are related to adversarial examples (Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow, and Fergus 2014), but the latter aim to deceive a model instead of explaining it (Freiesleben 2021).

Over the past few years, a variety of counterfactual explanation methods have been proposed. Overviews are given in Verma, Boonsanong, Hoang, Hines, Dickerson, and Shah (2022), Karimi, Schölkopf, and Valera (2021), and Stepin, Alonso, Catala, and Pereira-Fariña (2021). Most of the methods focus on classification models and use either optimization techniques or heuristic rules to search for counterfactuals. Existing methods are either model-specific in the sense that they are only applicable to certain model classes (e.g., linear or tree-based models) or model-agnostic, i.e., they are applicable to arbitrary models. Furthermore, the methods differ in whether and to what extent access to the underlying data is necessary, the number of counterfactuals they return, and the properties of counterfactuals targeted by a method (e.g., sparsity or actionability). We will present the most frequently targeted properties in Definition 1. Counterfactual explanation methods which explicitly target actionable feature changes are also called *recourse* (Verma *et al.* 2022).

Despite the increasing amount of proposed counterfactual methods in research, the current software landscape is rather sparse. To the best of our knowledge, the only counterfactual methods available in R (R Core Team 2022) as dedicated packages are *MOC* (Dandl *et al.* 2020b; Dandl, Molnar, and Binder 2020a) and Feature Tweaking (Tolomei, Silvestri, Haines, and Lalmas 2017; Kato 2018). Feature Tweaking is a model-specific method tailored to random forests and its R implementation only allows forests specifically trained with the **randomForest** package. In contrast, *MOC* is a model-agnostic method and its implementation allows all regression or classification models fitted with popular toolboxes such as **caret** (Kuhn 2021) and **mlr3** (Lang, Binder, Richter, Schratz, Pfisterer, Coors, Au, Casalicchio, Kotthoff, and Bischl 2019). Models of other packages can also be processed using a wrapper function. In Python (Van Rossum and Drake Jr 1995), the **CARLA** library (Pawelczyk, Bielawski, den Heuvel, Richter, and Kasneci 2021) provides a variety of (model-agnostic and model-specific) counterfactual explanation methods for classification models. **CARLA** currently calls the original Python implementations of the methods, which often only allow models of specific ML libraries as an input. Furthermore, a library for the model-agnostic method *NICE* (Brughmans and Martens 2022; Brughmans 2021) exists which could process all models fitted with **scikit-learn** (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, and Duchesnay 2011). Implementations of the methods MACE (Karimi, Barthe, Balle, and Valera 2020), MINT (Karimi *et al.* 2021) and LORE (Guidotti, Monreale, Ruggieri, Pedreschi, Turini, and Giannotti 2018) are available (Karimi and Mohammadi 2021; Guidotti 2018), but these are only meant to reproduce the experiments of the original paper, and are therefore limited to certain datasets and models. Apart from *MOC*, all the mentioned methods are not capable of returning multiple counterfactuals (in one run).

In summary, existing implementations are predominantly available in Python in different repositories or libraries and at different stages of development. R users can only access a limited number of methods, and the usability and comparability of these methods are severely limited because there is no common user interface. Most Python libraries only allow methods for classification models and focus primarily on methods returning a single counterfactual.

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 3

Contributions: With the `counterfactuals` package, we offer the first R package that provides a user-friendly and unified interface for model-specific as well as model-agnostic counterfactual explanation methods. Therefore, it complements other R-based toolkits for interpreting machine learning models such as `IML` (Molnar 2022) and `DALEX` (Biecek 2018). The package provides common functionalities to evaluate and visualize counterfactuals of diverse methods. It is flexible enough to be easily extended by other counterfactual methods for classification or regression models. Currently, the package provides three counterfactual explanation methods. We discuss some (optional) extensions we have made to these methods: first, to generalize them to diverse scenarios (for example, to regression models or multiclass classifiers), and second, to improve their comparability, for example, by letting the two methods, that return only one counterfactual, return several ones just like the third method. Our work is therefore one of the few that explicitly advocates methods that simultaneously generate multiple, qualitatively comparable counterfactuals rather than a single one. We are also among the first to provide an evaluation approach for *different sized* sets of counterfactuals by comparing the three implemented methods in a benchmark study. In contrast, previous work primarily focused on one counterfactual per method (de Oliveira and Martens 2021; Pawelczyk *et al.* 2021; Moreira, Chou, Hsieh, Ouyang, Jorge, and Pereira 2022). Because the package and benchmark study code are freely available, we encourage readers to add counterfactual approaches to our R package and compare them to the ones that have already been implemented.

In the upcoming section, we present the three currently implemented methods. In Section 3, we explain the overall structure and handling of the package as well as its most important functionalities. We present use cases for a regression and classification task to show the main functionalities of the package in Section 4, followed by an example in Section 5 illustrating how additional counterfactual explanation methods can be easily integrated into our package. In Section 6, we show the general setup and results of the benchmark study. We summarize our findings as well as open questions in Section 7.

2. Methodological background and extensions

Our definition of counterfactual explanations is based on the work of Dandl *et al.* (2020b) and Verma *et al.* (2022).

Definition 1 (Counterfactual explanation). Let $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$ be a prediction function with $\mathcal{X} \subset \mathbb{R}^p$ as the feature space. While our definition naturally covers regression models, for classification tasks, we assume that \hat{f} returns the score or probability for a predefined class of interest, usually the so-called positive class. Let further $\mathbf{X} := (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)})$ with $\mathbf{x}^{(i)} \in \mathcal{X}, i \in \{1, \dots, n\}$ be the observed data and $Y' = [Y'_l, Y'_u]$ be an interval of desired predictions. We define a point \mathbf{x} as a counterfactual explanation for an observation \mathbf{x}^* if \mathbf{x} fulfills (at least some of) the following desired properties:

- i *Validity*: \mathbf{x} leads to a desired prediction, i.e., $\hat{f}(\mathbf{x}) \in Y'$. This could be assessed, e.g., by (Dandl *et al.* 2020b)

$$\alpha_{\text{valid}}(\hat{f}(\mathbf{x}), Y') = \begin{cases} 0, & \text{if } \hat{f}(\mathbf{x}) \in Y' \\ \min_{y' \in Y'} |\hat{f}(\mathbf{x}) - y'|, & \text{otherwise} \end{cases}. \quad (1)$$

- ii *Proximity*: \mathbf{x} is close to \mathbf{x}^* , which could be measured, e.g., by the Gower distance d_G (Gower 1971) for mixed feature spaces

$$o_{\text{prox}}(\mathbf{x}, \mathbf{x}^*) = d_G(\mathbf{x}, \mathbf{x}^*) := \frac{1}{p} \sum_{j=1}^p \delta_G(x_j, x_j^*) \in [0, 1] \quad (2)$$

with

$$\delta_G(x_j, x_j^*) = \begin{cases} \frac{1}{\hat{R}_j} |x_j - x_j^*| & \text{if } x_j \text{ is numerical} \\ \mathbb{1}_{x_j \neq x_j^*} & \text{if } x_j \text{ is categorical} \end{cases}.$$

where $\hat{R}_j = \max(\mathbf{X}_j) - \min(\mathbf{X}_j)$ is the value range of feature j in \mathbf{X} .

- iii *Sparsity*: \mathbf{x} differs from \mathbf{x}^* in only a few features. This can be measured by the L_0 norm

$$o_{\text{sparse}}(\mathbf{x}, \mathbf{x}^*) = \|\mathbf{x} - \mathbf{x}^*\|_0 = \sum_{j=1}^p \mathbb{1}_{x_j \neq x_j^*}. \quad (3)$$

- iv *Plausibility*: \mathbf{x} is realistic, i.e., close to the data manifold. Metrics are the (weighted) Gower distance to the k closest training samples $\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[k]} \in \mathbf{X}$ (Dandl et al. 2020b)

$$o_{\text{plaus}}(\mathbf{x}, \mathbf{X}) = \sum_{i=1}^k w^{[i]} d_G(\mathbf{x}^{[i]}, \mathbf{x}^*) \in [0, 1] \text{ where } \sum_{i=1}^k w^{[i]} = 1 \quad (4)$$

or the reconstruction error of a variational autoencoder (VAE) trained on the training samples (Brughmans and Martens 2022).

- v *Actionability*: \mathbf{x} does not alter immutable features (e.g., country of birth) and only proposes changes within an actionable range (e.g., non-negative age).
- vi *Causality*: \mathbf{x} reflects the underlying causal structure and takes causal relations of features into account. This property could be only examined if the causal graph (Pearl 2009) is (at least partially) known (Karimi et al. 2020, 2021; Mahajan, Tan, and Sharma 2020). Since this is rarely the case, most counterfactual methods (including the ones implemented in the **counterfactuals** package) disregard this property (Verma et al. 2022).

While some desired properties have a common tendency, others are rather opposed: if an explanation is sparse (iii), it also tends to be proximal (ii), since a counterfactual tends to be close to the original data point when only a few features are changed. However, a counterfactual that is close to the original data point tends to have a similar prediction, which may be far from a desired prediction, thus making the counterfactual less valid (i). The exact interdependence between the properties depends on the prevailing circumstances. Existing counterfactual methods vary in the desired properties they consider and how they measure and optimize them. An overview of methods is given in Verma et al. (2022). The methods also vary in whether a single counterfactual or a set of diverse ones is generated for a \mathbf{x}^* . We argue that a set of counterfactuals is more valuable than a single one. This is because there could exist different equally good counterfactuals with the desired prediction (Rashomon effect (Breiman 2001)) and it is more likely that a set contains a counterfactual that satisfies a user's (hidden) preferences (Dandl et al. 2020b).

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 5

Below, we introduce the three counterfactual methods currently available in the **counterfactuals** package: *MOC* (Dandl *et al.* 2020b), *WhatIf* (Wexler, Pushkarna, Bolukbasi, Wattenberg, Viégas, and Wilson 2019), and *NICE* (Brughmans and Martens 2022). By addressing their limitations, we motivate *optional* extensions of the methods that we implemented in our package. In particular, these extensions enable all methods to return multiple counterfactuals for binary and multiclass classification models, as well as regression models.

2.1. Multi-objective counterfactual explanations

Original method

The multi-objective counterfactuals (*MOC*) method by Dandl *et al.* (2020b) searches for counterfactuals by solving a multi-objective minimization problem

$$\min_{\mathbf{x}} \mathbf{o}(\mathbf{x}) := \min_{\mathbf{x}} \left(o_{\text{valid}}(\hat{f}(\mathbf{x}), Y'), o_{\text{prox}}(\mathbf{x}, \mathbf{x}^*), o_{\text{sparse}}(\mathbf{x}, \mathbf{x}^*), o_{\text{plaus}}(\mathbf{x}, \mathbf{X}) \right). \quad (5)$$

The single objectives correspond to the desired properties *Validity*, *Proximity*, *Sparsity*, and *Plausibility* formalized in Equations 1 to 4 as part of Definition 1. *MOC* also considers *Actionability* by allowing the specification of “fixed features” that remain unchanged and of alteration ranges for continuous features.

To tackle the optimization problem in (5), *MOC* uses a customized version of the non-dominated sorting genetic algorithm (NSGA-II) of Deb, Pratap, Agarwal, and Meyarivan (2002): unlike the original algorithm, *MOC* employs mixed-integer evolutionary strategies (Li, Emmerich, Eggermont, Bäck, Schütz, Dijkstra, and Reiber 2013) to handle mixed feature spaces and computes the crowding distance not only in the objective space but also in the feature space. A description of the steps of the algorithm as implemented in the **counterfactuals** package is given in Algorithm 1 of Appendix A.

The algorithm first initializes a population. The authors proposed several strategies:

- *Random*: Feature values of new individuals are uniformly sampled from the range of observed values. Subsequently, some features are randomly reset to their initial value in \mathbf{x}^* to induce sparsity.
- *ICE curve*: As in *Random*, feature values are sampled from the range of observed values. Then, however, features are reset with probabilities relative to their feature importance: the higher the importance of a feature \mathbf{x}_j , the higher the probability that its values differ from \mathbf{x}_j^* . The importance of one feature is measured using the standard deviation of its corresponding individual conditional expectation (ICE) curve (Goldstein, Kapelner, Bleich, and Pitkin 2015).
- *Standard deviation*: This method is similar to *Random*, except that the sample ranges of numerical features are limited to one standard deviation from their value in \mathbf{x}^* .
- *Training data*: Contrary to the other strategies, individuals are drawn from non-dominated previous observations in the dataset. If insufficient observations are available, the remaining individuals are initialized by random sampling. Subsequently, some features are randomly reset to their initial value in \mathbf{x}^* (as for *Random*).

Dandl *et al.* (2020b) discussed only the first two strategies in their paper, although the third and fourth strategies were also available in their implementation (Dandl *et al.* 2020a). In subsequent generations, the algorithm recombines and mutates individuals of the population and their features with predefined probabilities so that the initial population evolves. For mutation, the authors state two approaches: the first is to apply a scaled Gaussian mutator to numerical features and a uniform discrete mutator to categorical features (Li *et al.* 2013); the second approach aims to take feature distributions into account by sampling conditionally on the other feature values using a transformation tree (Hothorn and Zeileis 2021).

After recombination and mutation, some features are randomly reset to their initial value in \mathbf{x}^* with prespecified probabilities to induce sparsity. The recombination and mutation steps in the algorithm can be customized via multiple control parameters. An overview is given in Appendix B.2. To emphasize *Validity* (i), individuals whose prediction exceeds a specified target distance $\epsilon \in \mathbb{R}_{\geq 0}$ can be penalized using the approach of Deb *et al.* (2002). *MOC* terminates either after a prespecified number of generations or when the hypervolume (HV) indicator (Zitzler and Thiele 1998) of the objectives in (5) does not improve for a prespecified number of consecutive generations. As counterfactuals, *MOC* returns all (unique) non-dominated individuals across all generations.

Contrary to most other methods, *MOC* is inherently applicable to both classification and regression tasks. Moreover, *MOC* does not require the user to weigh the objectives *a priori* and thus avoids the risk of arbitrarily affecting the solution set. Instead, it returns a Pareto set of counterfactuals so that the objectives can be weighted *a posteriori*.

Modifications

We did not rely on the previous implementation of *MOC* (Dandl *et al.* 2020a) in the **counterfactuals** R package. Instead, we reimplemented an updated version of *MOC*: we replaced the NSGA-II implementation in **mosmafs** (Binder, Dandl, and Moosbauer 2020) with its extended and more versatile successor **miesmuschel** (Binder 2023), and parameter spaces are now defined by the **paradox** package (Lang, Bischl, Richter, Sun, and Binder 2022) instead of **ParamHelpers** (Bischl, Lang, Richter, Bossek, Horn, and Kerschke 2020).

2.2. WhatIf

Original method

WhatIf is the counterfactual method for classification models proposed by Wexler *et al.* (2019) as part of the What-If Tool¹. Wexler *et al.* (2019) assume that the underlying model $\hat{h} : \mathcal{X} \rightarrow \mathcal{Y}$ predicts a class label and define the set of desired predictions Y' as the set of all labels other than the current one. As a counterfactual \mathbf{x}' for an observation \mathbf{x}^* , *WhatIf* returns the data point most similar to \mathbf{x}^* from previous observations $\tilde{\mathbf{X}} = \{\mathbf{x} \in \mathbf{X} : \hat{h}(\mathbf{x}) \neq \hat{h}(\mathbf{x}^*)\}$ whose predicted class is different from that of \mathbf{x}^* . This leads to the minimization problem:

$$\mathbf{x}' \in \underset{\mathbf{x} \in \tilde{\mathbf{X}}}{\operatorname{argmin}} d(\mathbf{x}, \mathbf{x}^*). \quad (6)$$

The function d is a slightly adapted version of the Gower distance (Equation 2): for numerical

¹<https://pair-code.github.io/what-if-tool/>

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 7

features, the authors scale the distances with the standard deviations $\hat{\sigma}_j$; for categorical features, the feature distances are set equal “to the probability that any two examples across the entire dataset would share the same value for that feature” if their values differ, and 0 otherwise (Wexler *et al.* 2019). By definition, *WhatIf* aims for valid (i), proximal (ii), and plausible (iv) counterfactuals. *WhatIf* often serves as a baseline method in benchmark studies (Dandl *et al.* 2020b; Schleich, Geng, Zhang, and Suci 2021; Carreira-Perpiñán and Hada 2021) because it is easily implementable and adaptable.

Modifications

For better comparability with *MOC*, we use the original Gower distance as the default for d in the **counterfactuals** package. We allow users to replace this with other dissimilarity measures (see Section 4.2.1). We also extended the method to work with \hat{f} that returns the probability of a prespecified class of interest for classification tasks instead of a hard label classifier \hat{h} . This allows us to define the set of desired predictions Y' as a probability interval $[Y'_l, Y'_u] \subseteq [0, 1]$. Additionally, our approach makes *WhatIf* applicable to regression tasks without further modifications. In this case, Y' can simply be any real interval. $\tilde{\mathbf{X}}$ is then redefined as $\tilde{\mathbf{X}} = \{\mathbf{x} \in \mathbf{X} : \hat{f}(\mathbf{x}) \in Y'\}$.

As argued in Section 1, methods that can find multiple counterfactuals for a single observation are preferable. Therefore, we implemented an extended *WhatIf* version that returns the $l \in \mathbb{N}$ closest data points of $\tilde{\mathbf{X}}$ to \mathbf{x}^* with the desired prediction. This is equivalent to minimizing the following objective instead of (6)

$$\{\mathbf{x}'_1, \dots, \mathbf{x}'_l\} \in \underset{\mathbf{z} \in \tilde{\mathbf{X}}, |\mathbf{Z}|=l}{\operatorname{argmin}} \sum_{\mathbf{z} \in \mathbf{Z}} d_G(\mathbf{z}, \mathbf{x}^*). \tag{7}$$

2.3. Nearest instance counterfactual explanations

Original method

Nearest instance counterfactual explanations (*NICE*) introduced by Brughmans and Martens (2022) is a counterfactual explanation method for binary score classifiers $\hat{f} : \mathcal{X} \rightarrow [-1, 1]$. Accordingly, they define the set of desired predictions Y' as the set of all scores that lead to a different class than the current one. *NICE* starts the counterfactual search for an observation \mathbf{x}^* by finding its most similar *correctly classified* instance \mathbf{x}_{nm} . Brughmans and Martens (2022) assess similarity by the heterogeneous euclidean overlap method (Wilson and Martinez 1997) with L_1 -norm aggregation, which corresponds to the Gower distance without averaging (i.e., Equation 2 without $\frac{1}{p}$).

Once \mathbf{x}_{nm} is found, *NICE* generates new instances in the first iteration ($m = 1$) by replacing single feature values of \mathbf{x}^* with the corresponding value of \mathbf{x}_{nm} . *NICE* evaluates the created instances with a reward function that optimizes either sparsity, proximity, or plausibility (see Brughmans and Martens 2022, for details).

If the prediction of the instance with the highest reward value is in Y' , the algorithm terminates and returns this instance as a counterfactual. Otherwise, *NICE* creates new instances in the next iteration by replacing single feature values of the best performing instance of the previous iteration with the corresponding value of \mathbf{x}_{nm} . The search continues as long as the prediction for the highest reward value instance is not in Y' .

Modifications

We generalized *NICE* for regression models and multiclass classifiers: first, we extend \hat{f} to predict real-values (regression) or the probability of a predefined class k , respectively (see Definition 1). Second, we conceptualize the search for \mathbf{x}_{nn} as the following minimization problem:

$$\mathbf{x}_{nn} = \underset{\mathbf{x} \in \tilde{\mathbf{X}}'}{\operatorname{argmin}} o_{\text{prox}}(\mathbf{x}, \mathbf{x}^*) \quad (8)$$

with o_{prox} as defined in Equation 2. For classification, $\tilde{\mathbf{X}}' = \{\mathbf{x} \in \mathbf{X} : \hat{f}(\mathbf{x}) \in Y' \wedge h(\hat{f}(\mathbf{x})) = y\}$ is the set of all correctly classified observations whose prediction is in the set of desired predictions Y' . y is the true class label of \mathbf{x} and $h(\cdot)$ is a transformation function that maps class scores onto class labels. For regression, $\tilde{\mathbf{X}}' = \{\mathbf{x} \in \mathbf{X} : \hat{f}(\mathbf{x}) \in Y' \wedge |\hat{f}(\mathbf{x}) - y| \leq \epsilon\}$ is the set of all observations with a prediction in the desired real interval Y' and a prediction error of less than a user-specified $\epsilon \in \mathbb{R}_{\geq 0}$. Similar to *WhatIf*, o_{prox} in Equation 8 could be replaced with user-defined distance measures in our implementation (demonstrated in Section 4.2.1).

The whole process after finding \mathbf{x}_{nn} is already applicable to both multiclass classification and regression tasks. We only updated the proposed reward functions for an iteration m to

$$R_O(\mathbf{x}) = \frac{o_{\text{valid}}(\hat{f}(\mathbf{x}_{m-1, R_{\max}}), Y') - o_{\text{valid}}(\hat{f}(\mathbf{x}), Y')}{O(\mathbf{x}, \mathbf{x}_{m-1, R_{\max}} | \mathbf{x}^*)}, \quad (9)$$

where $\mathbf{x}_{i-1, R_{\max}}$ is the highest reward instance of the previous iteration ($m-1$), and o_{valid} is defined in Equation 1. The denominator $O(\cdot, \cdot)$ corresponds to the originally proposed functions aiming either at sparsity, proximity, or plausibility.

Although multiple instances could have the desired prediction (and similar reward values), the original *NICE* algorithm only returns a single counterfactual. In the **counterfactuals** package, we implemented two (optional) extensions that enable *NICE* to return multiple counterfactuals. Our first extension returns all created instances (from all iterations) with a desired prediction as counterfactuals after termination. Our second extension does not terminate when the prediction of the highest reward instance is in the desired interval. Instead, it continues until \mathbf{x}_{nn} is recreated. This leads to a total number of $(d^2+d)/2$ created instances, where d is the number of feature values that differ between \mathbf{x}^* and \mathbf{x}_{nn} . Like our first extension, it then returns all created instances with a desired prediction as counterfactuals. Compared to counterfactuals in earlier iterations, a counterfactual created in a later iteration is inferior w.r.t. *Proximity* (ii) and *Sparsity* (iii) (as more feature values are changed), but may be superior w.r.t. *Plausibility* (iv). The pseudocode of our modified *NICE* version is shown in Algorithm 2 of Appendix A.

In contrast to *MOC*, *NICE* does not consider all the desired counterfactual properties (listed in Definition 1) simultaneously: while *NICE* guarantees *Validity* by design (provided that a correctly classified observation with a desired prediction exists), the user must prioritize the other desired properties under the given circumstances and choose the reward function accordingly. If there is no clear preference for the properties *a priori*, we recommend running our second *NICE* extension for each of the reward functions, combining the counterfactuals, removing duplicates, and evaluating the remaining counterfactuals *a posteriori*. We chose this strategy for our benchmark study in Section 6.

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 9

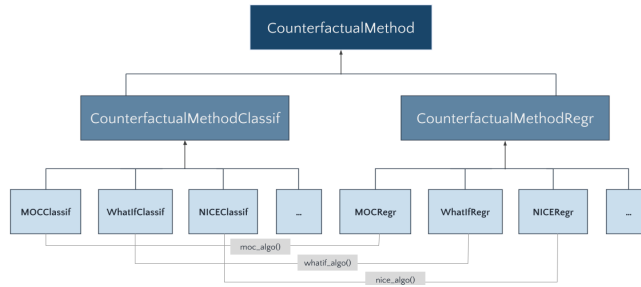


Figure 1: Inheritance diagram of the **counterfactuals** package; a more detailed version is included in Appendix B.1.

A not yet implemented extension is to set lower and upper bounds on \mathbf{x}_{nm} to constrain the feature values of the counterfactuals, enhancing their *Actionability* (v). Another extension would be to run the algorithm multiple times, defining \mathbf{x}_{nm} in the l -th run as the l -th most similar (correctly classified) data point of \mathbf{x}^* , which increases the diversity of the counterfactuals.

3. counterfactuals R package

In this section, we introduce the **counterfactuals** R package and explain its structure and workflow. The package is available from the Comprehensive R Archive Network (CRAN) (Dandl, Hofheinz, Binder, and Casalicchio 2023).

Inspired by the **iml** package (Molnar, Bischl, and Casalicchio 2018), each counterfactual method described in the previous section is implemented in R6 classes (Chang 2021). Datasets and counterfactuals are represented as **data.table** objects (Dowle and Srinivasan 2021) to allow efficient data manipulations and computations. Depending on whether a counterfactual method supports classification or regression tasks, its class inherits from the (abstract) R6 class **CounterfactualMethodClassif** or **CounterfactualMethodRegr** classes, respectively. Counterfactual methods that support both tasks are split into two separate classes. Figure 1 illustrates the inheritance structure. For instance, as *MOC* is applicable to classification and regression tasks, we implemented two classes: **MOCClassif** and **MOCRegr**. Both classes rely on the same (private) code base (`moc_algo()`) to generate counterfactuals to avoid code repetitions. **MOCClassif** inherits features from its superclass **CounterfactualMethodClassif**, while **MOCRegr** inherits from **CounterfactualMethodRegr**. Both of these superclasses in turn have the **CounterfactualMethod** as their superclass.

To generate counterfactuals for an arbitrary model with a specific counterfactual explanation method, the following steps are necessary: First, an `iml::Predictor` object which encapsulates a fitted model and the underlying data must be initialized. The **Predictor** object is a wrapper for any machine learning model and ensures a unified interface and output for model predictions. It offers the necessary flexibility to generate counterfactuals for models fitted with a variety of popular machine learning interfaces (e.g., fitted with the **caret** (Kuhn 2021), **mlr** (Bischl, Lang, Kotthoff, Schiffner, Richter, Studerus, Casalicchio, and Jones 2016), or **mlr3**

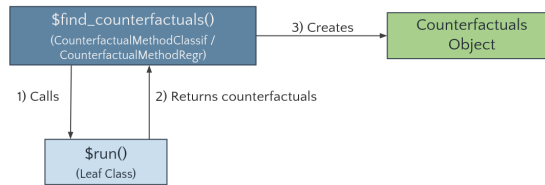


Figure 2: Call graph of the **counterfactuals** package. The `find_counterfactuals()` method (1) calls a private `run()` method – implemented by the leaf classes – which performs the search and (2) returns the counterfactuals as a `data.table`; `find_counterfactuals()` then (3) creates a **Counterfactuals** object, which contains the counterfactuals and provides several methods for their evaluation and visualization.

packages (Lang *et al.* 2019)). We showcase this in the upcoming sections and Appendix B.3. The instantiated **Predictor** object serves as an input for the `predictor` field of the initialization method of the **WhatIfClassif/-Regr**, **MOCClassif/-Regr** or **NICEClassif/-Regr** classes. Additionally, the user can change the parameters of the used methods when initializing the object – such as the mutation probability for *MOC* or the used reward function for *NICE*. Overviews of the parameters are given in Tables 2 - 4 in Appendix B.2.

Counterfactuals are generated by calling the `$find_counterfactuals()` method of the initialized object inherited from the classes **CounterfactualMethodClassif/-Regr**. Figure 2 illustrates the internal call graph. As input, `find_counterfactuals()` requires the observation of interest \mathbf{x}^* for which we seek counterfactuals as well as the desired prediction. The method then calls the `$run()` method, which is implemented in the leaf classes, and creates a **Counterfactuals** object that contains the generated counterfactuals. How the computational burden scales with the number of observations and number of features for the different methods is assessed in Section 6. Several tools are available to visualize and evaluate the counterfactuals. They are showcased and explained in more detail in the upcoming section. These tools are primarily based on the codebase underlying Dandl *et al.* (2020b). More tools will be added in the future.

4. Use cases

In this section, we illustrate the **counterfactuals** workflow by applying *MOC* (Section 2.1) to a classification task and our *NICE* extension (Section 2.3) to a regression task.

4.1. MOC applied to a classification task

As training data, we use the German Credit data set from the **rchallenge** package (Todeschini 2021).² The dataset originally contains 20 features on credit and personal information of 1000 bank customers. For illustrative purposes, we only consider the seven features: `duration`, `amount`, `purpose`, `age`, `employment_duration`, `housing` and `number_credits`. The tar-

²The dataset was originally donated to UCI (Dua and Graff 2017) by Prof. Dr. Hofmann from Universität Hamburg and was later corrected by Grömping (2019).

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 11

get variable `credit_risk` indicates whether a credit is a good/low or bad/high risk for the bank.

```
R> library("counterfactuals")
R> library("iml")
R> library("randomForest")
R> data("german", package = "rchallenge")
R> credit = german[, c("duration", "amount", "purpose", "age",
+   "employment_duration", "housing", "number_credits", "credit_risk")]
```

We train a random forest with the `randomForest` package to predict the `credit_risk` (Liaw and Wiener 2002). We omit observation 998 from the training data, which is x^* , to imitate the situation of finding counterfactuals for a new observation.³

```
R> set.seed(20210816)
R> rf = randomForest(credit_risk ~ ., data = credit[-998L,])
```

An `iml::Predictor` object serves as a wrapper for different model types. It contains the model and the data for its analysis. We set `type = "prob"` such that class probabilities instead of hard labels are predicted. For our observation of interest x^* – denoted in the code as `x_interest` – the model predicts a probability of being a good credit risk of 38.2%:

```
R> predictor = iml::Predictor$new(rf, type = "prob")
R> x_interest = credit[998L, ]
R> predictor$predict(x_interest)
```

```
##      bad  good
## 1 0.618 0.382
```

Generation of counterfactuals

Now, we examine which risk factors must be changed to increase the predicted probability of being a good credit risk to at least 60%. Since we want to apply *MOC* to a classification model, we initialize a `MOCClassif` object. As explained in Section 2.1, individuals whose prediction is farther away from the desired interval than a prespecified value `epsilon` can be penalized. Here, we set `epsilon = 0` to penalize all individuals whose prediction is outside the desired interval. With the `fixed_features` argument, we fix the non-actionable features `age` and `employment_duration` to the respective value of x^* . By setting the termination criterion to `genstag`, we stop once the HV indicator does not increase for `n_generations = 10L` consecutive generations.

```
R> moc_classif = MOCClassif$new(
+   predictor, epsilon = 0, fixed_features = c("age", "employment_duration"),
+   termination_crit = "genstag", n_generations = 10L)
```

³This does not rule out the possibility to generate counterfactuals for training data points.

We use the `$find_counterfactuals()` method to search for counterfactuals for `x_interest`. As we aim to find counterfactuals with a predicted probability of being a good credit risk of at least 60%, we set the `desired_class` to "good" and the `predicted_prob` to `c(0.6, 1)`; this is equivalent to setting the `desired_class` to "bad" and `desired_prob` to `c(0, 0.4)`.

```
R> cfactuals = moc_classif$find_counterfactuals(
+   x_interest, desired_class = "good", desired_prob = c(0.6, 1))
```

The Counterfactuals object

The resulting `Counterfactuals` object holds the counterfactuals in the `data` field and possesses several methods for their evaluation and visualization. Printing a `Counterfactuals` object gives an overview of the results. Overall, we generated 82 counterfactuals.

```
R> print(cfactuals))

## 82 Counterfactual(s)
##
## Desired class: good
## Desired predicted probability range: [0.6, 1]
##
## Head:
##   duration amount purpose age employment_duration housing number_credits
## 1:      21   7460  others  30              >= 7 yrs      own           1
## 2:      21   7054  others  30              >= 7 yrs      own           1
## 3:      21   6435  others  30              >= 7 yrs      own           1
```

The `$predict()` method returns the predictions for the counterfactuals.

```
R> head(cfactuals$predict(), 3L)

##      bad good
## 1: 0.322 0.678
## 2: 0.318 0.682
## 3: 0.296 0.704
```

The `$evaluate()` method returns the counterfactuals along with some predefined quality measures `dist_x_interest`, `no_changed`, `dist_train`, and `dist_target` for the desired properties *Proximity*, *Sparsity*, *Plausibility*, and *Validity* (listed in Definition 1). The quality measures are equal to the objectives of *MOC*. Setting the `show_diff` argument to `TRUE` displays the counterfactuals as their difference from `x_interest`: for a numeric feature, positive values indicate an increase compared to the feature value in `x_interest` and negative values indicate a decrease; for factors, the feature value is displayed if it differs from `x_interest`; NA means "no difference".

```
R> head(cfactuals$evaluate(show_diff = TRUE, measures = c("dist_x_interest",
+   "dist_target", "no_changed", "dist_train")), 3L)
```

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 13

```
## duration amount purpose age employment_duration housing number_credits
## 1:      NA  -5220   <NA> NA                        <NA>   <NA>           <NA>
## 2:      NA  -5626   <NA> NA                        <NA>   <NA>           <NA>
## 3:      NA  -6245   <NA> NA                        <NA>   <NA>           <NA>
## dist_x_interest no_changed dist_train dist_target
## 1:      0.04103193          1 0.04215022          0
## 2:      0.04422330          1 0.03895885          0
## 3:      0.04908897          1 0.03409318          0
```

By design, there is no guarantee that all counterfactuals generated with MOC have a prediction $\in Y'$. Therefore, we use the `$subset_to_valid()` method to omit all non-valid counterfactuals. The method `$revert_subset_to_valid()` can reverse this step.

```
R> cfactuals$subset_to_valid()
R> nrow(cfactuals$data)
## [1] 40
```

Of the 82 counterfactuals, 40 have the desired predictions. To detect which features are the most important levers to obtain a certain prediction, the relative frequency of feature changes across all counterfactuals can be plotted via the `$plot_freq_of_feature_changes()` method. Setting `subset_zero = TRUE` excludes all unchanged features from the plot. Figure 3 shows that all counterfactuals require changes in the credit amount.

```
R> cfactuals$plot_freq_of_feature_changes(subset_zero = TRUE)
```

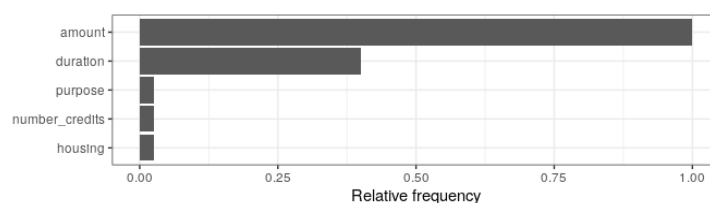


Figure 3: Relative frequency of feature changes across all counterfactuals. Features without proposed changes are omitted.

The parallel plot (Figure 4) – created with the `$plot_parallel()` method – compares the feature values of the counterfactuals among each other (one gray line per counterfactual) and with `x_interest` (blue line). Equal to Dandl *et al.* (2020b), all features are scaled between 0 and 1. The argument `feature_names` filters the features and orders them, NULL means “all”. Using `$get_freq_of_feature_changes()`, we order the features according to their frequency of changes. The `digits_min_max` argument specifies the maximum number of digits for plotted values. The default value is 2L. All counterfactuals propose a decrease in the credit amount while the duration either needs no modifications, an increase or a decrease. For one counterfactual, additionally the purpose was set to a new car, the housing type was set to rented and the number_credits was increased.

```
R> cfactuals$plot_parallel(feature_names = names(
+   cfactuals$get_freq_of_feature_changes()), digits_min_max = 2L)
```

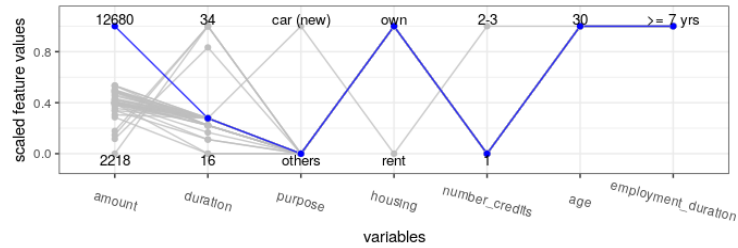


Figure 4: Parallel plot along (standardized) features. The blue line represents \mathbf{x}^* ($\mathbf{x}_{\text{interest}}$), whereas gray lines represent generated counterfactuals.

The `$plot_surface()` method generates prediction surface plots/2-dimensional ICE plots (Dandl *et al.* 2020b). The method requires the names of two features (argument `feature_names`) as an input. The white dot in Figure 5 represents $\mathbf{x}_{\text{interest}}$. All counterfactuals that differ from $\mathbf{x}_{\text{interest}}$ *only* in the two selected features (here, `duration` and `amount`) are displayed as black dots. We observe that either a change in `amount` alone, or in `amount` *and* the `duration` is advocated. The rug lines next to the axes indicate the marginal distribution of the training data. It should be noted that the multi-objective approach does not consider counterfactuals farther away from $\mathbf{x}_{\text{interest}}$ as suboptimal because these counterfactuals outperform others in their proximity to the observed data points (plausibility property (iv)).

```
R> cfactuals$plot_surface(feature_names = c("duration", "amount"))
```

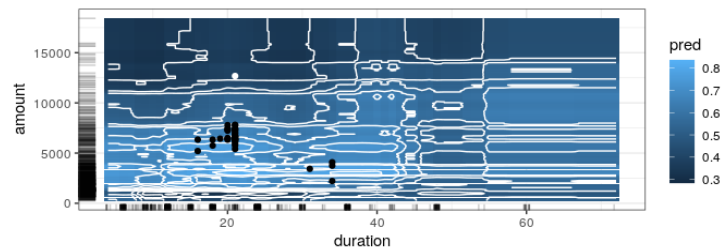


Figure 5: Prediction surface plotted along features `duration` and `amount`. Other feature values are held constant at \mathbf{x}^* . The white point displays \mathbf{x}^* . Black points are counterfactuals with variations only in the two displayed features. Rugs represent marginal distributions of the observed data.

MOC diagnostics

The aforementioned plotting and evaluation methods are part of the class `Counterfactuals`

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 15

and all counterfactuals created by *MOC*, *WhatIf*, or *NICE* can be evaluated with them. For *MOC*, additional diagnostic tools are available. Since they are only applicable to *MOC*, they cannot be called by the `Counterfactuals` class but rather by instances from the `MOCClassif` and `MOCRegr` class after counterfactuals were generated. To evaluate the estimated Pareto front, Dandl *et al.* (2020b) use a HV indicator (Zitzler and Thiele 1998) with reference point $s = (\inf_{y' \in Y'} |f(\mathbf{x}^*) - y'|, 1, p, 1)$ representing the maximal values of the objectives (o_{valid} , o_{prox} , o_{sparse} , o_{plaus} of Equations 1 to 4). The evolution of the HV indicator can be plotted together with the evolution of mean and minimum objective values using the `$plot_statistics()` method. The `centered_obj` argument allows the user to control whether the objective values should be centered: if set to `FALSE`, each objective value is visualized in a separate plot, since they (usually) have different scales; if set to `TRUE` (default), they are visualized in a single plot, as shown in Figure 6.

```
R> moc_classif$plot_statistics(centered_obj = TRUE)
```

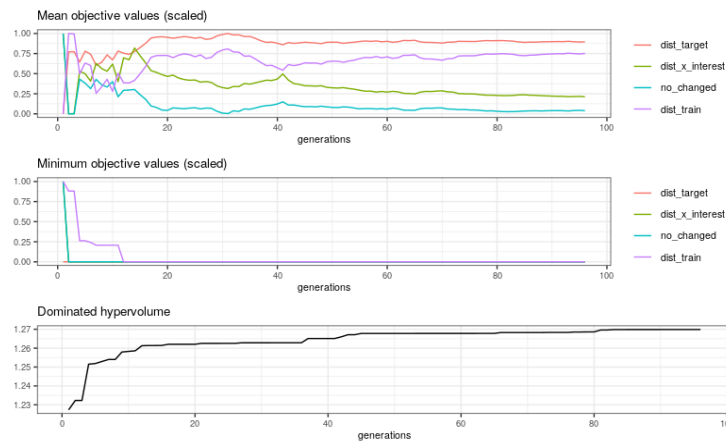


Figure 6: Evolution of the mean and minimum objective values together with the dominated HV over the generations. The mean and minimum objective values were scaled between 0 and 1.

Ideally, the mean value of each objective decreases, while the HV increases over the generations. However, there is often a trade-off between the objectives in the sense that when the mean value of one objective slightly decreases, it might slightly increase for another objective. This trade-off is also visible in the scatter plot created with the `$plot_search()` method that visualizes the values of two specified `objectives` of all emerged individuals. Ideally, one would like to have a point shift to the lower-left corner over the generations, which implies lower and thus better objective values.

```
R> moc_classif$plot_search(objectives = c("dist_train", "dist_target"))
```

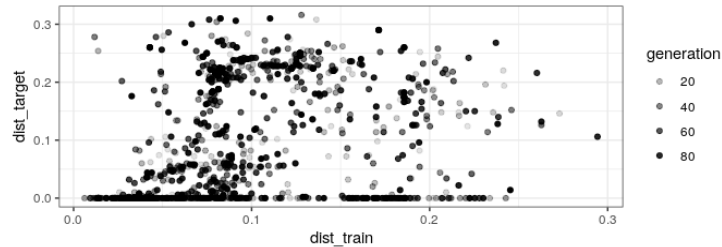



Figure 7: Evolution of the objectives `dist_train` and `dist_target` over the generations.

According to Figure 7, many counterfactual have predictions in the desired prediction range (`dist_target = 0`). However, many points for the objectives `dist_train` and `dist_target` are also located in the middle region. This underlines the difficulty of minimizing both objectives simultaneously. For the objectives `dist_train` and `dist_x_interest` (Figure 8) (Figure 8), on the other hand, there is a clearer shift to the lower-left corner over the generations. The distinct boundary on the lower left indicates that the optimization potential for these two objectives might be fully exploited.

```
R> moc_classif$plot_search(objectives = c("dist_x_interest", "dist_train"))
```

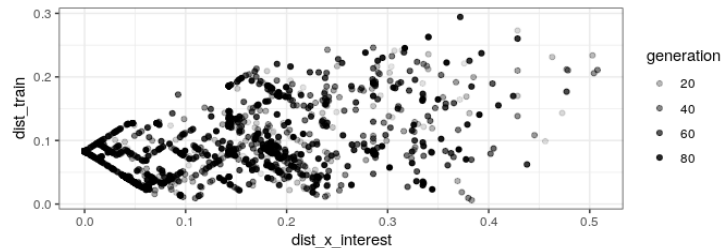


Figure 8: Evolution of the objectives `dist_x_interest` and `dist_train` over the generations.

4.2. NICE applied to a regression task

Searching for counterfactuals for regression models works analogously to classification models. In this example, we use our *NICE* extension for regression models to search for multiple counterfactuals for a predictor of plasma retinol concentration. This is interesting because low concentrations are associated with an increased risk for some types of cancer (see Xie, Song, Lin, Guo, Wang, Tang, Liu, Huang, Yang, Ling, and et al. (2019) for an overview).

As training data, we use the plasma dataset (Harrison Jr and Rubinfeld 1978) from the `gamlss.data` package (Stasinopoulos, Rigby, and De Bastiani 2021). The dataset contains 315 observations with 13 features describing personal and dietary factors (e.g., age, number of alcoholic drinks per week or the measured plasma beta-carotene level) and the (continuous)

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 17

target variable `retplasma` – the plasma retinol concentration in ng/ml. We train a regression tree with the `mlr3` package to predict `retplasma` (Lang *et al.* 2019). We reserve the 100th row of the data for x^* – denoted as `x_interest`.

```
R> library("mlr3")
R> data("plasma", package = "gamlss.data")
R> x_interest = plasma[100L,]
R> tsk = mlr3::TaskRegr$new(id = "plasma", backend = plasma[-100L,],
+   target = "retplasma")
R> tree = lrn("regr.rpart")
R> model = tree$train(tsk)
```

Then, we initialize an `iml::Predictor` object. For `x_interest`, the model predicts a plasma concentration of 342.92 ng/ml.

```
R> predictor = Predictor$new(model, data = plasma, y = "retplasma")
R> predictor$predict(x_interest)
```

```
##   pred
## 1 342.92
```

Since we want to apply *NICE* to a regression model, we initialize a `NICERegr` object. The initial version of *NICE* restricted to classification models starts the search by finding the most similar correctly classified datapoint. For regression models, we define a correctly predicted datapoint when its prediction is less than a user-specified value (`margin_correct`) away from the true outcome. In this example, we allow for a deviation of 0.5. The argument `optimization` specifies the reward function we want to optimize. We aim for the most proximal counterfactual by setting this argument to `proximal` and by setting `return_multiple` to `FALSE`.

We call the `$find_counterfactuals()` method to search for counterfactuals for `x_interest` with a predicted concentration of more than 500 ng/ml, i.e. a concentration in the interval $[500, Inf]$.

```
R> nice_regr = NICERegr$new(predictor, optimization = "proximity",
+   margin_correct = 0.5, return_multiple = FALSE)
R> cfactuals = nice_regr$find_counterfactuals(x_interest,
+   desired_outcome = c(500, Inf))
```

The result is a `Counterfactuals` object, which we can analyze with the same methods as in Section 4.1.2. The surface plot of plasma beta-carotene (`betaplasma`) and age (Figure 9), for example, reveals that increasing the beta-carotene concentration (e.g., by eating more kale, carrots, etc.) is sufficient for predicting a plasma concentration ≥ 500 ng/ml for x^* , while changing the age alone has no effect on the prediction.

```
R> cfactuals$plot_surface(feature_names = c("betaplasma", "age"), grid_size = 200)
```

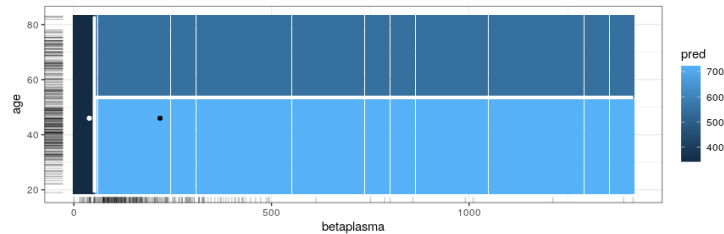


Figure 9: Prediction surface plotted along features `betaplasma` and `age`. Other feature values are held constant at \mathbf{x}^* . The white point displays \mathbf{x}^* . Black points are counterfactuals with variations only in the two displayed features. Rugs represent marginal distributions of the observed data. White horizontal lines are plotting artifacts.

User-defined distance function

As stated in Equation 8, *NICE* determines the most similar (correctly classified) datapoint by minimizing the Gower distance. However, the input parameter `distance_measure` of the initialization method of `NICERegr` (and `NICEClassif`) allows a different distance measure. The parameter requires a function with arguments `x`, `y`, and `data`, that returns a numeric matrix with number of rows and columns corresponding to the number of observations in `x` and `y`, respectively. As an example, we replace the Gower function with the L_0 norm. First, we set up the function and illustrate its functionality in a short example.

```
R> l0_norm = function(x, y, data) {
+   res = matrix(NA, nrow = nrow(x), ncol = nrow(y))
+   for (i in seq_len(nrow(x))) {
+     for (j in seq_len(nrow(y))) {
+       res[i, j] = sum(x[i,] != y[j,])
+     }
+   }
+   res
+ }
R> xt = data.frame(a = c(0.5), b = c("a"))
R> yt = data.frame(a = c(0.5, 3.2, 0.1), b = c("a", "b", "a"))
R> l0_norm(xt, yt, data = NULL)

##      [,1] [,2] [,3]
## [1,]    0    2    1
```

Next, we forward this function to the `distance_function` argument of `NICERegr`.

```
R> nice_regr = NICERegr$new(predictor, optimization = "proximity",
+   margin_correct = 0.5, return_multiple = FALSE,
+   distance_function = l0_norm)
R> nice_regr$find_counterfactuals(x_interest, desired_outcome = c(500, Inf))
```

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 19

```
## 1 Counterfactual(s)
##
## Desired outcome range: [500, Inf]
##
## Head:
##   age sex smokstat   bmi vituse calories   fat fiber alcohol cholesterol
## 1:  46  1         3 35.26     3  2667.5 131.6 10.1       0       550.5
##   betadiet retdiet betaplasma
## 1:    1210    1291        218
```

The initialization methods of *MOC* and *WhatIf* also have a `distance_function` argument: for *MOC*, its input replaces the Gower distances used for o_{prox} and o_{plaus} (Equations 2 & 4); for *WhatIf*, its input replaces the Gower distance in Equation 7.

5. Extension of the package

We have designed the **counterfactuals** package to be quickly extensible by new methods. Here, we illustrate how to add new methods to the package by integrating the **featureTweakR** package (Kato 2018), which implements Feature Tweaking (Tolomei *et al.* 2017), a counterfactual method that can be applied to (classification) tree ensembles fitted with the **randomForest** package. Feature Tweaking starts the search for counterfactuals for an observation \mathbf{x}^* by finding all trees in the ensemble that do not predict the desired class. For each of these trees, it attempts to change (or “tweak”) \mathbf{x}^* as little as possible to switch the prediction of that tree to the desired class. From all tweaked instances that also switch the ensemble prediction to the desired class, it returns the tweaked instance that changes \mathbf{x}^* the least as a counterfactual.

The **featureTweakR** package has a couple of limitations, e.g., factors in the training data cause problems or that it is only applicable to random forests trained on standardized features with the **randomForest** package (Liaw and Wiener 2002). Due to these limitations, **featureTweakR** is not part of the **counterfactuals** package but does serve as a suitable example here. First, we install **featureTweakR** and its dependency **pfforeach** (Makiyama 2015) and load the required libraries.

```
R> devtools::install_github("katokohaku/featureTweakR")
R> devtools::install_github("hoxo-m/pforeach")
R> library("featureTweakR")
R> library("counterfactuals")
R> library("iml")
R> library("randomForest")
R> library("R6")
```

5.1. Class structure

At least two methods must be implemented for a new class: `$initialize()` and `$run()`. The `$print_parameters()` method is not mandatory but still strongly recommended, as it gives objects of that class an informative `print()` output. As elaborated above, a new

class inherits from either `CounterfactualMethodClassif` or `CounterfactualMethodRegr`, depending on which task it supports. Since Feature Tweaking supports classification tasks, the new `FeatureTweakerClassif` class inherits from the former.

```
R> FeatureTweakerClassif = R6::R6Class("FeatureTweakerClassif",
+   inherit = CounterfactualMethodClassif,
+   public = list(
+     initialize = function() {
+       # **see below**
+     }
+   ),
+   private = list(
+     run = function() {
+       # **see below**
+     },
+     print_parameters = function() {
+       # **see below**
+     }
+   )
+ )
```

Implementation of the `$initialize()` method

In the next step, we implement the `$initialize()` method, which must have a `predictor` argument that takes an `iml::Predictor` object. In addition, it may have further arguments specific to the counterfactual method. Feature Tweaking has the following hyperparameters: `ktree` representing the number of trees to be considered, `epsilon`⁴ as the upper threshold of feature changes, and `resample` indicating whether trees are randomly selected or not.

```
R> initialize = function(predictor, ktree = NULL, epsilon = 0.1,
+   resample = FALSE) {
+   # adds predictor to private$predictor field
+   super$initialize(predictor)
+   private$ktree = ktree
+   private$epsilon = epsilon
+   private$resample = resample
+ }
```

We also fill the `$print_parameters()` method with the parameters of Feature Tweaking.

```
R> print_parameters = function() {
+   cat(" - epsilon: ", private$epsilon, "\n")
+   cat(" - ktree: ", private$ktree, "\n")
+   cat(" - resample: ", private$resample)
+ }
```

⁴Please note that this is not a typo on our part, but the naming in the original implementation (Kato 2018).

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 21

Implementation of the `$run()` method

The `$run()` method performs the search for counterfactuals. Its structure is completely free, which makes it flexible to add new counterfactual methods to the `counterfactuals` package. The only requirement is that a `data.table` with the generated counterfactuals is returned at the end. The columns display the features and rows the counterfactuals.

The `$run()` method is called by the method `$find_counterfactuals()` implemented in the `CounterfactualMethodsClassif` class. As shown in Section 4.1, `$find_counterfactuals` requires as input `x_interest`, `desired_class`, and `desired_prob`, which are saved in private fields. Thus, `$run()` could directly access the information and preprocesses them before it passes them on to the implemented methods of `featureTweakR`.

The workflow of finding counterfactuals for `x_interest` with the `featureTweakR` package for a fitted random forest model `rf` consists of three steps: First, decision trees are transformed to data frames of paths by `getRules()`. Then, `set.eSatisfactory()` generates new instances by slightly altering feature values. Finally, `tweak()` generates counterfactuals for a specific instance x^* . Further information could be found in the documentation of the package (Kato 2018). The `$run()` method encapsulates these steps and returns a `data.frame` of generated counterfactuals.

```
R> run = function() {
+   # Extract info from private fields
+   predictor = private$predictor
+   y_hat_interest = predictor$predict(private$x_interest)
+   class_x_interest = names(y_hat_interest)[which.max(y_hat_interest)]
+   rf = predictor$model
+   # Call functions in featureTweakR
+   rules = getRules(rf, ktree = private$ktree, resample = private$resample)
+   es = set.eSatisfactory(rules, epsilon = private$epsilon)
+   tweaks = tweak(
+     es, rf, private$x_interest, label.from = class_x_interest,
+     label.to = private$desired_class, .dopar = FALSE
+   )
+   return(tweaks$suggest)
+ }
```

The composite code of our new class can be seen in Appendix B.4.

5.2. Feature Tweaking applied to a classification task

For demonstration purposes, we apply the implemented Feature Tweaking to the `iris` dataset (Fisher 1936; Anderson 1936). We train a random forest on the dataset and set up the `iml::Predictor` object, again omitting `x_interest` (here, row 130) from the training data.

```
R> set.seed(78546)
R> X = subset(iris, select = -Species)[-130L,]
R> y = iris$Species[-130L]
R> rf = randomForest(X, y, ntree = 20L)
```

```
R> predictor = iml::Predictor$new(rf, data = iris[-130L, ],
+ y = "Species", type = "prob")
```

For `x_interest`, the model predicts a probability of 30% for `versicolor`.

```
R> x_interest = iris[130L, ]
R> predictor$predict(x_interest)
```

```
## setosa versicolor virginica
## 1      0      0.3      0.7
```

Now, we use Feature Tweaking to address the question: “What changes in `x_interest` are necessary for the model to predict a probability of at least 60% for `versicolor`?”.

```
R> # Set up FeatureTweakerClassif
R> ft_classif = FeatureTweakerClassif$new(predictor, ktree = 10L,
+ resample = TRUE)
R> # Find counterfactuals and create a Counterfactuals object
R> cfactuals = ft_classif$find_counterfactuals(
+ x_interest, desired_class = "versicolor", desired_prob = c(0.6, 1)
+ )
```

As for *MOC* and *NICE*, the result is a `Counterfactuals` object which could be visualized and evaluated as shown in Section 4.1.2.

6. Benchmarking

In this section, we use a benchmark study to answer the following research questions:

1. How do the different methods implemented in the **counterfactuals** R package perform according to the properties validity (i), proximity (ii), sparsity (iii) and plausibility (iv) of Definition 1, and according to the HV indicator and number of non-dominated counterfactuals?
2. How do the methods differ in their runtime for an increasing number of observations (n) and number of features (p)?

The overall design of our benchmark study is strongly inspired by the work of Dandl *et al.* (2020b) who also compared different methods according to the four properties of Definition 1. Additionally, we evaluate the methods with regard to their runtime behavior and HV. Furthermore, we added *NICE* as another comparison method. Since our source code is openly available⁵, we encourage readers to add other counterfactual methods to our R package and to compare them to the already implemented ones using our study code.

⁵https://github.com/slds-lmu/benchmark_2022_counterfactuals

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 23

OpenML ID	Name	Obs	Cont	Cat
31	credit_g	1,000	7	13
37	diabetes	768	8	0
50	tic_tac_toe	958	0	9
725	bank8FM	8,192	8	0
1479	hill_valley	1,212	100	0
40922	run_or_walk_information	88,588	6	0

Table 1: Description of the OpenML datasets used for benchmarking. Obs displays the no. of observations, Cont the no. of continuous features and Cat the no. of categorical features.

6.1. Setup

We used six datasets from the OpenML platform (Vanschoren, van Rijn, Bischl, and Torgo 2014) with binary classes, no missing values, and varying numbers of observations and features. Table 1 provides an overview of the datasets. To study the runtime behavior, we also ran all available methods on row-wise subsets (with differing number of observations $n \in \{886 (1\%), 8859 (10\%), 88588 (100\%)\}$) of the `run_or_walk_information` dataset and column-wise subsets (with differing number of features $p \in \{10, 30, 100\}$) of the `hill_valley` dataset. The subsets were randomly generated and identical for all models and methods.

On each dataset, we tuned and trained five models using the `mlr3` R package (Lang et al. 2019): a random forest (`ranger`), an `xgboost`, an RBF support vector machine (`svm`), a logistic regression (`logreg`), and a neural network with one hidden layer (`neuralnet`).⁶ Beforehand, we standardized numerical features and one-hot-encoded categorical ones. For tuning, we employed random search with 30 evaluations and 5-fold cross-validation (CV) using the misclassification error as a performance measure. Further details on the tuning search space and the classification accuracies are given in Appendix C.1. Before training, we randomly selected ten observations from each dataset as \mathbf{x}^* and omitted them from the training data. For each \mathbf{x}^* , we set the desired class probability interval Y' to the opposite of the predicted class (based on a threshold of 0.5):

$$Y' = \begin{cases}]0.5, 1] & \text{if } f(\mathbf{x}^*) \leq 0.5 \\ [0, 0.5] & \text{else} \end{cases} . \quad (10)$$

For each dataset, model, and \mathbf{x}^* , we computed counterfactuals with *WhatIf*, *NICE* and *MOC*. Apart from the stopping criterion, all *MOC* control parameters were set to their default values selected through iterated F-racing (López-Ibáñez, Dubois-Lacoste, Cáceres, Birattari, and Stützle 2016) (see Appendix B.2). Notably, we used different datasets for tuning than for the benchmark study. The stopping criterion was convergence of the HV over 10 generations, with a total maximum of 500 generations. For all three counterfactual methods, we set the `distance_function` to `'gower_c'` – a C-based, more efficient version of Gower’s distance based on the `gower` R package (Van der Loo 2022).

As stated in Section 2, we prefer a set of counterfactuals over a single one. *MOC* is designed to return multiple counterfactuals and we also let *NICE* and *WhatIf* return multiple ones. Therefore, the *NICE* control parameter `finish_early` was set to `FALSE`, corresponding to

⁶For the `hill_valley` dataset with 100 features, two dense layers were necessary.

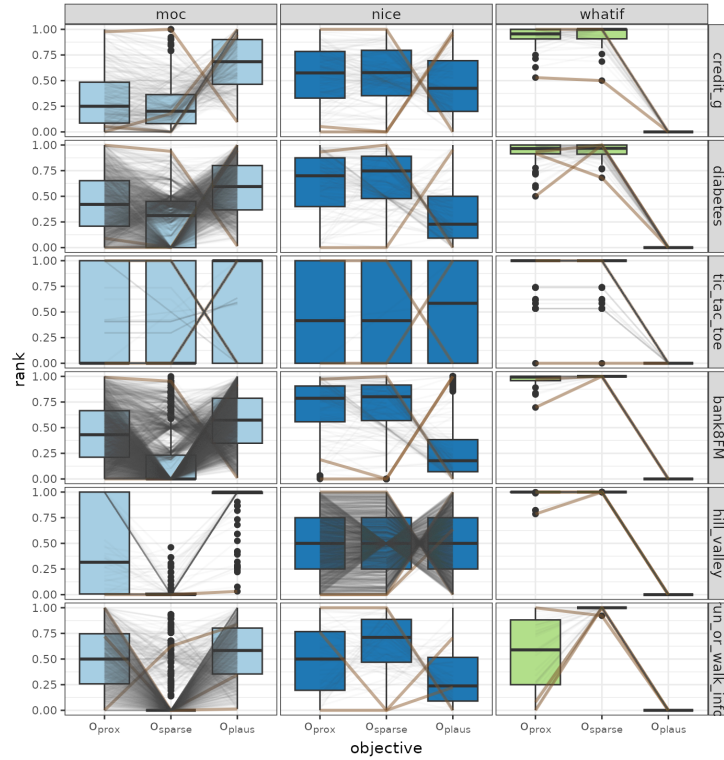


Figure 10: Comparison of *NICE*, *WhatIf*, and *MOC* w.r.t. their rank in the properties *Proximity* (ii, o_{prox}), *Sparsity* (iii, o_{spars}) and *Plausibility* (iv, o_{plaus}). Each gray line reflects a counterfactual (for clarity purposes, only a maximum of 2000 counterfactuals are displayed). The counterfactuals with the lowest and therefore best rank in an objective display the brown lines. Lower values are better.

our second *NICE* extension (Section 2.3). In addition, we computed counterfactuals for each of the three different reward functions by varying the `optimization` hyperparameter and combined them for a final set of counterfactuals, as recommended in Section 2.3. For *WhatIf*, the number of counterfactual was set to 10 via the `n_counterfactuals` parameter, in accordance with Dandl *et al.* (2020b). All other *NICE* and *WhatIf* control parameters (except the `distance_function`, see above) were set to their default values (Appendix B.2).

For the evaluation, we only considered the counterfactuals that (1) achieve the desired prediction such that $o_{\text{valid}} = 0$ and (2) are not dominated by other counterfactuals produced by the same method according to the remaining three objectives (o_{prox} , o_{spars} and o_{plaus}). By design of the three methods, criterion (1) always holds for counterfactuals of *WhatIf* and *NICE* and (2) always for *MOC*.

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 25

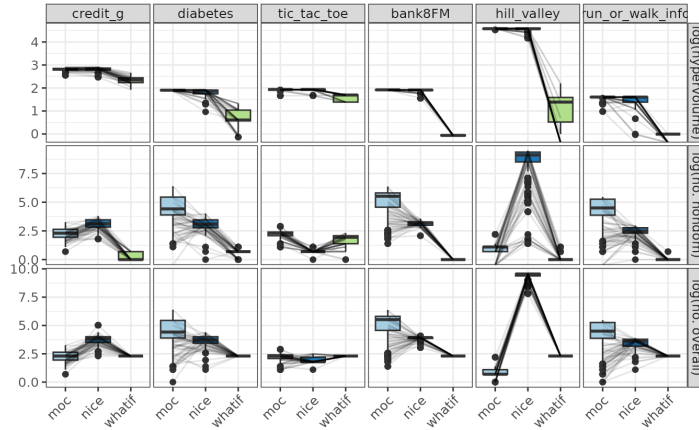


Figure 11: Comparison of *NICE*, *WhatIf*, and *MOC* w.r.t. their HV, the number of non-dominated and valid counterfactuals (*no. nondom*) and the number of all returned counterfactuals (*no. overall*). The values were logarithmized. Higher values are better.

For Research Question 1, we evaluated the generated counterfactuals by means of the desired properties stated in Definition 1: *Validity* (i, o_{valid}), *Proximity* (ii, o_{prox}), *Sparsity* (iii, o_{spars}) and *Plausibility* (iv, o_{plaus}). We ranked all counterfactuals per dataset, model, and \mathbf{x}^* by their values in the desired properties, normalized the ranks between 0 and 1, and compared the normalized ranks between the methods. The ranking ensures that counterfactuals are comparable over all datasets and models. To take into account all three properties at once, we also computed the HV indicator, which measures the HV in the objective space between the non-dominated counterfactuals and a (worst-case) reference point (1 for o_{prox} , no. features for o_{spars} and 1 for o_{plaus}). For Research Question 2, we tracked the runtime behavior for all methods in generating counterfactuals for (row-wise or colum-wise subsets of) the *run_or_walk_information* and *hill_valley* datasets.

6.2. Results

In the following, we present the results for the two stated research questions.

Research Question 1

Figure 10 compares the ranking of counterfactuals according to the desired properties for *MOC*, *NICE* and *WhatIf* for each dataset separately. Figure 14 in the Appendix does the same for each model separately. Since our setup ensured that all compared counterfactuals achieved the desired prediction, we omitted the results for the first property *Validity* (i, o_{valid}). Each gray line reflects a counterfactual. The counterfactuals with the lowest and therefore best rank in one of the three remaining objectives display the brown lines. Appendix C.2 shows the results on the property instead of the raking scale for each model and dataset

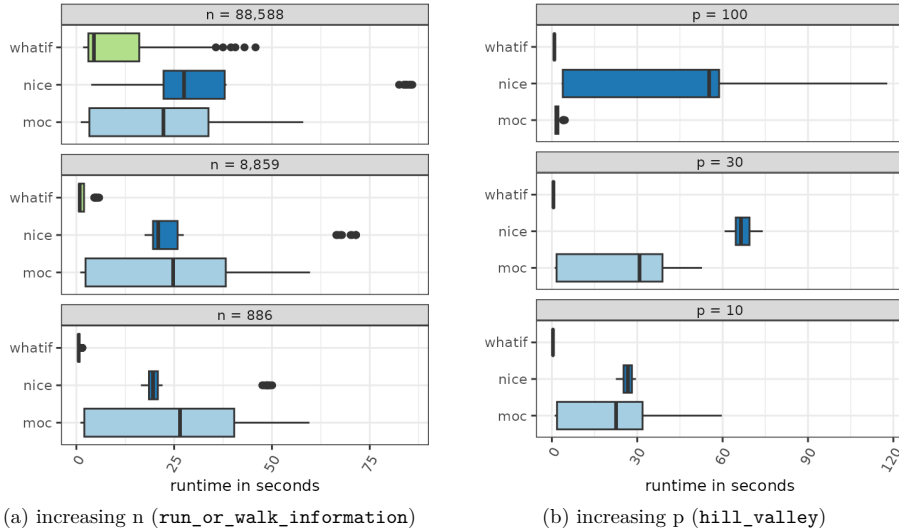


Figure 12: Speed comparison of *NICE*, *WhatIf*, and *MOC* based on row-wise subsets of the `run_or_walk_information` dataset and column-wise subsets of the `hill_valley` dataset. The runtimes of *NICE* were aggregated for its three reward function configurations.

separately. They agree with the results shown here.

WhatIf's counterfactuals changed on average more features (o_{spars}) and had the highest distances to \mathbf{x}^* (o_{prox}), making *WhatIf* inferior to the other methods w.r.t. the desired counterfactual properties *Sparsity* (iii) and *Proximity* (ii). However, its counterfactuals have low training data distances (o_{plaus}) by design, guaranteeing *Plausibility* (iv).

Compared with *MOC*, the counterfactuals of *NICE* on average changed more features and had often a higher distance to \mathbf{x}^* , indicating that *NICE* was overall inferior to *MOC* w.r.t. *Sparsity* and *Proximity*. However, on average, the counterfactuals of *NICE* had lower training data distances (measuring *Plausibility*) than *MOC*'s counterfactuals.

Figure 11, displays the HV, the number of non-dominated, valid counterfactuals, and the overall number of returned counterfactuals (including dominated and/or non-valid ones) on the log scale for each dataset and method. Overall, *MOC*'s counterfactuals achieved the highest HV closely followed by *NICE*, indicating that *MOC* is slightly superior when considering all objectives simultaneously. The HV of *WhatIf*'s counterfactuals is comparably low except for the `tic_tac_toe` dataset with a low number of categorical features. While all counterfactuals of *MOC* are (by design) non-dominated by other counterfactuals returned by the method, many of the counterfactuals of *NICE* or *WhatIf* are dominated by others generated by the same method. Apart from the `tic_tac_toe` dataset, *WhatIf* produced the least non-dominated counterfactuals. *MOC* generated the most non-dominated counterfactuals except for the `credit_g` and `hill_valley` datasets.

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 27

Research Question 2

Figure 12 compares the runtimes of our extended *WhatIf* and *NICE* versions with *MOC*. *WhatIf* was the fastest and best scaling method. *NICE* ran on average 17 times longer than *MOC* for high p and almost 1.6 times longer for high n . This is because for the `hill_valley` dataset with $p = 100$ features, the method at worst needs to evaluate $(p^2 + p)/2 = 5050$ observations for each of the three reward functions. For low p the differences diminished between *NICE* and *MOC*. For low n , *NICE* was on average even faster than *MOC*.

6.3. Discussion

In the following, we briefly discuss the suitability of each method for different scenarios based on the results of our benchmark study. *MOC* returned on average the most non-dominated counterfactuals of highest-quality when considering all desired properties simultaneously. Our extended *NICE* version had comparatively high runtimes for a medium to high number of features. *WhatIf* was the fastest method, but (by design) its counterfactuals suggested changes to many features, impeding the interpretation. The method is suitable in time-critical scenarios for datasets with a few categorical features.

7. Conclusion

In this work, we introduced the **counterfactuals** R package, which to the best of our knowledge is the first R package that provides several counterfactual methods via a unified interface. The package includes the method *MOC* as well as extended versions of *WhatIf* and *NICE*, which are all capable of returning multiple counterfactuals for regression and (binary and multiclass) classification models. In addition, we illustrated that the **counterfactuals** package is quickly extensible with new methods. This is crucial, as the variety of counterfactual methods proposed in research is growing rapidly, but the number of implemented methods in R is very limited. Furthermore, the package offers a variety of functionalities for evaluating and visualizing the counterfactuals. Thus, our package facilitates the application of counterfactual methods in practice for auditing machine learning models.

The results of our benchmark study and other research (e.g., Verma *et al.* 2022) suggest that no existing counterfactual method is superior in all situations. This underlines the benefit of the **counterfactuals** package, which makes a variety of methods readily available to the user. Furthermore, the object-oriented concept of our package and the openly available benchmark code allows new methods to easily compete with those currently available.

Computational details

The results in this work were obtained using R 4.2.2 R Core Team (2022). R itself and most of the packages used are available from CRAN – including the **counterfactuals** R package (Dandl *et al.* 2023). We included all data examples of Sections 4 and 5 in dedicated vignettes. To facilitate full reproducibility of the benchmark study of Section 6, we created a dedicated Github repository: https://github.com/slds-lmu/benchmark_2022_counterfactuals. The experiments were run in parallel with the help of the **batchtools** package (Lang, Bischl, and Surmann 2017) on a computer with a 2.60 GHz Intel(R) Xeon(R) processor, and 32 CPUs.

Training (incl. tuning) the models took 53 hours spread over 15 CPUs, generating the counterfactuals took 37 hours spread over 14 CPUs.

Acknowledgments

This work has been partially supported by the Federal Statistical Office of Germany.

References

- Anderson E (1936). “The Species Problem in Iris.” *Annals of the Missouri Botanical Garden*, **23**(3), 457–509. doi:10.2307/2394164.
- Biecek P (2018). “DALEX: Explainers for Complex Predictive Models in R.” *Journal of Machine Learning Research*, **19**(84), 1–5. URL <https://jmlr.org/papers/v19/18-416.html>.
- Binder M (2023). *miesmuschel: Mixed Integer Evolution Strategies*. R package version 0.0.3, URL <https://CRAN.R-project.org/package=miesmuschel>.
- Binder M, Dandl S, Moosbauer J (2020). *mosmafs: Multi-Objective Simultaneous Model and Feature Selection*. R package version 0.1.2, URL <https://CRAN.R-project.org/package=mosmafs>.
- Binder M, Pfisterer F, Lang M, Schneider L, Kotthoff L, Bischl B (2021). “**mlr3pipelines** - Flexible Machine Learning Pipelines in R.” *Journal of Machine Learning Research*, **22**(184), 1–7. URL <https://jmlr.org/papers/v22/21-0281.html>.
- Bischl B, Lang M, Kotthoff L, Schiffner J, Richter J, Studerus E, Casalicchio G, Jones ZM (2016). “**mlr**: Machine Learning in R.” *Journal of Machine Learning Research*, **17**(170), 1–5. URL <https://jmlr.org/papers/v17/15-066.html>.
- Bischl B, Lang M, Richter J, Bossek J, Horn D, Kerschke P (2020). *ParamHelpers: Helpers for Parameters in Black-Box Optimization, Tuning and Machine Learning*. R package version 1.14, URL <https://CRAN.R-project.org/package=ParamHelpers>.
- Breiman L (2001). “Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author).” *Statistical Science*, **16**(3), 199–231. doi:10.1214/ss/1009213726.
- Brughmans D (2021). “Nearest Instance Counterfactual Explanations.” Github repository. URL <https://github.com/DBrughmans/NICE>. Commit 6389a39692e9b98ecb1734f6603167029987f870.
- Brughmans D, Martens D (2022). “NICE: An Algorithm for Nearest Instance Counterfactual Explanations.” *arXiv 2104.07411 v2*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2104.07411.
- Carreira-Perpiñán MA, Hada SS (2021). “Counterfactual Explanations for Oblique Decision Trees: Exact, Efficient Algorithms.” *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**(8), 6903–6911.

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 29

- Chang W (2021). **R6**: *Encapsulated Classes with Reference Semantics*. R package version 2.5.1, URL <https://CRAN.R-project.org/package=R6>.
- Dandl S, Hofheinz A, Binder M, Casalicchio G (2023). **counterfactuals**: *An R Package for Counterfactual Explanation Methods*. R package version 0.1.2, URL <https://CRAN.R-project.org/package=counterfactuals>.
- Dandl S, Molnar C, Binder M (2020a). “counterfactuals: Counterfactual Explanations.” Github repository. URL <https://github.com/susanne-207/moc>. Commit: d2fa9e0918d157c5d46a822b4ef110e641b45b76.
- Dandl S, Molnar C, Binder M, Bischl B (2020b). “Multi-Objective Counterfactual Explanations.” In T Bäck, M Preuss, A Deutz, H Wang, C Doerr, M Emmerich, H Trautmann (eds.), *Parallel Problem Solving from Nature – PPSN XVI*, pp. 448–469. Springer International Publishing, Cham. doi:10.1007/978-3-030-58112-1_31.
- de Oliveira RMB, Martens D (2021). “A Framework and Benchmarking Study for Counterfactual Generating Methods on Tabular Data.” *Applied Sciences*, **11**(16). doi:10.3390/app11167274.
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002). “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II.” *IEEE Transactions on Evolutionary Computation*, **6**(2), 182–197. doi:10.1109/4235.996017.
- Dowle M, Srinivasan A (2021). **data.table**: *Extension of ‘data.frame’*. R package version 1.14.2, URL <https://CRAN.R-project.org/package=data.table>.
- Dua D, Graff C (2017). “UCI Machine Learning Repository.” URL <http://archive.ics.uci.edu/ml>.
- Erickson N, Mueller J, Shirkov A, Zhang H, Larroy P, Li M, Smola A (2020). “AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data.” *arXiv 2003.06505*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2003.06505.
- Fisher RA (1936). “The Use of Multiple Measurements in Taxonomic Problems.” *Annals of Eugenics*, (2), 179—188. doi:10.1111/j.1469-1809.1936.tb02137.x.
- Freiesleben T (2021). “The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples.” *Minds and Machines*, **32**(1), 77–109. doi:10.1007/s11023-021-09580-9.
- Goldstein A, Kapelner A, Bleich J, Pitkin E (2015). “Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation.” *Journal of Computational and Graphical Statistics*, **24**(1), 44–65. doi:10.1080/10618600.2014.907095.
- Gower JC (1971). “A General Coefficient of Similarity and Some of Its Properties.” *Biometrics*, **27**(4), 857–871. doi:10.2307/2528823.
- Grömping U (2019). “South German Credit Data: Correcting a Widely Used Data Set.” *Reports in Mathematics, Physics and Chemistry, Department II, Beuth University of Applied*

- Sciences Berlin, **04/2019**. URL http://www1.beuth-hochschule.de/FB_II/reports/Report-2019-004.pdf.
- Guidotti R (2018). “LORE – LOcal Rule-Based Explanations.” Github repository. URL <https://github.com/riccotti/LORE>.
Commit: 710ffb42bf764bae90e9295e14349f0250fc2628.
- Guidotti R, Monreale A, Ruggieri S, Pedreschi D, Turini F, Giannotti F (2018). “Local Rule-Based Explanations of Black Box Decision Systems.” *arXiv 1805.10820*, arXiv.org E-Print Archive. doi:10.48550/arXiv.1805.10820.
- Harrison Jr D, Rubinfeld DL (1978). “Hedonic Housing Prices and the Demand for Clean Air.” *Journal of Environmental Economics and Management*, **5**(1), 81–102. doi:10.1016/0095-0696(78)90006-2.
- Hothorn T, Zeileis A (2021). “Predictive Distribution Modelling Using Transformation Forests.” *Journal of Computational and Graphical Statistics*, **14**, 144–148. doi:10.1080/10618600.2021.1872581.
- Karimi AH, Barthe G, Balle B, Valera I (2020). “Model-Agnostic Counterfactual Explanations for Consequential Decisions.” In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 895–905. PMLR. URL <https://proceedings.mlr.press/v108/karimi20a.html>.
- Karimi AH, Mohammadi K (2021). “mace.” Github repository. URL <https://github.com/amirhk/mace>. Commit: 01e6a405ff74e24dc3438a005cd60892154d189d.
- Karimi AH, Schölkopf B, Valera I (2021). “Algorithmic Recourse: From Counterfactual Explanations to Interventions.” In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 353–362. doi:10.1145/3442188.3445899.
- Kato S (2018). “R Package for Actionable Feature Tweaking.” Github repository. URL <https://github.com/katokohaku/featureTweakR>.
Commit: 6f3e614531fe2c9e475703afdc4deb8aaf62f78f.
- Kingma DP, Ba J (2017). “Adam: A Method for Stochastic Optimization.” *arXiv 1412.6980 v9*, arXiv.org E-Print Archive. doi:10.48550/arXiv.1412.6980.
- Kuhn M (2021). *caret: Classification and Regression Training*. R package version 6.0-88, URL <https://CRAN.R-project.org/package=caret>.
- Kuhn M, Wickham H (2020). *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles*. URL <https://www.tidymodels.org>.
- Lang M, Binder M, Richter J, Schratz P, Pfisterer F, Coors S, Au Q, Casalicchio G, Kotthoff L, Bischl B (2019). “**mlr3**: A Modern Object-oriented Machine Learning Framework in R.” *Journal of Open Source Software*, **4**(44). doi:10.21105/joss.01903.
- Lang M, Bischl B, Richter J, Sun X, Binder M (2022). *paradox: Define and Work with Parameter Spaces for Complex Algorithms*. R package version 0.9.0, URL <https://CRAN.R-project.org/package=paradox>.

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 31

- Lang M, Bischl B, Surmann D (2017). “**batchtools**: Tools for R to Work on Batch Systems.” *Journal of Open Source Software*, **2**(10). doi:10.21105/joss.00135.
- Li R, Emmerich MT, Eggermont J, Bäck T, Schütz M, Dijkstra J, Reiber J (2013). “Mixed Integer Evolution Strategies for Parameter Optimization.” *Evolutionary Computation*, **21**(1), 29–64. doi:10.1162/EVCO_a_00059.
- Liaw A, Wiener M (2002). “Classification and Regression by **randomForest**.” *R News*, **2**(3), 18–22.
- López-Ibáñez M, Dubois-Lacoste J, Cáceres LP, Birattari M, Stützle T (2016). “The **irace** Package: Iterated Racing for Automatic Algorithm Configuration.” *Operations Research Perspectives*, **3**, 43–58. doi:10.1016/j.orp.2016.09.002.
- Mahajan D, Tan C, Sharma A (2020). “Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers.” *arXiv 1912.03277 v3*, arXiv.org E-Print Archive. doi:10.48550/arXiv.1912.03277.
- Makiyama K (2015). “**pforeach**: An Easy Way to Parallel Processing in R.” Github repository. <https://github.com/hoxo-m/pforeach>.
Commit: c44f3bf651a4b2d5d5657bf8be3a94f93769871.
- Molnar C (2022). *Interpretable Machine Learning*. 2nd edition. URL <https://christophm.github.io/interpretable-ml-book>.
- Molnar C, Bischl B, Casalicchio G (2018). “**iml**: An R Package for Interpretable Machine Learning.” *JOSS*, **3**(26), 786. doi:10.21105/joss.00786.
- Moreira C, Chou YL, Hsieh C, Ouyang C, Jorge J, Pereira JM (2022). “Benchmarking Counterfactual Algorithms for XAI: From White Box to Black Box.” *arXiv 2203.02399 v2*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2203.02399.
- Pawelczyk M, Bielawski S, den Heuvel JV, Richter T, Kasneci G (2021). “**CARLA**: A Python Library to Benchmark Algorithmic Recourse and Counterfactual Explanation Algorithms.” *arXiv 2108.00783*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2108.00783.
- Pearl J (2009). *Causality: Models, Reasoning and Inference*. 2nd edition. Cambridge University Press, USA. doi:10.1017/CB09780511803161.
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011). “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research*, **12**, 2825–2830. URL <http://jmlr.org/papers/v12/pedregosa11a.html>.
- Pfisterer F, Poon J, Lang M (2021). “**mlr3keras**: **mlr3** Keras Extension.” Github repository. <https://github.com/mlr-org/mlr3keras>.
Commit: bad8434b7898b51b2143fc680594057c00dc7080.
- R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

- Schleich M, Geng Z, Zhang Y, Suci D (2021). “GeCo: Quality Counterfactual Explanations in Real Time.” *Proceedings of the VLDB Endowment*, **14**(9), 1681–1693. doi:10.14778/3461535.3461555.
- Stasinopoulos M, Rigby B, De Bastiani F (2021). *gamlss.data: Data for Generalised Additive Models for Location Scale and Shape*. R package version 6.0.2, URL <https://CRAN.R-project.org/package=gamlss.data>.
- Stepin I, Alonso JM, Catala A, Pereira-Fariña M (2021). “A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence.” *IEEE Access*, **9**, 11974–12001. doi:10.1109/ACCESS.2021.3051315.
- Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2014). “Intriguing Properties of Neural Networks.” *arXiv 1312.6199 v4*, arXiv.org E-Print Archive. doi:10.48550/arXiv.1312.6199.
- Therneau T, Atkinson B (2019). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-15, URL <https://CRAN.R-project.org/package=rpart>.
- Todeschini A (2021). *rchallenge: A Simple Data Science Challenge System*. R package version 1.3.4, URL <https://CRAN.R-project.org/package=rchallenge>.
- Tolomei G, Silvestri F, Haines A, Lalmas M (2017). “Interpretable Predictions of Tree-Based Ensembles via Actionable Feature Tweaking.” In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 465–474. doi:10.1145/3097983.3098039.
- Van der Loo M (2022). *gower: Gower’s Distance*. R package version 1.0.0, URL <https://CRAN.R-project.org/package=gower>.
- Van Rossum G, Drake Jr FL (1995). “Python Tutorial.” *Technical Report CS-R9526*, Centrum voor Wiskunde en Informatica.
- Vanschoren J, van Rijn JN, Bischl B, Torgo L (2014). “OpenML: Networked Science in Machine Learning.” *SIGKDD Explorations Newsletter*, **15**(2), 49–60. doi:10.1145/2641190.2641198.
- Verma S, Boonsanong V, Hoang M, Hines KE, Dickerson JP, Shah C (2022). “Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review.” *arXiv 2010.10596 v3*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2010.10596.
- Wachter S, Mittelstadt B, Russell C (2018). “Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR.” *Harvard Journal of Law & Technology*, **31**(2), 841–887. doi:10.2139/ssrn.3063289.
- Wexler J, Pushkarna M, Bolukbasi T, Wattenberg M, Viégas F, Wilson J (2019). “The What-If Tool: Interactive Probing of Machine Learning Models.” *IEEE transactions on visualization and computer graphics*, **26**(1), 56–65. doi:10.1109/TVCG.2019.2934619.
- Wilson DR, Martinez TR (1997). “Improved Heterogeneous Distance Functions.” *Journal of Artificial Intelligence Research*, **6**, 1–34. doi:10.1613/jair.346.

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 33

Xie L, Song Y, Lin T, Guo H, Wang B, Tang G, Liu C, Huang W, Yang Y, Ling W, et al (2019). “Association of plasma retinol levels with incident cancer risk in Chinese hypertensive adults: a nested case-control study.” *British Journal of Nutrition*, **122**(3), 293–300. doi: [10.1017/S000711451900120X](https://doi.org/10.1017/S000711451900120X).

Zitzler E, Thiele L (1998). “Multiobjective Optimization using Evolutionary Algorithms – A Comparative Case Study.” In AE Eiben, T Bäck, M Schoenauer, HP Schwefel (eds.), *Parallel Problem Solving from Nature – PPSN V*, pp. 292–301. Springer Berlin Heidelberg. doi: [10.1007/BFb0056872](https://doi.org/10.1007/BFb0056872).

A. Algorithmic reference

Algorithm 1 MOC based on Dandl *et al.* (2020b) as implemented in the **counterfactuals** R package (Section 2.1)

Inputs:

Data point to explain prediction for $\mathbf{x}^* \in \mathcal{X}$

Desired outcome (range) $Y' \subset \mathbb{R}$

Prediction function $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$

Observed data \mathbf{X}

Number of generations $n_{\text{generations}}$

Size of population μ

Recombination and mutation methods including probabilities

Selection method and initialization method

Stopping criterion

(Additional user inputs, e.g., range of numerical features, immutable features, distance function)

```

1: Initialize population  $P_0$  with  $|P_0| = \mu$ 
2: Evaluate candidates according to the four objectives of Equation 5
3: Set  $t = 0$ 
4: while stopping criterion not met
5:    $C_t = \text{create\_offspring}(P_t)$ ,  $|C_t| = \mu$  by selecting, recombining and mutating
     parents with given probabilities
6:   Combine parents and offspring  $R_t = C_t \cup P_t$ 
7:   Assign candidates to a front according to their objective values:
      $(F_1, F_2, \dots, F_m) = \text{nondominated\_sorting}(R_t)$ 
8:   for  $i = 1, \dots, m$ 
9:     Sort candidates within a front with (tailored) crowding distance sorting:
      $\tilde{F}_i = \text{crowding\_distance\_sort}(F_i)$ 
10:  end for
11:  Set  $P_{t+1} = \emptyset$  and  $i = 1$ 
12:  while  $|P_{t+1}| + |\tilde{F}_i| \leq \mu$ 
13:     $P_{t+1} = P_{t+1} \cup \tilde{F}_i$ 
14:     $i = i + 1$ 
15:  end while
16:  Choose first  $\mu - |P_{t+1}|$  elements of  $\tilde{F}_i$ :  $P_{t+1} = P_{t+1} \cup \tilde{F}_i[1 : (\mu - |P_{t+1}|)]$ 
17:   $t = t + 1$ 
18: end while
19: Return unique, non-dominated candidates of  $\bigcup_{k=0}^t P_k \setminus \mathbf{x}^*$ 

```

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 35

Algorithm 2 *NICE* based on Brughmans and Martens (2022) as implemented in the **counterfactuals** R package

Inputs:

Data point to explain prediction for $\mathbf{x}^* \in \mathcal{X}$
 Desired outcome (range) $Y' \subset \mathbb{R}$
 Prediction function $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$
 Observed data \mathbf{X}
 Reward function R_O , $O \in \{\text{sparsity, proximity, plausibility}\}$
 Indicator whether multiple counterfactuals should be returned *return_multi*
 Indicator whether to terminate as soon as desired prediction is reached *finish_early*
 (Additional user inputs, e.g., distance function)

```

1: Find closest observed datapoint  $x^{nn} \in \mathbf{X}$  to  $\mathbf{x}^*$  with desired prediction (Equation 8)
2: Set  $\mathbf{x}^{best} = \mathbf{x}^*$ 
3: Initialize archive set  $A = \emptyset$ 
4: Set  $J = \{j \in \{1, \dots, p\} : x_j^{nn} \neq x_j^{best}\}$ 
5: while ( $\hat{f}(\mathbf{x}^{best}) \notin Y'$  & finish_early == TRUE) | ( $J \neq \emptyset$ )
6:    $j^{best} = \emptyset$ 
7:   for  $j \in J$ :
8:      $\mathbf{x} = \mathbf{x}^{best}$ 
9:     Create new candidate by replacing one feature:  $x_j = x_j^{nn}$ 
10:    if  $R_O(\mathbf{x}) > R_O(\mathbf{x}^{best})$ :  $\mathbf{x}^{best} = \mathbf{x}$  and  $j^{best} = j$ 
11:    Save created candidate in an archive:  $A = A \cup \mathbf{x}$ 
12:  end for
13:  Update  $J = J \setminus j^{best}$ 
14: end while
15: if return_multi: return  $\{\mathbf{a} \in A : \hat{f}(\mathbf{a}) \in Y'\}$ 
16: else return  $\mathbf{x}^{best}$ 

```

B. The counterfactuals R package

B.1. Class diagram

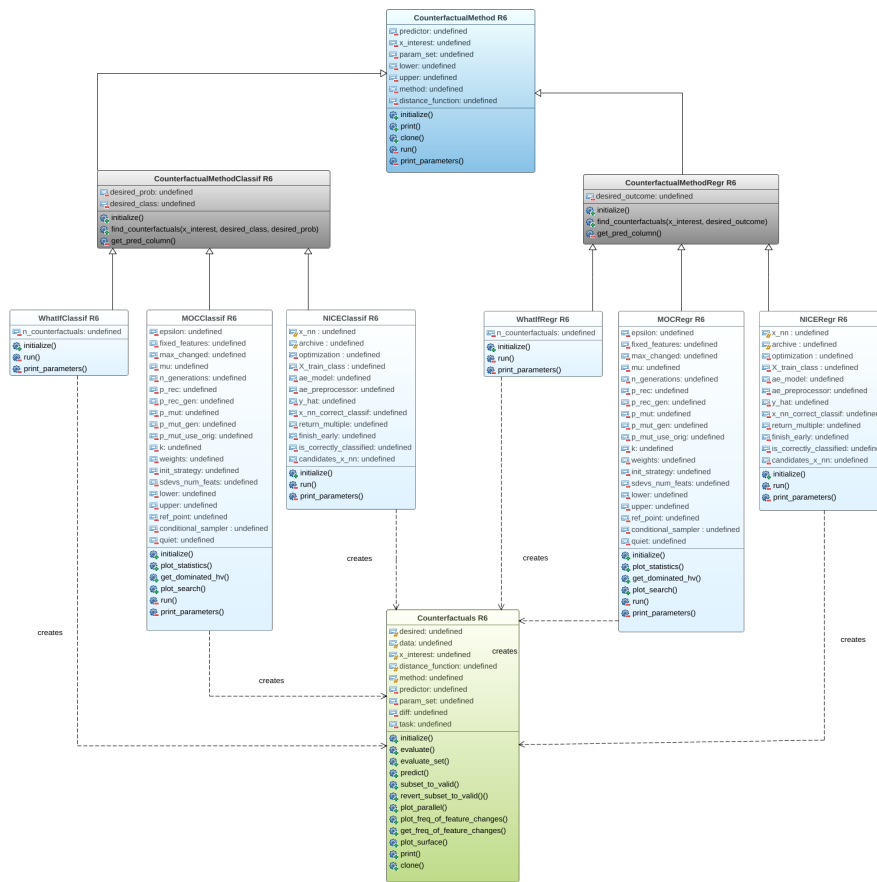


Figure 13: Detailed class diagram of the `counterfactuals` package.

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 37

B.2. Default values

The default parameter settings of the implementations of *WhatIf* and *NICE* should mimic the originally proposed methods in the corresponding papers (Wexler *et al.* 2019; Brughmans and Martens 2022). Our *MOC* implementation has the same parameters as the original *MOC* implementation proposed in (Dandl *et al.* 2020a) except for `p_rec_use_orig`. Instead of resetting after recombination *and* after mutation, we simplify things and reset only once after mutation with a probability of `p_mut_use_orig`. Due to the change in the dependency packages (`paradox` and `miesmuschel`, see Section 2.1), we re-tuned the *MOC* hyperparameters using the iterated F-race described in Dandl *et al.* (2020b) (see Appendix B). The code for tuning can be found here: https://github.com/dandls/moc/tree/irace_newversion. Although tuning identified the usage of the conditional mutator as a successor, we set `use_conditional_mutator` to `FALSE`, since it increases the runtime considerably.

Name	Description	Default
<code>n_counterfactuals</code>	The number of counterfactuals to be found.	1
<code>lower</code>	Vector of minimum values for numeric features named with the corresponding feature names. If <code>NULL</code> , the element for a numeric feature in <code>lower</code> is taken as its minimum value in observed data.	<code>NULL</code>
<code>upper</code>	Vector of maximum values for numeric features named with the corresponding feature names. If <code>NULL</code> , the element for a numeric feature in <code>upper</code> is taken as its maximum value in observed data.	<code>NULL</code>
<code>distance_function</code>	Distance function to compute the distances between the original and the training data points. Either the name of an already implemented distance function (<code>'gower'</code> or <code>'gower_c'</code>) or a function. If set to <code>'gower'</code> (default), then Gower's distance (Gower 1971) is used; <code>'gower_c'</code> is a C-based more efficient version of Gower's distance. A function must have three arguments <code>x</code> , <code>y</code> , and <code>data</code> , and must return a numeric matrix.	<code>'gower'</code>

Table 2: Parameters of *WhatIf* and their default values in the `counterfactuals` package.

Name	Description	Default
epsilon	If not NULL, candidates whose prediction is farther away from the desired interval than epsilon are penalized.	NULL
fixed_features	Names of features that are not allowed to be changed. NULL (default) allows all features to be changed.	NULL
max_changed	Maximum number of feature changes. NULL (default) allows any number of changes.	NULL
mu	The population size.	20
n_generations	The number of generations.	175
p_rec	Probability with which an individual is selected for recombination.	0.71
p_rec_gen	Probability with which a feature/gene is selected for recombination.	0.62
p_mut	Probability with which an individual is selected for mutation.	0.73
p_mut_gen	Probability with which a feature/gene is selected for mutation.	0.5
p_mut_use_orig	Probability with which a feature/gene is reset to its original value in <code>x_interest</code> after mutation.	0.4
k	The number of data points to use for the fourth objective (Equation (4)).	1
weights	The weights used to compute the weighted sum of dissimilarities for the fourth objective. It is either a single value or a vector of length <code>k</code> summing up to '1' (one weight for each of the <code>k</code> the closest points). NULL (default) means all data points are weighted equally.	NULL
lower	Vector of minimum values for numeric features named with the corresponding feature names. If NULL, the element for a numeric feature in lower is taken as its minimum value in observed data.	NULL
upper	Vector of maximum values for numeric features named with the corresponding feature names. If NULL, the element for a numeric feature in upper is taken as its maximum value in observed data.	NULL
init_strategy	The population initialization strategy. Can be 'random', 'sd', 'traidata' or 'icecurve'.	'icecurve'

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 39

use_conditional_mutator	Should a conditional mutator be used? The conditional mutator generates plausible feature values based on the values of the other features.	FALSE
distance_function	Distance function for the second and fourth objective. Either the name of an already implemented distance function ('gower' or 'gower_c') or a function. If set to 'gower' (default), then Gower's distance (Gower 1971) is used; 'gower_c' is a C-based more efficient version of Gower's distance. A function must have three arguments <i>x</i> , <i>y</i> , and <i>data</i> , and must return a numeric matrix.	'gower'

Table 3: Parameters of *MOC* and their default values in the **counterfactuals** package.

Name	Description	Default
optimization	The reward function to optimize. Can be 'sparsity' (default), 'proximity', or 'plausibility'.	'sparsity'
x_nn_correct	Should only correctly predicted observations be considered for the most similar instance search?	TRUE
margin_correct	Only for regression models. The accepted margin for considering a prediction as "correct". Ignored if x_nn_correct = FALSE. If NULL, the accepted margin is set to half the median absolute distance between the true and predicted outcomes in the observed data.	NULL
return_multiple	Should multiple counterfactuals be returned? If TRUE, the algorithm returns all created instances whose prediction is in the desired interval.	FALSE
finish_early	Should the algorithm terminate after an iteration in which the prediction for the highest reward instance is in the desired interval. If FALSE, the algorithm continues until x_nn is recreated.	TRUE

distance_function	Distance function for computing the distances between the original and the training data points for finding <code>x_nn</code> . Either the name of an already implemented distance function ('gower' or 'gower_c') or a function. If set to 'gower' (default), then Gower's distance (Gower 1971) is used; 'gower_c' is a C-based more efficient version of Gower's distance. A function must have three arguments <code>x</code> , <code>y</code> , and <code>data</code> , and must return a numeric matrix.	'gower'
-------------------	--	---------

Table 4: Parameters of *NICE* and their default values in the **counterfactuals** package.

B.3. Different Machine Learning Interfaces

The **counterfactuals** R package only allows machine learning models as an input that are instances of an `iml::Predictor` object. The `Predictor` class encapsulates a fitted model together with its underlying (training) data. In Section 4, we saw that it works off-the-shelf with models fitted with the **randomForest** and **mlr3** R packages (Liaw and Wiener 2002; Lang *et al.* 2019). In this section, we generate counterfactuals for the plasma retinol example of Section 4.2 for models trained with the **caret**, **tidymodels** and **mlr** packages (Kuhn 2021; Kuhn and Wickham 2020; Bischl *et al.* 2016). While all these machine learning interfaces allow training of a variety of models (linear models, model ensembles, etc.), for illustration, we focus on regression trees. Trees are fitted internally with **rpart** (Therneau and Atkinson 2019), such that – for the sake of completeness – we also show how to generate counterfactuals for a **rpart** tree. For each tree, we generate a counterfactual for the 100th row of the plasma dataset using the *NICE* method. The counterfactual should propose changes such that for the observation a plasma concentration larger than 500 ng/ml is predicted.

```
R> library("counterfactuals")
R> library("iml")
R> data("plasma", package = "gamlss.data")
R> x_interest = plasma[100L,]
```

caret package

First, we fit a regression tree model with the help of **caret**. To avoid tuning of the tree, we manually set the only tuning parameter `cp` to 0.01 – the default of the **rpart** package. Then, we initialize an `iml::Predictor` object with the fitted model as an input.

```
R> library("caret")
R> treecaret = caret::train(retplasma ~ ., data = plasma[-100L,],
+   method = "rpart", tuneGrid = data.frame(cp = 0.01))
R> predcaret = Predictor$new(model = treecaret, data = plasma[-100L,],
+   y = "retplasma")
R> predcaret$predict(x_interest)
```

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 41

```
## .prediction
## 1          342.92
```

For the 100th row of the plasma dataset (our `x_interest` or x^*), we predict a median value of 342.92 – the same as in Section 4.2. Next, we generate counterfactuals by initializing a `NICERegr` object with the instantiated `Predictor`.

```
R> nicecaret = NICERegr$new(predcaret, optimization = "proximity",
+   margin_correct = 0.5, return_multiple = FALSE)
R> nicecaret$find_counterfactuals(x_interest,
+   desired_outcome = c(500, Inf))

#> 1 Counterfactual(s)
#>
#> Desired outcome range: [500, Inf]
#>
#> Head:
#>   age sex smokstat  bmi vituse calories  fat fiber alcohol cholesterol
#> 1:  46  1      3 35.26    3  2667.5 131.6 10.1      0      550.5
#>   betadiet retdiet betaplasma
#> 1:    1210    1291      218
```

Since for all the examples shown in this section, we internally fit a `rpart` model to the same data, the prediction and the counterfactual for `x_interest` will be the same. We, therefore, omit the outputs for the prediction and counterfactual for the following machine learning interfaces.

tidymodels package

Regression trees of the `tidymodels` package also work off-the-shelf. However, for classification models, the `iml::Predictor` requires a prediction wrapper function (`predict.function`) such that class probabilities are returned instead of class labels. For details, the corresponding help page should be consulted.

```
R> library("tidymodels")
R> treetm = decision_tree(mode = "regression", engine = "rpart") %>%
  fit(retplasma ~ ., data = plasma[-100L,])
R> predtm = Predictor$new(model = treetm, data = plasma[-100L,],
+   y = "retplasma")
R> predtm$predict(x_interest)
R> nicetm = NICERegr$new(predtm, optimization = "proximity",
+   margin_correct = 0.5, return_multiple = FALSE)
R> nicetm$find_counterfactuals(x_interest = x_interest,
+   desired_outcome = c(500, Inf))
```

mlr package

For the `mlr` package, the workflow to generate counterfactuals is similar to the one for the `caret` package. We only need `mlr::RegrTask` and `mlr::regr.rpart` objects.

```
R> library("mlr")
R> task = mlr::makeRegrTask(data = plasma[-100L,], target = "retplasma")
R> mod = mlr::makeLearner("regr.rpart")
R> treemlr = mlr::train(mod, task)
R> predmlr = Predictor$new(model = treemlr, data = plasma[-100L,],
+   y = "retplasma")
R> predmlr$predict(x_interest)
R> nicemlr = NICERegr$new(predmlr, optimization = "proximity",
+   margin_correct = 0.5, return_multiple = FALSE)
R> nicemlr$find_counterfactuals(x_interest = x_interest,
+   desired_outcome = c(500, Inf))
```

rpart package

For sake of completeness, we also show how to generate counterfactuals for a regression model directly fitted with the **rpart** package.

```
R> library("rpart")
R> treerpart = rpart(retplasma ~ ., data = plasma[-100L,])
R> predrpart = Predictor$new(model = treerpart, data = plasma[-100L,],
+   y = "retplasma")
R> predrpart$predict(x_interest)
R> nicerpart = NICERegr$new(predrpart, optimization = "proximity",
+   margin_correct = 0.5, return_multiple = FALSE)
R> nicerpart$find_counterfactuals(x_interest = x_interest,
+   desired_outcome = c(500, Inf))
```

B.4. Class FeatureTweakerClassif

```
R> FeatureTweakerClassif = R6Class("FeatureTweakerClassif",
+   inherit = CounterfactualMethodClassif,
+   +
+   public = list(
+     initialize = function(predictor, ktree = NULL, epsilon = 0.1,
+       resample = FALSE) {
+       # adds predictor to private$predictor field
+       super$initialize(predictor)
+       private$ktree = ktree
+       private$epsilon = epsilon
+       private$resample = resample
+     }
+   ),
+   +
+   private = list(
+     ktree = NULL,
```

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 43

```

+   epsilon = NULL,
+   resample = NULL,
+
+   run = function() {
+     # Extract info from private fields
+     predictor = private$predictor
+     y_hat_interest = predictor$predict(private$x_interest)
+     class_x_interest = names(y_hat_interest)[which.max(y_hat_interest)]
+     rf = predictor$model
+
+     # Search counterfactuals by calling functions in featureTweakR
+     rules = getRules(rf, ktree = private$ktree,
+       resample = private$resample)
+     es = set.eSatisfactory(rules, epsilon = private$epsilon)
+     tweaks = featureTweakR::tweak(
+       es, rf, private$x_interest, label.from = class_x_interest,
+       label.to = private$desired_class, .dopar = FALSE
+     )
+     return(tweaks$suggest)
+   },
+
+   print_parameters = function() {
+     cat(" - epsilon: ", private$epsilon, "\n")
+     cat(" - ktree: ", private$ktree, "\n")
+     cat(" - resample: ", private$resample)
+   }
+ )
+ )

```

C. Benchmarking

C.1. Hyperparameter tuning

For hyperparameter tuning, we used random search (with 30 evaluations) and 5-fold CV with the misclassification error as a performance measure. Table 5 shows the tuning search space of each model. Numerical features were standardized and categorical ones were one-hot encoded using the **mlr3pipelines** package (Binder, Pfisterer, Lang, Schneider, Kotthoff, and Bischl 2021). The optimizer for the neural network was ADAM (Kingma and Ba 2017), and early stopping was imposed after 5 patience steps. All other hyperparameters were set to their default values in the packages of the mlr3 ecosystem (Lang *et al.* 2019). For the `hill_valley` dataset we used the default deep and wide architecture (two layers) inspired by Erickson, Mueller, Shirkov, Zhang, Larroy, Li, and Smola (2020) as implemented in the **mlr3keras** package without tuning (Pfisterer, Poon, and Lang 2021). Table 6 shows the accuracies of each model using nested resampling (with 5-fold CV in the inner and outer loop).

Model	Hyperparameter	Range
randomForest	ntrees	[0, 1000]
xgboost	nrounds	[0, 1000]
svm	cost	[0.01, 1]
logreg	-	-
neuralnet	lr	[0.00001, 0.1]
	layer_size	[1, 20]

Table 5: Tuning search space of each model. Hyperparameters `ntrees` and `nrounds` were log-transformed.

dataset	logistic_regression	neural_network	ranger	svm	xgboost
credit_g	0.72	0.71	0.71	0.73	0.70
diabetes	0.75	0.72	0.75	0.73	0.72
tic_tac_toe	0.97	0.98	0.95	0.79	0.98
bank8FM	0.94	0.94	0.94	0.95	0.94
hill_valley	0.60	0.53	0.56	0.48	0.57
run_or_walk_info	0.72	0.91	0.99	0.96	0.99

Table 6: Classification accuracies of each model on each dataset. The accuracies were computed using nested resampling with 5-fold CV in the inner and outer loop.

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 45

C.2. Additional results

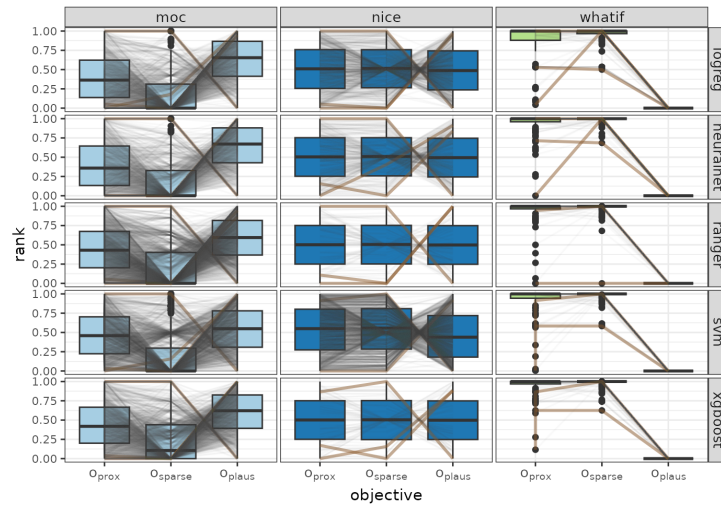
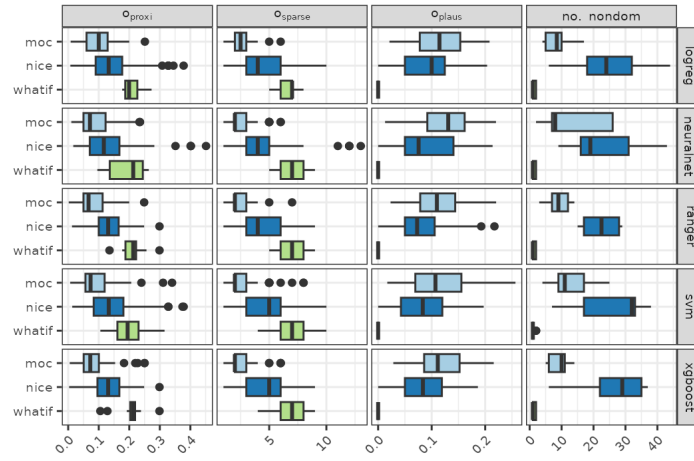
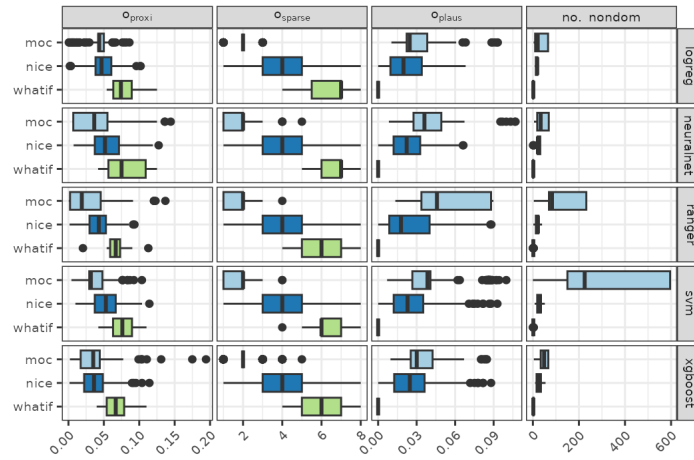


Figure 14: Comparison of *NICE*, *WhatIf*, and *MOC* w.r.t. their rank in the properties *Proximity* (ii, o_{prox}), *Sparsity* (iii, o_{spars}) and *Plausibility* (iv, o_{plaus}). Each gray line reflects a counterfactual (for clarity purposes, only a maximum of 2000 counterfactuals are displayed). The counterfactuals with the lowest and therefore best rank in an objective display the brown lines. Lower values are better.



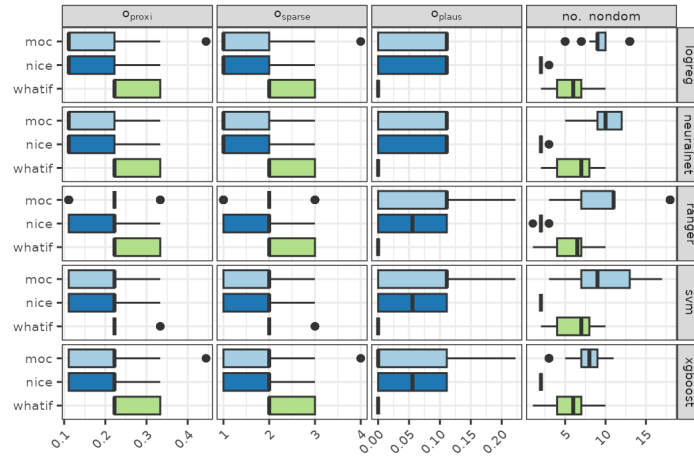
(a) credit_g



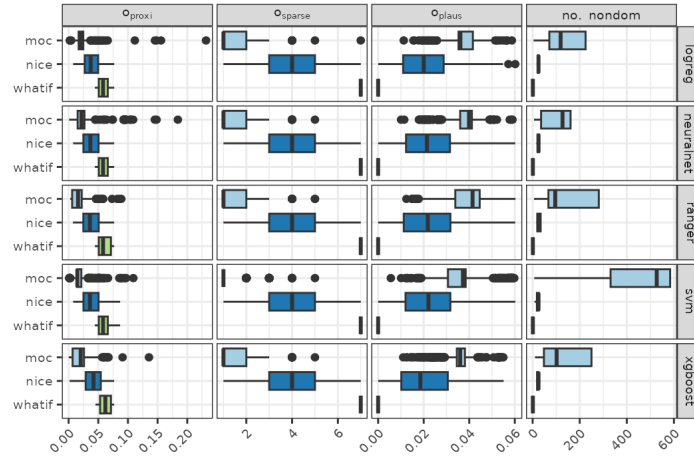
(b) diabetes

Figure 15: Comparison of *NICE*, *WhatIf*, and *MOC* w.r.t. the measures `dist_x_interest`, `no_changed`, `dist_train` (explained in Section 4), and `no. nondom` (number of non-dominated counterfactuals) for several models for the datasets `credit_g` and `diabetes`. O_{valid} was 0 for all counterfactuals. Lower values are better, except for `no. nondom`. The figure is based on [Dandl et al. \(2020b\)](#).

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 47

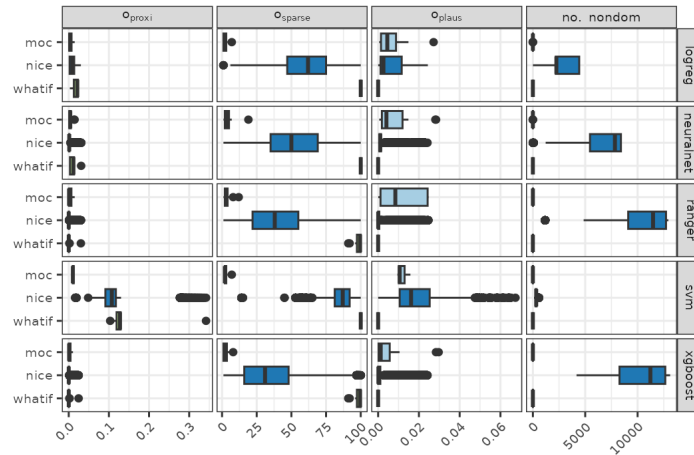


(a) `tic_tac_toe`

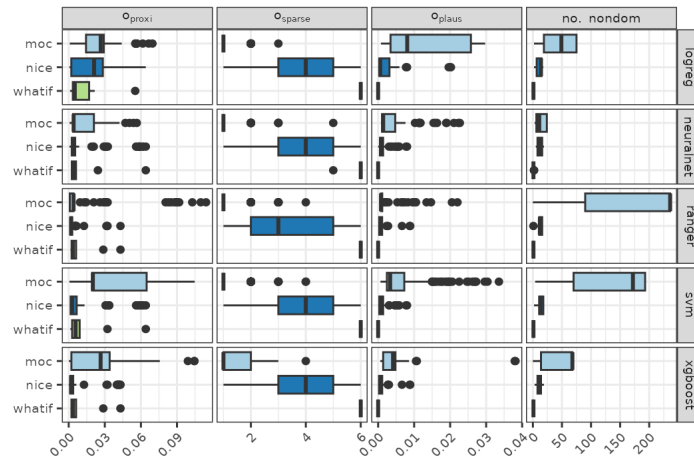


(b) `bank8FM`

Figure 16: Comparison of *NICE*, *WhatIf*, and *MOC* w.r.t. the measures `dist_x_interest`, `no_changed`, `dist_train` (explained in Section 4), and `no. nondom` (number of non-dominated counterfactuals) for several models for the datasets `tic_tac_toe` and `bank8FM`. O_{valid} was 0 for all counterfactuals. Lower values are better, except for `no. nondom`. The figure is based on Dandl *et al.* (2020b).



(a) hill_valley



(b) run_or_walk_info

Figure 17: Comparison of *NICE*, *WhatIf*, and *MOC* w.r.t. the measures `dist_x_interest`, `no_changed`, `dist_train` (explained in Section 4), and `no. nondom` (number of non-dominated counterfactuals) for several models for the datasets `hill_valley` and `run_or_walk_info`. o_{valid} was 0 for all counterfactuals. Lower values are better, except for `no. nondom`. The figure is based on Dandl *et al.* (2020b).

Susanne Dandl, Andreas Hofheinz, Martin Binder, Bernd Bischl, Giuseppe Casalicchio 49

Affiliation:

Susanne Dandl
Insitut für Statistik
Ludwig-Maximilians-Universität München, Germany
Ludwigstr. 33, 80539 Munich, Germany
Munich Center for Machine Learning (MCML), Germany
E-mail: Susanne.Dandl@stat.uni-muenchen.de

11 Interpretable Regional Descriptors: Hyperbox-Based Local Explanations

Contributing Article

Dandl S, Casalicchio G, Bischl B, Bothmann L (2023a). “Interpretable Regional Descriptors: Hyperbox-Based Local Explanations.” In D Koutra, C Plant, M Gomez Rodriguez, E Baralis, F Bonchi (eds.), *Machine Learning and Knowledge Discovery in Databases: Research Track (ECML PKDD 2023)*, pp. 479–495. Springer Nature Switzerland, Cham. doi: 10.1007/978-3-031-43418-1_29

Replication Code

The code for replicating this manuscript can be found at https://github.com/slds-lmu/supplementary_2023_ird.

Declaration of Contributions





Susanne Dandl had the idea for the general framework and proposed adaptations to previous methods to embed them into the framework. She programmed the related R package for the methods and implemented the use case. For the benchmark study, she proposed the list of quality measures and the general setup (data sets and machine learning algorithms). She implemented and performed the experiment, and aggregated and interpreted the results. She wrote the majority of the paper, including all the illustrations.

Contributions of Co-authors

Bernd Bischl had the initial idea to define regions with equal predictions. Ludwig Bothmann supervised and consistently provided guidance throughout the entire process. Giuseppe Casalicchio later joined the project and contributed to the benchmark study design. All co-authors provided valuable input and helped to revise the manuscript.



Interpretable Regional Descriptors: Hyperbox-Based Local Explanations

Susanne Dandl^{1,2} , Giuseppe Casalicchio^{1,2} , Bernd Bischl^{1,2} ,
and Ludwig Bothmann^{1,2} 

¹ Department of Statistics, LMU Munich, Ludwigstr. 33, 80539 Munich, Germany

² Munich Center for Machine Learning (MCML), Munich, Germany

Ludwig.Bothmann@stat.uni-muenchen.de

Abstract. This work introduces interpretable regional descriptors, or IRDs, for local, model-agnostic interpretations. IRDs are hyperboxes that describe how an observation’s feature values can be changed without affecting its prediction. They justify a prediction by providing a set of “even if” arguments (semi-factual explanations), and they indicate which features affect a prediction and whether pointwise biases or implausibilities exist. A concrete use case shows that this is valuable for both machine learning modelers and persons subject to a decision. We formalize the search for IRDs as an optimization problem and introduce a unifying framework for computing IRDs that covers desiderata, initialization techniques, and a post-processing method. We show how existing hyperbox methods can be adapted to fit into this unified framework. A benchmark study compares the methods based on several quality measures and identifies two strategies to improve IRDs.

Keywords: Interpretability · Semi-factual explanations · Hyperboxes

1 Introduction

Supervised machine learning (ML) models are widely used due to their good predictive performance, but they are often difficult to interpret due to their complexity. Post-hoc interpretation methods from the field of interpretable machine learning (IML) can help to draw conclusions about the inner processes of these models: local methods explain individual predictions and global methods explain the expected behavior of the model in general. Doshi-Velez and Kim [3] define model interpretability as “the ability to explain or to present in understandable terms to a human”. A topological form that satisfies this notion of interpretability is a hyperbox. In this work, we investigate hyperboxes as local interpretations that describe how the feature values of an observation can be changed without affecting its prediction. We call these boxes interpretable regional descriptors (IRDs). IRDs describe feature spaces by intervals for real-valued features and subsets of possible classes for categorical features (see Table 1).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14171, pp. 479–495, 2023.
https://doi.org/10.1007/978-3-031-43418-1_29

Table 1. Credit dataset [4,10] example with 9 features, showing the values of a customer with a moderate risk prediction. The IRD (generated by MaxBox & post-processing (Sect. 4)) shows how all features could be changed simultaneously so that the credit is still of moderate risk. \bar{B} shows how a single feature could be changed (keeping the other features fixed, see Sect. 4.1). For features in the upper half, the IRD covers the full observed value range (training data).

Feature	Customer	IRD	\bar{B} (1-dim IRD)	Range
sex	female	{female, male}	{female, male}	{female, male}
saving.accounts	little	{little, moderate, rich}	{little, moderate, rich}	{little, moderate, rich}
purpose	car	{car, radio/TV, furniture, others}	{car, radio/TV, furniture, others}	{car, radio/TV, furniture, others}
age	22	[19, 22]	[19, 75]	[19, 75]
job	skilled	{skilled, highly skilled}	{unskilled, skilled, highly skilled}	{unskilled, skilled, highly skilled}
housing	rent	{rent}	{own, free, rent}	{own, free, rent}
checking.account	moderate	{little, moderate}	{little, moderate}	{little, moderate, rich}
credit.amount	4000	[4000, 5389]	[2127, 8424]	[276, 18424]
duration	30	[26, 33]	[6, 44]	[6, 72]

1.1 Motivating Example for the Use of IRDs

A customer applies for a credit of €4000 at a bank to buy a new car. She is 22 years old, skilled, lives in a rented accommodation, has few savings and a moderate balance on her checking account. An ML model predicts whether the credit is of low, moderate or high risk. Due to a moderate risk prediction, the bank rejects the application. The IRD in Table 1 answers the question “to what extent the feature or multiple features can be changed such that the prediction is still in the moderate risk class”. From an IRD, multiple insights can be obtained.

First, IRDs offer a set of semi-factual explanations (SFEs) – also called a fortiori arguments – to justify a decision in the form of “even if” statements [23]. Compared to counterfactual explanations [31], SFEs reveal how feature values can be changed *without* affecting the prediction. For these statements to be convincing, domain knowledge is required, e.g., that higher balances in the savings account, and that higher skilled jobs decrease the risk for a bank. Given such knowledge, a multitude of SFEs can be derived from the IRD of Table 1 that (1) justify that a person is in the moderate risk class instead of the low risk class (e.g., “even if you had moderate savings and become highly skilled, your credit is still of moderate risk”)¹, and that (2) justify that a person is not in the high risk class (“even if you only have little balance in your checking account,

¹ In contrast, a counterfactual would be “if you had rich savings and become highly skilled, your credit would be a *low* risk”. Such statements are not covered by IRDs.

your credit would still be of moderate risk”). The latter represents a “safety bound” if some of the features change towards the undesired, higher risk class in the future.

Second, the interval width or cardinality of a feature in an IRD relative to its entire feature space can indicate whether a feature affects a prediction locally (under Theorems 1 and 2). For example, compared to credit amount or duration, savings or purpose seem to have no local effect on the prediction since the regional descriptor encompasses their entire observed feature ranges. These insights also reveal what can be options to change a given prediction.²

Third, IRDs are tools for model auditing. If the insights from a box (e.g., an SFE) agree with domain knowledge, users have more trust in the model, while disagreement helps to reveal unintended pointwise biases or implausibilities of a model. For example, an IRD that does not cover male customers *might* indicate that the model classifies individuals differently based on gender.³ An IRD that covers a credit amount of €300 and high balances in the checking account could indicate an inaccurate model because such customers should pose only a low risk to the bank. Other practical examples of IRDs shows Appendix A.⁴

1.2 Contributions

Our contributions are: 1) We introduce IRDs as a new class of local interpretations to describe regions in the feature space that do not affect the prediction of an observation; 2) We formalize the search for IRDs as an optimization problem and develop desired properties of IRD methods; 3) We introduce a unifying framework for computing IRDs including initialization and post-processing methods; 4) We show how existing hyperbox methods from data mining or IML can be adapted to fit into our unified framework; 5) We present a set of quality measures and compare our derived methods accordingly in a benchmark study; 6) We provide an open-access repository with an R package for the implemented approaches and the code for replicating the benchmark study.⁵

2 Methodology

Let $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$ be the prediction function of an ML model with $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_p$ as a p dimensional feature space. For classification models, we consider a pre-defined class of interest for which \hat{f} returns the predicted score or probability.

2.1 Formalizing the General Task for IRDs

Our goal is to find the largest hyperbox B covering a point of interest $\mathbf{x}' \in \mathcal{X}$ where all data points in B have a sufficiently close prediction to $\hat{f}(\mathbf{x}')$. The

² However, the concrete strategies can only reveal counterfactual explanations [31].

³ Note that if all genders are part of the box, it does not mean the model is fair.

⁴ https://github.com/slds-lmu/supplementary_2023_ird/blob/main/appendix.

⁵ https://github.com/slds-lmu/supplementary_2023_ird.

hyperbox B should have p dimensions $B = B_1 \times \dots \times B_p$

$$\text{with } B_j = \begin{cases} \{c|c \in \mathcal{X}_j\} & \text{categorical } X_j \\ [l_j, u_j] \subseteq \mathcal{X}_j & \text{numeric } X_j \end{cases},$$

consisting of intervals for numeric features and a subset of possible classes for categorical features. \mathcal{X}_j reflects the value space of the j th feature X_j . In accordance with Lemhadri et al. [22], a prediction is sufficiently close if it falls into a *closeness region*, which is a user-defined prediction interval $Y' = [\hat{f}(\mathbf{x}') - \epsilon_L, \hat{f}(\mathbf{x}') + \epsilon_H]$ with $\epsilon_L, \epsilon_H \in \mathbb{R}_{\geq 0}$.⁶ In the bank lending example, the closeness region should cover all model predictions that lead to the moderate risk class, e.g., a predicted probability of 30–60 % of defaulting, i.e., $Y' = [0.3, 0.6]$. To operationalize the above goal, we need three measures [25, 28]:

1. $\text{coverage}(B) = \mathbb{P}(\mathbf{x} \in B | \mathbf{x} \in \mathcal{X})$, which measures how much a hyperbox covers the entire feature space. Since, in practice, not all $\mathbf{x} \in \mathcal{X}$ are observable, we use an empirical approximation given data $(\mathbf{x}_i)_{1 \leq i \leq n}$ with $\mathbf{x}_i \in \mathcal{X}$

$$\widehat{\text{coverage}}(B) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(\mathbf{x}_i \in B). \tag{1}$$

2. $\text{precision}(B) = \mathbb{P}(\hat{f}(\mathbf{x}) \in Y' | \mathbf{x} \in B)$, the fraction of points within a box B whose predictions are inside Y' . Again, we use an empirical approximation

$$\widehat{\text{precision}}(B) = \frac{\sum_{i=1}^n \mathbb{I}(\mathbf{x}_i \in B \wedge \hat{f}(\mathbf{x}_i) \in Y')}{\sum_{i=1}^n \mathbb{I}(\mathbf{x}_i \in B)}. \tag{2}$$

3. an indicator of whether B covers \mathbf{x}'

$$\text{locality}(B) = \mathbb{I}(\mathbf{x}' \in B). \tag{3}$$

The following operationalizes the search for an IRD [25]:⁷

$$\begin{aligned} & \arg \max_{B \subseteq \mathcal{X}} (\widehat{\text{coverage}}(B)) \\ & \text{s.t. } \widehat{\text{precision}}(B) = 1 \text{ and } \text{locality}(B) = 1. \end{aligned} \tag{4}$$

Definition 1. A box is maximal if and only if no box could be added under full precision, such that for all numeric X_j , it holds that $(\nexists x_j \in \mathcal{X}_j \wedge x_j < l_j : \text{precision}(B \cup [x_j, l_j]) = 1) \wedge (\nexists x_j \in \mathcal{X}_j \wedge x_j > u_j : \text{precision}(B \cup [u_j, x_j]) = 1)$, and for all categorical X_j , it holds that $(\nexists x_j \in \mathcal{X}_j \setminus B_j : \text{precision}(B \cup x_j) = 1)$.

⁶ For classification models, $Y' \subset [0, 1]$ must hold.

⁷ For this, we extended the optimization task of Ribeiro et al. [25] to target IRDs by aiming for a precision of 1 and by including the locality constraint.

A box B with maximum coverage satisfies this maximality property. We aim for a maximal B , since B can then detect features that are not locally relevant for a prediction $\hat{f}(\mathbf{x}')$. We prove the following in Appendix B.

Theorem 1. *If B is maximal, $B_j = [\min(\mathcal{X}_j), \max(\mathcal{X}_j)]$ holds for numeric features X_j and $B_j = \mathcal{X}_j$ for categorical X_j that are not involved in model \hat{f} .*

Similarly, we aim for homogeneous boxes B such that $\text{precision}(B) = 1$. Then, B can detect features that are locally relevant for $\hat{f}(\mathbf{x}')$. We prove the following in Appendix C.

Theorem 2. *If $\text{precision}(B) = 1$, $B_j \subset \mathcal{X}_j$ holds for a feature that is locally relevant for $\hat{f}(\mathbf{x}')$.*

2.2 Desiderata for IRDs

In Sect. 3, we discuss related methods to generate B . The suitability of these methods as IRD methods relies on whether they consider all objectives of Eq. (4) and whether they satisfy the following desired properties for IRDs.

Interpretability. In order for B to be interpretable, we only consider methods that return a *single* p -dimensional hyperbox. The hyperrectangular structure of B allows for a natural interpretation, which is not the case for hyperellipsoids or polytopes formed by halfspaces [22]. According to Eq. (4), B needs to cover \mathbf{x}' , which is the case if the following holds: $\forall j \in \{1, \dots, p\} : x'_j \in B_j$.

Model-agnosticism. The definition of \hat{f} does not pose any restrictions on the ML model or the feature space. Therefore, methods should be model-agnostic such that they could explain both regression or classification models with various feature types (binary, nominal, ordinal or continuous).

Sparsity Constraints. Eckstein et al. [5] proved that the optimization task for the maximum box problem is \mathcal{NP} -hard if the features defining the box are not fixed. This also applies to the search for IRDs, which only additionally requires $\mathbf{x}' \in B$. Since the search space for hyperboxes grows with the number of features, it is infeasible to consider all potential solutions. Furthermore, the fact that IRDs have as many dimensions as the dataset impedes their interpretability – the very goal of IRDs in the first place. To reduce the number of features, methods should be able to adhere to user-defined sparsity constraints such that for some features X_j , $B_j = x'_j$. Section 7 discusses other solutions.

3 Related Work

The optimization task of Eq. (4) can be understood mathematically as finding the preimage of prediction values $\in Y'$ in the neighborhood of \mathbf{x}' . Therefore, IRDs can be seen as a subset of a level set for function values $\in Y'$. Level set

approximations often consist of points [7], and only a few approaches approximate these via hyperboxes [32, 33] (or other geometric forms). These methods produce multiple boxes instead of one and do not require to contain \mathbf{x}' . Hence, they are not interpretable in our sense and, therefore, not useful to produce IRDs.

In data mining, Eckstein et al. [5] proposed a maximum box (MaxBox) approach for datasets with binary outcomes to find the largest homogeneous hyperbox w.r.t. the positive class. Friedman and Fisher [11] derived the patient rule induction method (PRIM) for seeking boxes in the feature space in which the outcome mean is high. Both approaches do not require \mathbf{x}' to be in the box.

Table 2. Overview of approaches that search for hyperboxes in feature spaces.

	Objectives			Desiderata		
	Coverage	Precision	Locality	Interpretable	Agnostic	Sparse
Level set methods						
PBnB [32, 33]	✓	✓	×	×	✓	×
Data mining						
MaxBox [5]	✓	✓	×	✓	×	×
PRIM [11]	×	×	×	✓	×	×
Post-hoc IML						
Anchors [25]	✓	✓	✓	✓	×	×
MAIRE [28]	✓	✓	✓	✓	×	×
LORE [14–16]	×	×	✓	✓	✓	×
Interpretable classifier						
Column generation [1]	✓	✓	×	×	✓	×

As described earlier, IRDs may also be seen as a method to summarize a multitude of SFEs. Most proposed methods for SFEs return only a single point as an explanation [2, 17, 23]. In contrast, LORE by Guidotti et al. [14–16] returns a set of SFEs using surrogate trees. Their approach reveals which feature values are most important for deriving a prediction by following the path to the point of interest. The reliability of such a surrogate tree depends on the assumption that the tree can adequately replicate the underlying model, which may not always be the case [27]. Furthermore, LORE does not directly target Eq. (4) because the level of precision cannot be set [16] and homogeneous boxes are only possible with overfitting/deep-grown trees. This limits its coverage (the box could be larger than the terminal node (Figure S. 5 in the Appendix)) and makes this approach computationally expensive [6, 8]. Therefore, the tree structure is more suitable for deriving SFEs when the underlying model is tree-based [9, 29].

An IML method that utilizes hyperboxes is the Anchors approach [25]. The returned hyperbox indicates how features must be fixed or anchored to prevent a model from changing the classification of a data point. Anchors were originally

proposed to aim for hyperboxes that also partly cover observations of other classes; a precision of 0.95 is the default in its implementation [26]. Although the precision can be changed to 1, Anchors are nevertheless not suitable for the generation of IRDs due to their limited search space: Either the box boundary of a feature is set to the full feature range observed in the data, or to the value of \mathbf{x} . This bears the risk of “overly specific anchors” with low coverage [25]. For larger coverage, features can be binned beforehand. However, no established discretization technique for Anchors exists so far and the optimization procedure underlying Anchors does not allow adaptations of the bins during optimization.

To overcome the discretization problem, Sharma et al. [28] proposed the model-agnostic interpretable rule extraction (MAIRE) procedure. MAIRE finds more optimal boundaries for continuous features via gradient-based optimization. It still does not allow a more precise choice for categorical features; either the box allows no changes to a feature or it covers all possible values of a feature.

Equation (4) also overlaps with the problem of deriving interpretable (surrogate) models using a combination of rules [12] or hyperboxes [18] that cover the whole feature space (e.g., via column generation [1]). As such, the methods do not focus on locality and are not interpretable in our sense.

Table 2 summarizes whether the addressed methods are suitable for generating IRDs. Overall, none of the methods satisfies all objectives of Eq. (4) and desiderata from Sect. 2.2. Specifically, none of them addresses sparsity constraints, and only a few are model-agnostic. In Sect. 4.4, we modify MaxBox, PRIM, and MAIRE such that they fulfill all of our requirements to transform them into useful IRD methods. All other methods cannot be modified to the required extent due to their underlying, irreplaceable optimization methods that do not directly target Eq. (4) (LORE), target multiple boxes (PBnB) or have a very limited search space. The latter applies in particular to Anchors. However, the method serves as a baseline method for our benchmark study in Sect. 6.

4 Generating IRDs

We now present a unifying framework for generating IRDs, which consists of four steps: restriction, selection, initialization, and optimization. Optionally, a post-processing step can be conducted (Sect. 4.5).

4.1 Restriction of the Search Space

To restrict the initial search space for B , we propose a simple procedure to find the largest local box \bar{B} of \mathbf{x}' such that $B \subseteq \bar{B}$. For a continuous feature X_j , we vary its value x'_j of \mathbf{x}' on an equidistant grid. Upper and lower bounds of \bar{B}_j are set to the minimal changes in x'_j , yielding a prediction outside Y' . This approach is similar to individual conditional expectation (ICE) values [13]. For a categorical feature X_j , \bar{B}_j comprises all classes of \mathcal{X}_j that still lead to a prediction $\in Y'$ after adapting x'_j of \mathbf{x}' . If a user sets the sparsity constraint that feature X_j is immutable, $\bar{B}_j = x'_j$ must hold. We prove the following in Appendix D.

Theorem 3. *For any box B that solves the optimization problem of Eq. (4) it holds that $B \subseteq \underline{B}$.*

4.2 Selection of the Underlying Dataset

All methods need a dataset $\underline{\mathbf{X}}$ consisting of $\mathbf{x} \in \mathcal{X}$ as an input. This dataset is used for evaluating (competing) boxes w.r.t. the empirical versions of coverage and precision (Eq. (1) and Eq. (2)). For some methods, the dataset also offers a set of potential box boundaries to be evaluated. A suitable dataset is the training data. Since only instances $\in \underline{B}$ are relevant (Theorem 3), we remove all instances $\notin \underline{B}$ from $\underline{\mathbf{X}}$. Consequently, $x_j = x'_j \forall \mathbf{x} \in \underline{\mathbf{X}}$ holds for all immutable features X_j . More features and sparsity constraints increase the risk that $\underline{\mathbf{X}}$ is only sparsely populated around \mathbf{x}' . Furthermore, training data may not be readily available. Since we aim for IRDs that are faithful to the model and not to the data-generating process (DGP), data can be artificially generated by uniformly sampling from the admissible feature ranges of \underline{B} . In Sect. 6, we inspect how double-in-size sampled data⁸ within \underline{B} affects the quality of IRDs and IRD methods compared to using training data.

4.3 Initialization of a Box

All methods require an initial box B as an input, which is either set to the largest local box \underline{B} covering all $\underline{\mathbf{X}}$ or the smallest box possible, which only contains \mathbf{x}' . We define methods that start with the largest local box as top-down IRD methods, and methods that start with the smallest box possible as bottom-up methods.

4.4 Optimization of Box Boundaries

The last step comprises the optimization of the box boundaries. Top-down methods iteratively shrink the box boundaries of the largest local box to improve the box's precision (upholding that $\mathbf{x}' \in B$), while bottom-up methods iteratively enlarge the box boundaries of the smallest box to improve the box's coverage (upholding the precision at 1). In this section, we describe the MaxBox, MAIRE, and PRIM approaches and our extensions such that the methods optimize Eq. (4) and fulfill the desiderata of Sect. 2.2. Pseudocodes and illustrations of the inner workings of the extended approaches are given in Appendix E. All methods receive as input a dataset $\underline{\mathbf{X}}$ and an initial box B .

MaxBox – Top-down Method. MaxBox was originally proposed for binary classification problems – with a positive and negative class. The method starts with the largest box covering all data. A branch and bound (BnB) algorithm [21] inspects the options to shrink the box to optimize its precision w.r.t. the positive class. The branching rule creates new boxes by bracketing out a sample \mathbf{x} of

⁸ Double-in-size refers to the size of the training data, not of $\underline{\mathbf{X}}$.

the negative class, such that the box is shrunk to be either below or above the values of \mathbf{x} in at least one feature dimension (categorical features are one-hot encoded). Estimates of the upper bound for the coverage of a box determine which imprecise box is branched next, which sample is used for branching, and which boxes are discarded because their upper bound does not exceed the coverage of the current largest homogeneous box. If no boxes to shrink are left, the largest homogeneous box is returned as an IRD.

Extensions. By labeling observations with predictions $\in Y'$ as positive, the approach becomes model-agnostic. Since the original algorithm does not consider whether corresponding boxes still include \mathbf{x}' , we adapted the approach to discard boxes that do not contain \mathbf{x}' to guarantee locality.

PRIM – Top-down Method. The method originally aims for boxes with a high average outcome. The procedure starts with a box that includes all points. In the peeling phase, PRIM iteratively identifies a set of eligible subboxes (defined by the α - and $(1-\alpha)$ -quantile for numeric features and each present category for categorical features) and peels off the subbox that results in the highest average outcome after exclusion. This step is repeated until the number of points included in the box drops below a fraction of the total number of points. In the pasting phase, the box is iteratively enlarged by adding the subbox that increases the outcome mean the most. These subboxes consist of at least α observations with the nearest lower or higher values in one dimension (numeric X_j) or with a new category (categorical X_j).

Extensions. We adapted the approach to target Eq. (4): in each peeling iteration, the subbox is excluded such that the resulting box has the highest precision (coverage acts as a tiebreaker), and in each pasting iteration, the largest homogeneous subbox is added. If the precision and coverage are not sufficient to select a best box for peeling or pasting, a subbox is randomly selected from the best ones. Peeling stops as soon as the resulting box is homogeneous, while pasting stops as soon as there exists no homogeneous box to add. Furthermore, only subboxes that do not cover \mathbf{x}' are peeled. According to the authors' recommendation, we use $\alpha = 0.05$ for the benchmark study (Sect. 6).

MAIRE – Bottom-up Method. The method starts with a box covering \mathbf{x}' . In each iteration, the box boundaries are adapted via ADAM [19] by optimizing a differentiable approximation of the coverage measure. If the precision falls below a certain threshold or \mathbf{x}' is not part of the box, the method additionally optimizes a differentiable version of Eq. (2) and Eq. (3), respectively. MAIRE stops after a specified number of iterations. In the end, the method returns the largest homogeneous box over the iterations.

Extensions. The method requires 0–1-scaled features. To overcome the one-vs-all issue for categorical features (Sect. 3), we one-hot-encode categorical features. We implemented a convergence criterion for a fair comparison with the other (convergent) approaches: we let MAIRE enlarge the box boundaries until the precision falls below 1, then MAIRE is only allowed to run for another 100 iterations. The

implementation for the experiments in Sect. 6 is based on the authors' implementation [28] with the discussed modifications. The hyperparameters were set according to the authors' recommendations. We only set the precision threshold to 1, rather than 0.95.

4.5 Post-processing

All methods described in the previous section determine box boundaries based on a finite number of data points in $\underline{\mathbf{X}}$. The limited access carries the risk that some regions of the feature space are not represented in $\underline{\mathbf{X}}$ and that the boundaries of a generated B are suboptimal: There could be areas in B that have predictions $\notin Y'$, or there could be adjacent areas outside of B that also have predictions $\in Y'$. To improve the box boundaries of a given box B , we developed the following post-processing method using newly sampled data. The procedure consists of peeling and pasting as PRIM.

First, the precision of B is measured based on newly sampled data. If $\exists \mathbf{x} \in B$ with $\hat{f}(\mathbf{x}) \notin Y'$, subboxes with the lowest precision in proportion to their size (according to newly sampled data within this subbox) are iteratively peeled. If all subboxes to peel are homogeneous, peeling stops. In the subsequent pasting step, the largest subboxes that proved to be homogeneous (according to newly sampled data within this subbox) are added. If the best box cannot be determined (because several boxes have the same precision and coverage), a subbox is randomly chosen. The method has three hyperparameters: the number of samples used for evaluation, the relative box size (in relation to the size of \mathcal{X}_j) for peeling or pasting boxes for continuous features, and a threshold for the minimum box size. The latter acts as a stopping criterion for pasting. If no homogeneous subbox can be added, the relative box size to add for continuous features is halved as long as the relative box size is not lower than the threshold. The pseudocode of our method displays Appendix F.

Section 6 investigates whether our post-processing method improves IRDs. For the experiments, we set the number of samples to evaluate boxes to 100, the relative box size to 0.1, and the threshold for the minimum box size to 0.05.

5 Quality Measures

We now present a set of quality measures for *generated IRDs* and *IRD methods*. These measures apply to a single instance \mathbf{x}' to be explained, where B is the returned IRD of \mathbf{x}' of an IRD method G . The assessment requires evaluation data \mathbf{E} consisting of $\mathbf{x} \in \mathcal{X}$; for the benchmark study in Sect. 6, we use training data and new data uniformly sampled from \underline{B} . Training data helps to assess whether the methods use the training data appropriately during IRD generation (e.g., precision should be 1), while a proliferated number of newly generated data $\in \underline{B}$ leads to a more precise evaluation w.r.t. the model, not the DGP.

Locality. The IRD should cover \mathbf{x}' . This property is fulfilled if $locality(B) = \mathbb{I}(\mathbf{x}' \in B)$ equals 1.

Coverage. Given two IRDs with equal precision, we prefer the one with higher coverage (Eq. (1)). To evaluate the coverage, we use samples $\mathbf{x} \in \mathbf{E}$ from the connected convex level set \mathcal{L} covering \mathbf{x}' .

Definition 2. A data point \mathbf{x} with $\hat{f}(\mathbf{x}) \in Y'$ is part of \mathcal{L} of \mathbf{x}' iff there exists a path between \mathbf{x} and \mathbf{x}' for which all intermediate points have a prediction $\in Y'$.

Paths are identified via the identification algorithm of Kuratomi et al. [20], details are given in Appendix G.

Precision. Given two IRDs with equal coverage, the IRD with higher precision is preferred (Eq. (2)).

Maximality. A box should be maximal (Definition 1) based on $\mathbf{x} \in \mathbf{E}$.

No. of Calls. Lower number of calls to \hat{f} of an IRD method are preferred.⁹

Robustness. If we rerun method G on the same \mathbf{x}' and \hat{f} R times using the same $\underline{\mathbf{X}}$, the produced IRDs B_1, \dots, B_R should overlap with the originally produced B , such that $robustness(G) = \min_{k \in \{1, \dots, R\}} \frac{\sum_{\mathbf{x} \in \mathbf{E}} \mathbb{I}(\mathbf{x} \in B \cap B_k)}{\sum_{\mathbf{x} \in \mathbf{E}} \mathbb{I}(\mathbf{x} \in B \cup B_k)}$ has a high value.

6 Performance Evaluation

In a benchmark study, we address the following research questions (RQs):

1. How do MaxBox, MAIRE and PRIM perform against each other w.r.t. the quality measures of Sect. 5 (training data as $\underline{\mathbf{X}}$, no post-processing)?
2. What effect do double-in-size sampled data originating from \underline{B} have on the quality compared to using training data?
3. What effect does the post-processing (Sect. 4.5) have on the quality?

As a baseline method, we use the Anchors approach [25] with a precision of 1 and 20-quantile-based bins for numeric features (see Sect. 3 for details).

6.1 Setup

To answer the RQs, we utilize six datasets from the OpenML platform [30], either with a binary, multi-class or continuous target variable. Table 3 summarizes the datasets' dimensions, target and feature types. For each dataset, five data points were randomly sampled to be \mathbf{x}' .¹⁰ On each of the datasets, four models were trained: a hyperbox model, a logistic regression/multinomial/linear

⁹ We prefer this measure over computation time because it is independent of the concrete implementation. We have made our best efforts to implement the methods efficiently, but there is usually room for improvement.

¹⁰ These data points can also be excluded from the data before training a model. However, our experiments showed the results for the RQs are almost the same.

model (depending on the outcome), a neural network with one hidden layer, and a random forest model. The number of trees for the random forest and the neurons on the hidden layer were tuned (details are given in Appendix H). The hyperbox model is derived from a classification and regression tree (CART) model for each \mathbf{x}' individually. For a given \mathbf{x}' , the post-processed model predicts 1 if a point falls in the same terminal node as \mathbf{x}' and 0 otherwise.¹¹

Table 3. Overview of benchmark datasets.

Name	OpenML ID	Target type	Rows	Continuous	Categorical
diabetes	37	binary	768	8	0
tic_tac_toe	50	binary	958	0	9
cmc	23	three-class	1473	2	7
vehicle	54	four-class	846	18	0
no2	886	regression	500	7	0
plasma_retinol	511	regression	315	10	3

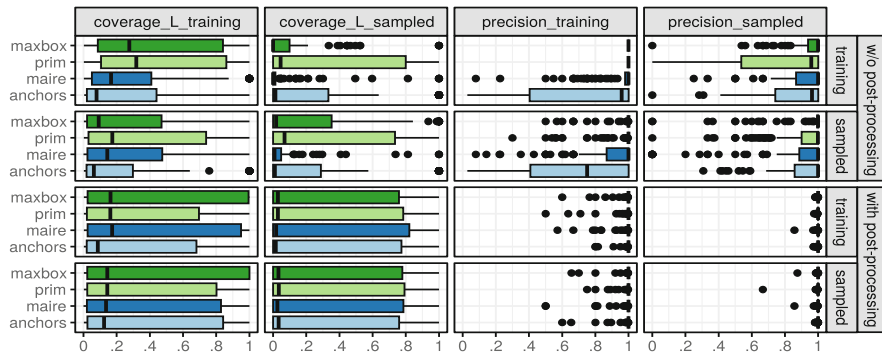


Fig. 1. Comparison of methods w.r.t. coverage and precision. Addendum L means that for the coverage evaluation only training or sampled points within \mathcal{L} are considered. Each point in the boxplot reflects one IRD. Methods were either run or evaluated on training data or uniformly sampled data from \underline{B} , and with or without post-processing. Higher values are better.

For classification models, the prediction function returns the probability of the class with the highest probability for \mathbf{x} . For binary targets, we set $Y' = [0.5, 1]$. For regression and multi-class targets, Y' is set to $[\hat{f}(\mathbf{x}) - \delta, \hat{f}(\mathbf{x}) + \delta]$

¹¹ The true hyperbox of the CART model might be larger than the terminal node-induced hyperbox (see Figure S. 5 in the Appendix).

with δ as the standard deviation of predictions \hat{f} of the training data. For multi-class, the interval is additionally capped between 0 and 1. For each dataset, model, and \mathbf{x}' , we generate IRDs with MaxBox, PRIM, and MAIRE, as well as Anchors – our baseline method. The hyperparameters of the methods were set according to Sect. 4. The methods were either run on training or on uniformly sampled data from \bar{B} (RQ 2), and either without or with post-processing (RQ 3). For the robustness evaluation, we repeated the experiments $R = 5$ times.

The methods and their generated IRDs were evaluated based on the performance measures of Sect. 5 – either evaluated on the training data or 1000 new instances sampled uniformly from \bar{B} . We also compared the methods statistically by conducting Wilcoxon rank-sum tests for the hypothesis that the distribution of the coverage and precision values do not differ between two (IRD) methods (RQ 1), for a method using training vs. sampled data (RQ 2), and for a method without vs. with post-processing (RQ 3). The experiments were conducted on a computer with a 2.60 GHz Intel(R) Xeon(R) processor, and 32 CPUs. Overall, generating the boxes took 63 h spread over 20 CPUs. The five repetitions for the robustness evaluation required another 316 h.

Table 4. Comparison of methods w.r.t. maximality and no. of calls to \hat{f} averaged over all datasets, models and \mathbf{x}' . Each method was run or evaluated on training data or uniformly sampled data from \bar{B} , and without (0) or with (1) post-processing. Higher maximality and lower no. of calls are better.

	Training data						Sampled					
	Max _{training}		Max _{sampled}		No. calls to \hat{f}		Max _{training}		Max _{sampled}		No. calls to \hat{f}	
	0	1	0	1	0	1	0	1	0	1	0	1
MaxBox	0.60	0.42	0.06	0.41	184	55769	0.23	0.45	0.24	0.43	1621	37627
PRIM	0.42	0.37	0.18	0.39	184	46070	0.20	0.42	0.25	0.39	1621	42958
MAIRE	0.18	0.41	0.04	0.41	184	68126	0.06	0.41	0.11	0.35	1621	92976
Anchors	0.27	0.42	0.16	0.40	26402	94448	0.31	0.42	0.18	0.36	77818	129276

6.2 Results

Figure 1 compares the coverage and precision values of the methods visually. Table 4 shows the frequency of fulfilling maximality and the number of calls to \hat{f} of the methods. The separate results for each dataset and model, the statistical analysis, and the results of robustness are shown in Appendix I. We omitted the results for the locality measure because all returned IRDs covered \mathbf{x}' .

RQ 1 - Comparison of Methods. Without post-processing and training data as \bar{X} (first row, Fig. 1), MaxBox had the highest precision as evaluated on training and newly sampled data. The IRDs of PRIM had on average the largest coverage, but they also covered sampled data with predictions outside Y' . Due to the randomized choice of a subbox in the case of ties, PRIM is not robust according

to our robustness metric. None of the methods outperformed the other methods w.r.t. maximality. By design, MAIRE’s optimizer disregards the constraints on the search space (\underline{B}), resulting in precisions below 1 on training data. Overall, all methods outperformed the baseline method Anchors according to coverage and precision. While all other methods called \hat{f} $|\underline{X}|$ times, Anchors evaluates column-wise permutations of the observed data.

RQ 2 - Training vs. Sampled Data. On average, double-in-size sampled data originating from \underline{B} led to slightly higher coverage, precision and maximality rates w.r.t. newly sampled data but not w.r.t. the training data. Due to the increase in the size of \underline{X} , more calls to \hat{f} were necessary.¹²

RQ 3 - Without vs. With Post-processing Post-processing increased the coverage and precision of IRDs for all methods. The difference in the quality of IRDs between the methods and between the underlying data scheme (training data vs. sampled data) diminished. Quality enhancement comes at the cost of efficiency and robustness; on average, post-processing resulted in 57,000 additional calls to \hat{f} and the sampling of new data decreased the robustness. MAIRE required on average the most post-processing iterations, followed by Anchors.

7 Conclusion, Limitations and Outlook

Conclusion. We introduced IRDs that describe regions in the feature space that do not affect the prediction of an instance in the form of hyperboxes. These hyperboxes provide a set of semi-factual explanations to justify a prediction, and indicate which features affect a prediction and whether there might be pointwise biases or implausibilities. We formalized the search for IRDs, and introduced desiderata, a unifying framework and quality measures for IRD methods. We discussed three existing hyperbox methods in detail and adapted them to search for IRDs. The lack of a method “ruling it all” in the benchmark study emphasizes the need for a unifying framework comprising multiple methods. The study also revealed that a larger, uniformly sampled dataset and our post-processing method can further enhance the quality of IRDs (at the cost of efficiency).

Limitations. Our work offers potential for further research, e.g., on the sensitivity of the methods’ hyperparameters, on the influence of sampling sizes, on the methods’ robustness w.r.t. slight changes in \mathbf{x}' or the underlying data, and if the hyperbox-based explanations adhere to human reasoning (user studies). While we only considered low-dimensional datasets in the benchmark study, for high-dimensional datasets we proposed two strategies to restrict the search space: either by letting users decide which features can be changed and which cannot (Sect. 2.2), or by deriving the largest local box $B \subset \underline{B}$ based on ICE curves (Sect. 4.1). Further research can explore: (1) the use of other IML methods, such as feature importance methods, to select features for which changes are investigated (all other features are set to their admissible value range); (2)

¹² The size decuples instead of doubles compared to the training data, because not all training data are $\in \underline{B}$ and, thus, not in \underline{X} .

the consideration of feature correlations or causal relations to generate IRDs, which not only naturally restricts the search space but also makes the IRD faithful to the DGP. While all presented methods are model-agnostic, we leave investigations on image and text data to future research.

Outlook. We believe that our work can also be a starting point for investigations on the application of IRDs in other fields, e.g., for hyperparameter (HP) tuning: if a promising HP set for an ML model was identified by a tuning method, IRDs can reveal its sensitivity and whether there are other equally good but more efficient HP settings. IRDs might also identify high-fidelity regions for interpretable local surrogate models, like LIME [24]. LIME approximates predictions of a black-box model $\hat{f}(\mathbf{x})$ around an observation \mathbf{x}' using a (regularized) linear model $\hat{g}(\mathbf{x})$. Here, it might be useful to understand in which region B the linear model approximates the black-box model (high-fidelity region); \hat{g} only provides valuable insights in the region B around \mathbf{x}' where $\forall \mathbf{x} \in B : \hat{h}(\mathbf{x}) := |\hat{f}(\mathbf{x}) - \hat{g}(\mathbf{x})| \leq \epsilon$ for a user-defined $\epsilon > 0$. With \hat{h} as the prediction model and $Y' = [0, \epsilon]$, IRD methods might identify such high-fidelity regions B in an interpretable manner.

Acknowledgements. This work has been partially supported by the Federal Statistical Office of Germany.

Ethical Statement. For this work, no personal data was collected or processed. Only open source datasets were used for the illustrative example and the benchmark study. Furthermore, our work does not aim at a possible use for policing or military.






References

1. Dash, S., Günlük, O., Wei, D.: Boolean decision rules via column generation. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS 2018, pp. 4660–4670. Curran Associates Inc., Red Hook, NY, USA (2018)
2. Dhurandhar, A., et al.: Explanations based on the missing: towards contrastive explanations with pertinent negatives. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS 2018, pp. 590–601. Curran Associates Inc., Red Hook, NY, USA (2018)
3. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv 1702.08608 v2, [arXiv.org](https://arxiv.org/abs/1702.08608) E-Print Archive (2017). 10.48550/arXiv.1702.08608
4. Dua, D., Graff, C.: UCI machine learning repository (2017). [www.archive.ics.uci.edu/ml/datasets/statlog+\(german+credit+data\)](http://www.archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data))
5. Eckstein, J., Hammer, P.L., Liu, Y., Nediak, M., Simeone, B.: The maximum box problem and its application to data analysis. *Comput. Optim. Appl.* **23**(3), 285–298 (2002). <https://doi.org/10.1023/a:1020546910706>
6. El Shawi, R., Sherif, Y., Al-Mallah, M., Sakr, S.: Interpretability in healthcare: a comparative study of local machine learning interpretability techniques. *Comput. Intell.* **37**(4), 1633–1650 (2021). <https://doi.org/10.1111/coin.12410>

7. Emmerich, M.T.M., Deutz, A.H., Krusselbrink, J.W.: On quality indicators for black-box level set approximation. In: Tantar, E., et al. (eds.) *EVOLVE- A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation*, pp. 157–185. Springer, Berlin (2013). https://doi.org/10.1007/978-3-642-32726-1_4
8. Fan, M., Wei, W., Xie, X., Liu, Y., Guan, X., Liu, T.: Can we trust your explanations? Sanity checks for interpreters in android malware analysis. *IEEE Tran. Inf. Forensics Secur.* **16**, 838–853 (2021). <https://doi.org/10.1109/TIFS.2020.3021924>
9. Fernandez, G., Aledo, J.A., Gamez, J.A., Puerta, J.M.: Factual and counterfactual explanations in fuzzy classification trees. *IEEE Trans. Fuzzy Syst.* **30**(12), 5484–5495 (2022). <https://doi.org/10.1109/tfuzz.2022.3179582>
10. Ferreira, L.: German credit risk (2018). www.kaggle.com/datasets/kabure/german-credit-data-with-risk. Accessed 23 Jan 2023
11. Friedman, J.H., Fisher, N.I.: Bump hunting in high-dimensional data. *Stat. Comput.* **9**(2), 123–143 (1999). <https://doi.org/10.1023/A:1008894516817>
12. Fürnkranz, J., Kliegr, T.: A brief overview of rule learning. In: Bassiliades, N., Gotlob, G., Sadri, F., Paschke, A., Roman, D. (eds.) *RuleML 2015*. LNCS, vol. 9202, pp. 54–69. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21542-6_4
13. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat.* **24**(1), 44–65 (2015). <https://doi.org/10.1080/10618600.2014.907095>
14. Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., Turini, F.: Factual and counterfactual explanations for black box decision making. *IEEE Intell. Syst.* **34**(6), 14–23 (2019). <https://doi.org/10.1109/MIS.2019.2957223>
15. Guidotti, R., Monreale, A., Ruggieri, S., Naretto, F., Turini, F., Pedreschi, D., Giannotti, F.: Stable and actionable explanations of black-box models through factual and counterfactual rules. *Data Min. Knowl. Disc.* (2022). <https://doi.org/10.1007/s10618-022-00878-5>
16. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local rule-based explanations of black box decision systems. *arXiv 1805.10820*, [arXiv.org E-Print Archive](https://arxiv.org/abs/1805.10820) (2018). 10.48550/arXiv.1805.10820
17. Kenny, E.M., Keane, M.T.: On generating plausible counterfactual and semi-factual explanations for deep learning. *Proc. AAAI Conf. Artif. Intell.* **35**(13), 11575–11585 (2021). <https://doi.org/10.1609/aaai.v35i13.17377>
18. Khuat, T.T., Ruta, D., Gabrys, B.: Hyperbox-based machine learning algorithms: a comprehensive survey. *Soft Comput.* **25**(2), 1325–1363 (2020). <https://doi.org/10.1007/s00500-020-05226-7>
19. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv 1412.6980 v9*, [arXiv.org E-Print Archive](https://arxiv.org/abs/1412.6980) (2017). 10.48550/arXiv.1412.6980
20. Kuratomi, A., Miliou, I., Lee, Z., Lindgren, T., Papapetrou, P.: JUICE: JUStified counterfactual explanations. In: Pascal, P., Ienco, D. (eds.) *Discovery Science*. pp. 493–508. LNCS, Springer, Cham (2022). https://doi.org/10.1007/978-3-031-18840-4_35
21. Land, A.H., Doig, A.G.: An automatic method of solving discrete programming problems. *Econometrica* **28**(3), 497–520 (1960). <https://doi.org/10.2307/1910129>
22. Lemhadri, I., Li, H.H., Hastie, T.: RbX: region-based explanations of prediction models. *arXiv 2210.08721*, [arXiv.org E-Print Archive](https://arxiv.org/abs/2210.08721) (2022). 10.48550/arXiv.2210.08721
23. Nugent, C., Doyle, D., Cunningham, P.: Gaining insight through case-based explanation. *J. Intell. Inf. Syst.* **32**(3), 267–295 (2009). <https://doi.org/10.1007/s10844-008-0069-0>

24. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you? Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)
25. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1 (2018). <https://doi.org/10.1609/aaai.v32i1.11491>
26. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchor. Github repository. www.github.com/marcotcr/anchor (2022), Commit: b1f5e6ca37428613723597e85c38558e8cd21c2e
27. Schwartzberg, C., van Engers, T.M., Li, Y.: The fidelity of global surrogates in interpretable machine learning. BNAIC/BeneLearn 2020 (2020)
28. Sharma, R., Reddy, N., Kamakshi, V., Krishnan, N.C., Jain, S.: MAIRE - a model-agnostic interpretable rule extraction procedure for explaining classifiers. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) CD-MAKE 2021. LNCS, vol. 12844, pp. 329–349. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84060-0_21
29. Stepin, I., Alonso, J.M., Catala, A., Pereira-Fariña, M.: Generation and evaluation of factual and counterfactual explanations for decision trees and fuzzy rule-based classifiers. In: 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–8. IEEE, Glasgow, United Kingdom (2020). <https://doi.org/10.1109/FUZZ48607.2020.9177629>
30. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: OpenML: networked science in machine learning. SIGKDD Explor. Newsl. **15**(2), 49–60 (2014). <https://doi.org/10.1145/2641190.2641198>
31. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harvard J. Law Technol. **31**(2), 841–887 (2018)
32. Zabinsky, Z.B., Huang, H.: A partition-based optimization approach for level set approximation: probabilistic branch and bound. In: Smith, A.E. (ed.) Women in Industrial and Systems Engineering. WES, pp. 113–155. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-11866-2_6
33. Zabinsky, Z.B., Wang, W., Prasetio, Y., Ghate, A., Yen, J.W.: Adaptive probabilistic branch and bound for level set approximation. In: Proceedings of the 2011 Winter Simulation Conference (WSC), pp. 4146–4157. IEEE, Phoenix, AZ, USA (2011). <https://doi.org/10.1109/WSC.2011.6148103>

Supplementary Material: Interpretable Regional Descriptors: Hyperbox-Based Local Explanations

Susanne Dandl^{1,2} , Giuseppe Casalicchio^{1,2} ,
Bernd Bischl^{1,2} , and Ludwig Bothmann ^{1,2} 

¹ Department of Statistics, LMU Munich, Ludwigstr. 33, 80539 Munich, Germany

² Munich Center for Machine Learning (MCML), Munich, Germany
Ludwig.Bothmann@stat.uni-muenchen.de

A Application Examples

In addition to the credit application in Section 1, we show in the following a medical and jurisdictional application.

Medical Consider an ML model that predicts if a person will develop diabetes in the future. (For simplicity, we assume this model accurately approximates real world relationships.) In the following, we discuss two cases:

(1) A person that is predicted to develop diabetes wants to know why this is the case and what can be options to prevent this. There are different potential actions to take: more sport, less red meat, homeopathic medicine, etc. The IRD can tell which action is not promising, e.g., sports when all realistic amounts of sport are inside the box. However, changing the diet might be an option, because changing the diet by just eating meat one day a week is not part of the box (concrete strategies for prevention can reveal counterfactual explanations).

(2) A person that is predicted not to develop diabetes wants to know how flexible their life-style is without changing the prediction. It may be okay for a person to gain weight without having a higher risk of developing diabetes, as long as they do not change their diet towards including more red meat.

Jurisdiction Consider an ML model that predicts if a person will commit a crime in the next 2 years. A person that gets a high score wants to know why. IRDs that do not contain all groups of protected attributes, such as gender, can indicate unfair discrimination against these groups. Hence, IRDs can initiate further investigations on fairness and biases of an ML model.

B Proof of Theorem 1

Proof. Given a feature X_j that is not involved in the prediction model \hat{f} such that $\forall \tilde{\mathbf{x}} \in \mathcal{X} \wedge \forall x_j \in \mathcal{X}_j$:

$$\hat{f}(\tilde{x}_1, \dots, \tilde{x}_{j-1}, \tilde{x}_j, \tilde{x}_{j+1}, \dots, \tilde{x}_p) = \hat{f}(\tilde{x}_1, \dots, \tilde{x}_{j-1}, x_j, \tilde{x}_{j+1}, \dots, \tilde{x}_p), \quad (1)$$

2 S. Dandl et al.

and given a box B for \mathbf{x}' that is maximal according to Definition 1. We assume now that Theorem 1 does not hold such that $B_j = [l_j, u_j] \subset \mathcal{X}_j$. However, since Eq. (1) holds, either $(\exists x_j \in \mathcal{X}_j \wedge x_j < l_j : \text{precision}(B \cup [x_j, l_j]) = 1)$, or $(\exists x_j \in \mathcal{X}_j \wedge x_j > u_j : \text{precision}(B \cup [u_j, x_j]) = 1)$ for numeric \mathcal{X}_j or $(\exists x_j \in \mathcal{X}_j \setminus B_j : \text{precision}(B \cup x_j) = 1)$ for categorical \mathcal{X}_j holds which contradicts the maximality assumption of B .

C Proof of Theorem 2

Proof. Given a box B with $\text{precision}(B) = 1$ and $\mathbf{x}' \in B$, and given a feature X_j that is relevant for $\hat{f}(x')$ such that $\exists x_j \in \mathcal{X}_j \setminus B_j : \hat{f}(x'_1, \dots, x'_{j-1}, x_j, x'_{j+1}, \dots, x'_p) \notin Y'$. We assume now that Theorem 2 does not hold, such that $B_j = \mathcal{X}_j$. This contradicts the statement that $\text{precision}(B) = 1$ because x_j that leads to a prediction $\notin Y'$ for \mathbf{x}' is also covered by the box.

D Proof of Theorem 3

Proof. Without loss of generality, we assume that we only have numeric features. Assume we computed $\bar{B} = \bigcup_{j=1}^p [l_j, u_j]$ such that $\forall j \in \{1, \dots, p\}$:

$$\underbrace{\hat{f}(x'_1, \dots, x'_{j-1}, l_j, x'_{j+1}, \dots, x'_p)}_{:=\mathbf{x}'_l} \notin Y' \wedge \underbrace{\hat{f}(x'_1, \dots, x'_{j-1}, u_j, x'_{j+1}, \dots, x'_p)}_{:=\mathbf{x}'_u} \notin Y'.$$

We assume that $B \subset \bar{B}$ is not true for now such that there is a homogeneous B with $\min(B_j) < l_j$ or $\max(B_j) > u_j$ and $\mathbf{x}' \in B$. However, then either \mathbf{x}'_l or \mathbf{x}'_u would also be part of B but for both $\hat{f}(\mathbf{x}'_u) \notin Y'$ or $\hat{f}(\mathbf{x}'_l) \notin Y'$ holds, which contradicts that B is homogeneous.

E Pseudocode and Illustrations of IRD Methods

E.1 Pseudocode

Algorithm 1 Adapted MaxBox approach [2]

Input: Targeted instance \mathbf{x}' , desired range Y' , prediction model $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$, input dataset $\tilde{\mathbf{X}}$, initial box B
Initialize candidates = [], upper_bound_coverage_best = -Inf, current_best = []
if $\exists \mathbf{x} \in \tilde{\mathbf{X}} \wedge \mathbf{x} \in B : \hat{f} \notin Y'$ **then**
 candidates = candidates.append(B)
 while length(candidates) > 0 **do**
 $B^{best} = choose_best(candidates)$
 ▷ if upper_bound_coverage_best < 0, B^{best} corresponds to the box with the most no. of shrinking steps done before (with the upper bound of the coverage as a tiebreaker), else, B^{best} corresponds to the box that maximizes $\frac{|\{\mathbf{x} \in B | \hat{f}(\mathbf{x}) \in Y'\}|}{|\{\mathbf{x} \in B | \hat{f}(\mathbf{x}) \notin Y'\}|}$.
 candidates = candidates.remove(B^{best})
 children = create_new_candidates(B^{best}) ▷ in Figure S. 1, C and D are new candidates created from the initial box
 for $B \in children$ **do**
 if $\forall \mathbf{x} \in B : \hat{f}(\mathbf{x}) \in Y'$ **then**
 coverage = upper_bound_coverage(B)
 if coverage > upper_bound_coverage_best **then**
 current_best = B
 upper_bound_coverage_best = coverage
 end if
 else
 if upper_bound_coverage(B) > upper_bound_coverage_best **then**
 candidates = candidates.append(B)
 end if
 end if
 end for
 end while
else
 current_best = B
end if
return current_best

4 S. Dandl et al.

Algorithm 2 Adapted PRIM approach [3]

Input: Targeted instance \mathbf{x}' , desired range Y' , prediction model $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$, input dataset $\tilde{\mathbf{X}}$, initial box B

while $\exists \mathbf{x} \in \tilde{\mathbf{X}} \wedge \mathbf{x} \in B : \hat{f} \notin Y'$ **do**

for $j \in \{1, \dots, p\}$ **do**

$C_j = []$ ▷ create candidates for peeling

if X_j **numeric then**

$C_j = C_j.append(B_j^-, B_j^+)$ where $B_j^- = [l_j, \min(X_{j(\alpha)}, x'_j)]$ and $B_j^+ = [\max(X_{j(1-\alpha)}, x'_j), u_j]$ with $x_{j(\alpha)}$ and $x_{j(1-\alpha)}$ as the α - and $(1-\alpha)$ -quantiles of X_j in the current box B

else if X_j **categorical then**

$C_j = \{s \in B_j \mid s \neq x'_j\}$

end if

end for

$b^{best} = \arg \max_{b \in C_j, j \in \{1, \dots, p\}} precision(B \setminus b)$

$B = B \setminus b^{best}$

end while

homogeneous = TRUE

while homogeneous **do**

for $j \in \{1, \dots, p\}$ **do**

$C_j = []$ ▷ create candidates for pasting

if X_j **numeric then**

 inbox = $\{\mathbf{x} \in \tilde{\mathbf{X}} \mid x_k \in B_k\}$, for $k \in \{1, \dots, j-1, j+1, \dots, p\}$

 number_added = $|\{\mathbf{x} \in \tilde{\mathbf{X}} \mid \mathbf{x} \in B\}| \cdot \alpha$

$C_j = C_j.append(B_j^-, B_j^+)$ with $B_j^- = [x_j^l, l_j]$ and $B_j^+ = [u_j, x_j^u]$ with x_j^l as the j th feature value of the (number_added)th observation $\mathbf{x} \in$ inbox with a value x_j lower than l_j and x_j^u as the j th feature value of the (number_added)th observation $\mathbf{x} \in$ inbox with a value x_j higher than u_j

else if X_j **categorical then**

$C_j = \{s \in X_j \mid s \notin B_j\}$

end if

$C_j = \{b \in C_j \mid precision(B \cup b) = 1\}$

end for

if $\exists j \in \{1, \dots, p\} : |C_j| > 0$ **then**

$b^{best} = \arg \max_{b \in C_j, j \in \{1, \dots, p\}} coverage(B \setminus b)$

$B = B \cup b$

else

 homogeneous = FALSE

end if

end while

return B

Algorithm 3 Adapted MAIRE approach [7]

Input: Targeted instance \mathbf{x}' , desired range Y' , prediction model $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$, input dataset $\bar{\mathbf{X}}$, initial box B , precision threshold τ (default 1), maximum number of iterations max_iterations (default 100)
Scale all feature values of $\mathbf{x} \in \bar{\mathbf{X}}$ and \mathbf{x}' to 0-1 range
 $\text{best_coverage} = 0$
 $\text{converged} = \text{FALSE}$
 $\text{best_candidate} = B$
 $i = 0$
while $i \leq \text{max_iterations}$ **do**
 $B = \text{optimize_with_adam}(B)$
 \triangleright optimizes differentiable versions of coverage, precision and locality
 if $\text{precision}(B) \geq \tau \wedge \text{coverage}(B) \geq \text{best_coverage}$ **then**
 $\text{best_candidate} = B$
 else if $\text{precision}(B) < \tau$ **then**
 $\text{converged} = \text{TRUE}$
 end if
 if $\text{converged} = \text{TRUE}$ **then**
 $i = i + 1$
 end if
end while
return best_candidate

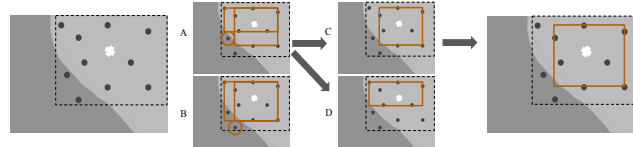
E.2 Illustrations

Fig. S. 1: Illustration of the adapted MaxBox algorithm. The algorithm starts with \bar{B} (dashed box). In the box are two data points with predictions $\notin Y'$ (called negative samples) and the box needs to be further optimized. First, a negative sample is chosen - either the one in A or B. Therefore, the number of samples with predictions $\in Y'$ after excluding the points in one feature dimension are inspected. The resulting boxes of both negative samples cover a maximum of seven samples. We chose the one of A (B is also fine). Its resulting boxes are the new subproblems/candidates (C and D). Both boxes in C and D only include samples with predictions $\in Y'$, but the box in C is chosen as an optimum because it includes more samples with predictions $\in Y'$. D is discarded because it has a lower number. Since C and D cannot be further split because no negative samples are within both boxes, the returned box by MaxBox is the box in C.

6 S. Dandl et al.

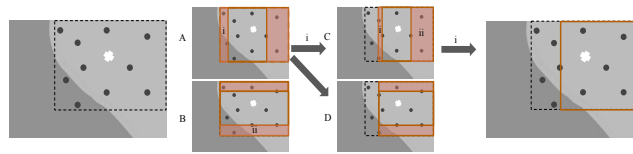


Fig. S. 2: Illustration of the adapted PRIM algorithm. The algorithm starts with \bar{B} . In the first iteration, there exist four potential subboxes (two in each feature dimension (A vs. B)) that could be removed. The subbox i is chosen because it has the highest precision but compared to ii it has a smaller size. In the next step (C & D), again four subboxes can be potentially removed. Again, we choose i for the same reason as before. After its removal, the resulting box is at the same time the final box because in the pasting step only one subbox could be added - i again. All other dimensions are maximal.

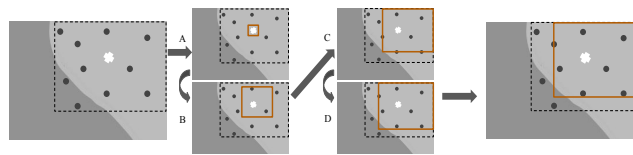


Fig. S. 3: Illustration of the adapted MAIRE algorithm. The algorithm starts with the smallest box possible. The box boundaries are then iteratively enlarged (A-D). The box boundaries are only updated if the precision of the new box = 1.

F Pseudocode of Post-Processing Approach

Algorithm 4 Post-processing algorithm - peeling (inspired by [3])

Input: Targeted instance \mathbf{x}' , desired range Y' , prediction model $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$, initial box B , number of samples for evaluation M (default 100), relative subbox size of continuous features α (default 0.1)

for $j \in \{1, \dots, p\}$ **do**

if X_j numeric **then**

$s_j = (\max(\mathcal{X}_j) - \min(\mathcal{X}_j)) \cdot \alpha$ ▷ derive subbox sizes for numeric features based on \mathcal{X}

if X_j integer **then**

$s_j = \text{round}(s_j)$

end if

end for

$\bar{\mathbf{X}} = \text{sample_uniformly}(B, n = M \cdot 5)$ ▷ sample new data to check if B homogeneous

if $\exists \mathbf{x} \in \bar{\mathbf{X}} \wedge \mathbf{x} \in B : \hat{f} \notin Y'$ **then**

 not_homogeneous = TRUE ▷ start peeling

while not_homogeneous **do**

for $j \in \{1, \dots, p\}$ **do**

$C_j = []$ ▷ create candidates for peeling

if X_j numeric **then**

$C_j = C_j.\text{append}(B_j^-, B_j^+)$

 where $B_j^- = [l_j, \min(l_j + s_j, x'_j)]$ and $B_j^+ = [\max(u_j - s_j, x'_j), u_j]$

else if X_j categorical **then**

$C_j = \{s \in B_j \mid s \neq x'_j\}$

end if

$C_j = \{b \in C_j \mid \text{precision}(B_j^b) < 1\}$ with $B_j^b = (B_1 \times \dots \times B_{j-1} \times b \times B_{j+1} \times \dots \times B_p)$

end for

if $\exists j \in \{1, \dots, p\} : |C_j| > 0$ **then**

$b^{\text{best}} = \arg \max_{b \in C_j, j \in \{1, \dots, p\}} \text{precision_to_boxsize}(B_j^b)$ ▷ evaluate on M new instances sampled within B_j^b

$B^{\text{best}} = (B_1 \times \dots \times B_{j-1} \times b^{\text{best}} \times B_{j+1} \times \dots \times B_p)$ ▷ choose the one with lowest precision relative to size

$B = B^{\text{best}}$

else

 not_homogeneous = FALSE

end if

end while

end if

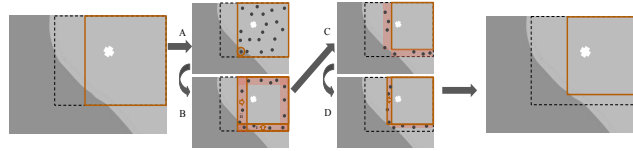
return $B, \mathbf{s} = \{s_j \mid X_j \text{ numeric}\}$

8 S. Dandl et al.

Algorithm 5 Post-processing algorithm - pasting (inspired by [3])

Input: Targeted instance \mathbf{x}' , desired range Y' , prediction model $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$, initial box B (potentially peeled), number of samples for evaluation M (default 100), relative subbox size of continuous features α (default 0.1), lower threshold for relative subbox size α_0 (default 0.05), subbox sizes of numeric features \mathbf{s}
 homogeneous = TRUE ▷ start pasting
 stepsize = 1
while homogeneous **do**
 for $j \in \{1, \dots, p\}$ **do**
 $C_j = []$ ▷ create candidates/subboxes for pasting
 if X_j numeric **then**
 $C_j = C_j.append(B_j^-, B_j^+)$
 where $B_j^- = [l_j - stepsize \cdot s_j, l_j]$ and $B_j^+ = [u_j, u_j + stepsize \cdot s_j]$
 else if X_j categorical **then**
 $C_j = \{s \in X_j \mid s \notin B_j\}$
 end if
 $C_j = \{b \in C_j \mid precision(B_j^b) = 1\}$ with $B_j^b = (B_1 \times \dots \times B_{j-1} \times b \times B_{j+1} \times \dots \times B_p)$
 end for
 if $\exists j \in \{1, \dots, p\} : |C_j| > 0$ **then**
 $b^{best} = \arg \max_{b \in C_j, j \in \{1, \dots, p\}} size(B_j^b)$ ▷ evaluate on M new instances sampled within
 B_j^b
 $B = B \cup b$ ▷ choose largest one with precision 1
 else
 if stepsize $\geq \alpha_0$ **then**
 stepsize = stepsize/2 ▷ if no box with precision 1 exists,
 consider reducing the subbox sizes
 else
 homogeneous = FALSE
 end if
 end while
return B

Fig. S. 4: Illustration of the post-processing algorithm. The algorithm starts with the box generated by another method (solid brown box, which is a subbox of the dashed box \bar{B}). First, new points are sampled and it is assessed whether the box is homogeneous (A). If not, the subboxes with the lowest precision compared to their size are peeled iteratively (B). The precision is assessed based on newly sampled points within the subboxes. First subbox i is peeled then subbox ii (both contain a sample with a prediction $\notin Y'$). If no subbox with precision < 1 exists, it is assessed whether the box could be further enlarged (C). If all considered subboxes have precisions < 1 , the subbox sizes are halved (D) as long as the relative subbox size does not fall below a threshold.



G Level Set Identification

The algorithm by Kuratomi et al. [5] starts at \mathbf{x}' and tries to find a connection $\in Y'$ between the nominal, then the ordinal, and then the continuous features of \mathbf{x} and \mathbf{x}' . If a path is found, \mathbf{x} is part of \mathcal{L} . For categorical features, all permutations of feature orders are inspected.¹ For continuous features, the shortest linear path for a given number of equidistant steps is checked. Kuratomi et al. [5] used DBSCAN, for which the choice of the maximum distance threshold is ambiguous. The identification algorithm has a complexity of $O(c! \cdot c + o! \cdot \sum_{j=1}^o k_j + q)$ with c and o as the number of nominal and ordinal features, respectively, k_j as the number of possible values of an ordinal feature X_j and q as the number of inspected steps for continuous features.

The level set could be further enriched by attempting to find connections between the unconnected and connected points. For the comparison of IRD methods, however, a convex level set is sufficient, since the hyperbox itself is convex.

H Tuning of ML models

For hyperparameter tuning, we used random search (with 15 evaluations), and 5-fold cross-validation (CV) with the misclassification error (classification) or mean squared error (regression) as a performance measure. Table S. 1 shows the tuning search space of each model. The rather limited tuning setup should be sufficient

¹ If the number of permutations exceeds 100 permutations, 100 feature orders are randomly chosen.

10 S. Dandl et al.

for our task of explaining a prediction model – a less accurate model is not a hindrance. Unbalanced datasets such as *tic_tac_toe*, *diabetes* and *cmc* were balanced with the SMOTE algorithm [1]. For SMOTE, numeric features were standardized and categorical ones were one-hot encoded. The optimizer for the neural network was ADAM [4] with 500 epochs. For all other hyperparameters, the default values of the mlr3keras R package were used [6] (apart from the no. of layer units, see Table S. 1). Table S. 2 shows the accuracies of each model using nested resampling with 5-fold CV in the inner and outer loop).

Table S. 1: Tuning search space of each model. Hyperparameter values of *num.trees* were log-transformed.

Model	Hyperparameter	Range
random forest	num.trees	[1, 1000]
logistic regression	-	-
linear model	-	-
multi-nomial model	-	-
hyperbox/rpart	-	-
neural net	layer_units	[1, 20]

Table S. 2: Classification error or mean squared error (regression) of each model on each dataset. The performances were computed using nested resampling with 5-fold CV in the inner and outer loop. We did not measure the performance of the (terminal node) hyperbox model because the model differs for each x' .

	Random forest	Linear model	Neural net	Hyperbox
diabetes	0.233	0.224	0.229	-
tic_tac_toe	0.036	0.019	0.094	-
cmc	0.466	0.495	0.389	-
vehicle	0.256	0.201	0.254	-
no2	33502.856	37678.319	77866.331	-
plasma_retinol	45391.218	59224.452	297481.249	-

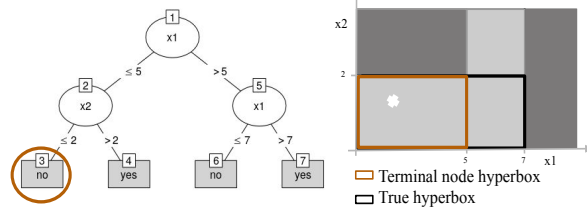


Fig. S. 5: True hyperbox vs. terminal node hyperbox for a CART tree. The white cross corresponds to x' .

I Benchmark - Additional Results

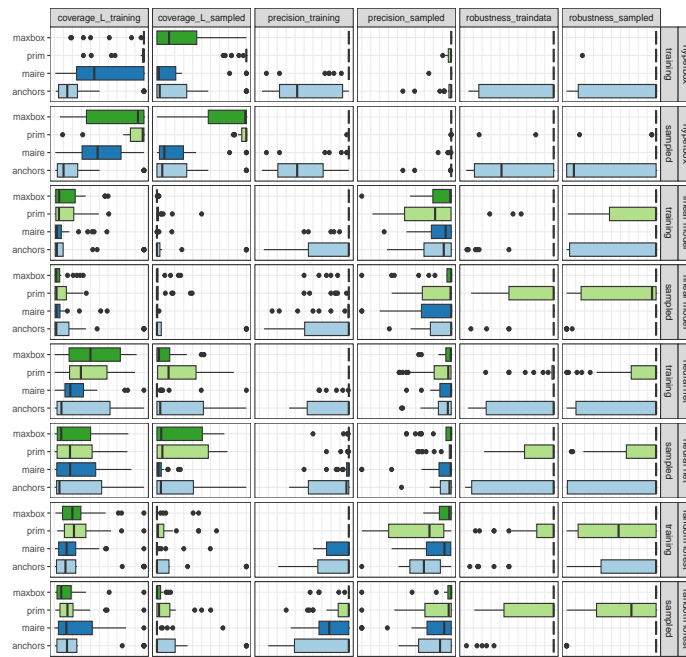


Fig. S. 6: Comparison of MaxBox, PRIM, Anchors, and MAIRE w.r.t. coverage and precision for each model separately. Each method was either run or evaluated on training data or uniformly sampled data from \bar{B} without post-processing. Higher values for precision and coverage are better.

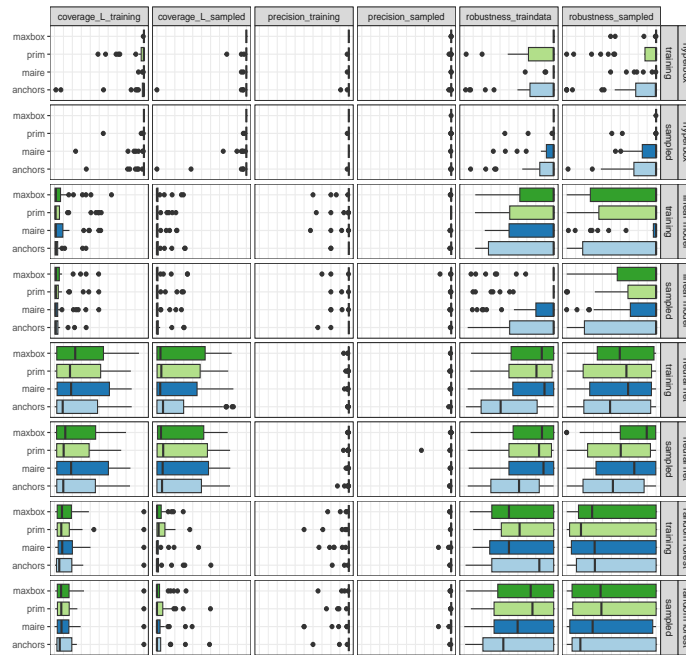


Fig. S. 7: Comparison of MaxBox, PRIM, Anchors, and MAIRE w.r.t. coverage and precision for each model separately. Each method was either run or evaluated on training data or uniformly sampled data from \bar{B} with post-processing. Higher values for precision and coverage are better.

14 S. Dandl et al.

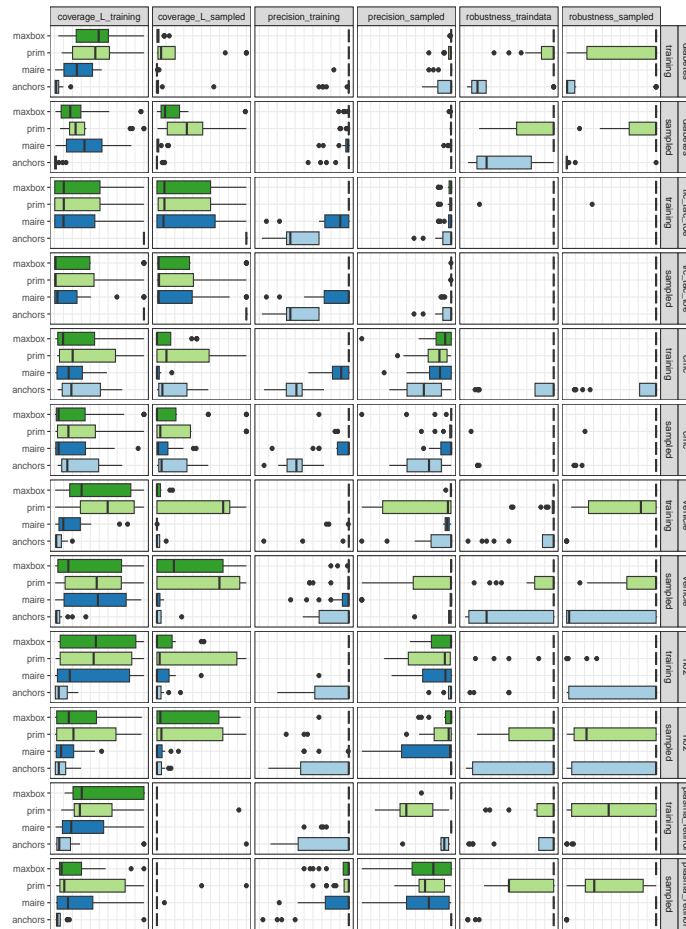


Fig. S. 8: Comparison of MaxBox, PRIM, Anchors, and MAIRE w.r.t. coverage and precision for each dataset separately. Each method was either run or evaluated on training data or uniformly sampled data from \mathbb{B} without post-processing. Higher values for precision and coverage are better.

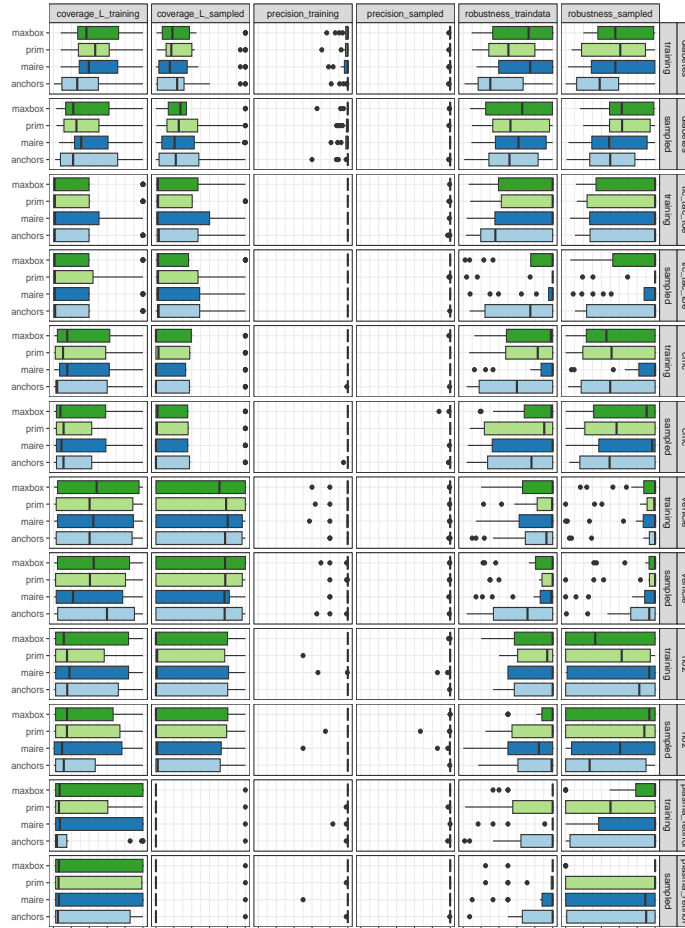


Fig. S. 9: Comparison of MaxBox, PRIM, Anchors, and MAIRE w.r.t. coverage and precision for each dataset separately. Each method was either run or evaluated on training data or uniformly sampled data from \bar{B} with post-processing. Higher values for precision and coverage are better.

Table S. 3: Statistical analysis of RQ 1. Pairwise comparison of MaxBox, PRIM, Anchors, and MAIRE w.r.t. coverage and precision. Each value corresponds to the p-value obtained for the Wilcoxon rank-sum test with H_0 that the performances do not differ. Cells printed in bold font correspond to p-values that are lower than $\alpha = 0.05/36$ (Bonferroni-adjustment) and indicate that one method outperforms the other. Only methods run on the training data without post-processing were compared.

measure	MaxBox = PRIM	MaxBox = Anchors	MaxBox = MAIRE	PRIM = Anchors	PRIM = MAIRE	Anchors = MAIRE
coverage_training	0.761	0.618	0	0.579	0	0.473
coverage_sampled	0	0.044	0	0	0	0
coverage_L_training	0.431	0.001	0	0	0	0.127
coverage_L_sampled	0	0.035	0.004	0.059	0	0
precision_training	1	0	0	0	0	0
precision_sampled	0.025	0	0.623	0.042	0.104	0.004

Table S. 4: Statistical analysis of RQ 2. Pairwise comparison of using training data vs. sampled data for \bar{X} . Each value corresponds to the p-value obtained for the Wilcoxon rank-sum test with H_0 that the performance of methods using training data is better than the performance of methods using sampled data. Cells printed in bold font correspond to p-values that are lower than $\alpha = 0.05/30$ (Bonferroni-adjustment) and indicate a preference towards using sampled data. Comparisons were only conducted for the methods run without post-processing.

measure	overall	MaxBox	PRIM	Anchors	MAIRE
coverage_training	1.00	1	0.999	0.445	0.744
coverage_sampled	0.00	0	0.987	0.402	0.003
coverage_L_training	1.00	1	1	0.781	0.896
coverage_L_sampled	0.00	0	0.236	0.476	0.172
precision_training	1.00	0.995	0.998	0.782	0.993
precision_sampled	0.00	0.011	0	0.001	0.381

Table S. 5: Statistical analysis of RQ 3. Pairwise comparison of using no post-processing vs. using post-processing. Each value corresponds to the p-value obtained for the Wilcoxon rank-sum test with H_0 that the performance of methods using no post-processing is better than the performance of methods using post-processing. Cells printed in bold font correspond to p-values that are lower than $\alpha = 0.05/60$ (Bonferroni-adjustment) and indicate a preference towards post-processing.

method	coverage_	training_coverage_	sampled_coverage_	L_	training_coverage_	L_	sampled_precision_	training_precision_	sampled_precision_
traindata	0.95	0	0.369	0	0	0	0	0	0
MaxBox	0.97	0	0.982	0	0	0.995	0.999	0	0.003
PRIM	1.00	1	0.452	0.452	0.999	0	0	0	0
anchors	0.92	0.001	0.065	0.054	0	0	0	0	0
MAIRE	0.10	0	0.003	0	0	0	0	0	0.001
sampled	0.12	0	0	0	0	0	0	0	0
MaxBox	0.00	0	0.262	0	0	0.085	0.003	0.021	0.001
PRIM	0.45	0.19	0.035	0.468	0.061	0	0	0	0
anchors	0.92	0	0.003	0	0	0	0	0	0
MAIRE	0.18	0	0.003	0	0	0	0	0	0.009

18 S. Dandl et al.

References

1. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* **16**, 321–357 (2002). <https://doi.org/10.1613/jair.953>
2. Eckstein, J., Hammer, P.L., Liu, Y., Nediak, M., Simeone, B.: The maximum box problem and its application to data analysis. *Computational Optimization and Applications* **23**(3), 285–298 (2002). <https://doi.org/10.1023/a:1020546910706>
3. Friedman, J.H., Fisher, N.I.: Bump hunting in high-dimensional data. *Statistics and Computing* **9**(2), 123–143 (1999). <https://doi.org/10.1023/A:1008894516817>
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv 1412.6980 v9, arXiv.org E-Print Archive (2017). <https://doi.org/10.48550/arXiv.1412.6980>
5. Kuratomi, A., Miliou, I., Lee, Z., Lindgren, T., Papapetrou, P.: JUICE: JUStified Counterfactual Explanations. In: Pascal, P., Ienco, D. (eds.) *Discovery Science*. pp. 493–508. *Lecture Notes in Computer Science*, Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-18840-4_35
6. Pfisterer, F., Poon, J., Lang, M.: mlr3keras: mlr3 keras extension. Github repository. URL <https://github.com/mlr-org/mlr3keras> (2022), Commit: bad8434b7898b51b2143fc680594057c00dc7080
7. Sharma, R., Reddy, N., Kamakshi, V., Krishnan, N.C., Jain, S.: MAIRE - a model-agnostic interpretable rule extraction procedure for explaining classifiers. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) *Machine Learning and Knowledge Extraction*, vol. 12844, pp. 329–349. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-84060-0_21, Series Title: *Lecture Notes in Computer Science*

Part III

Conclusion and Outlook

12 Conclusion and Outlook

This thesis comprises seven articles that address causality concepts in machine learning to enhance HTE estimation and model interpretation. The exploration of RF-based approaches – MOB and CFs – unraveled what elements benefit HTE estimation and how they can be combined to be applicable to a wide range of use cases beyond randomized trials and continuous outcomes.

For model interpretation, the thesis reviewed methods for SFEs and CFEs and introduced two methods to deal with the multiplicity of (equally good) explanations – one of the many pitfalls of post-hoc interpretation methods. The proposed multi-objective CFE method (MOC) returns a set of counterfactuals that reflect different trade-offs between the desired properties of CFEs. The method can be flexibly applied to other use cases (e.g., counterfactual fairness) and is available in a modular and user-friendly R package. The proposed hyperbox-based interpretation method – interpretable regional descriptors – offers a summary of SFEs and opens new exciting research paths for further application.

These contributions still leave unanswered questions and allude to unexplored areas that need to be addressed in future research. Some were stated in the contributing articles’ respective conclusions and outlook sections. The following briefly touches upon some of these open points and mentions a few in addition (without claiming to be complete).

Treatment Effect Estimation: Exploring Violations of Assumptions

Section 3.1.2 provided an overview of the identifying assumptions that allowed the causal treatment effect estimand to be transformed into a statistical estimand. Section 3.1.2 showed that it is easier to justify these assumptions for randomized trials than for observational studies. The literature proposed multiple methods for dealing with violations: For violations of Assumption 1 (unconfoundedness), instrumental variables can help (Angrist *et al.*, 1996); for violations of Assumption 2 (positivity), trimming might offer a solution (Crump *et al.*, 2009); for violations of Assumption 3 (no interference), tailored estimation procedures exist (Hudgens and Halloran, 2008); violations of Assumption 4 (consistency) can be circumvented by allowing multiple versions of treatment in the POF (VanderWeele and Hernán, 2013).

In Section 2 of the contribution in Chapter 6, an additional assumption was stated: “[W]e assume that \mathbf{X} includes all relevant variables to explain heterogeneity both in the treatment effect and the outcome Y and that the base model underlying model-based forests is correctly specified”. This was necessary to circumvent model misspecifications that are a problem if the underlying model of MOB is noncollapsible. Noncollapsibility of a model means that, for a given X , the mean of the conditional treatment effects is not equal to the marginal treatment effect. Due to this model characteristic, misspecifications in the model cannot be absorbed by the error term, which affects the estimation of $\tau(\mathbf{x})$, inhibiting its interpretation as a causal effect. Examples of noncollapsible

models are the Cox model and members of the exponential family without an identity or log link (Greenland *et al.*, 1999; Aalen *et al.*, 2015).

Problems with noncollapsibility can arise under misspecifications of the prognostic effect $\mu(\mathbf{x})$. These misspecifications may arise because the estimation of $\mu(\mathbf{x})$ is ignored by focusing only on $\tau(\mathbf{x})$, because the complexity of $\mu(\mathbf{x})$ is underestimated, or because not all prognostic variables are observed. For the first two causes, MOBs offer a solution: MOBs simultaneously focus on heterogeneity in treatment *and* prognostic effects, and the tree-ensemble can model complex relations between prognostic variables \mathbf{X} and Y .

For the last reason (lack of knowledge of all prognostic variables), no solution exists. That is why the contributing article of Chapter 6 assumes that all prognostic variables are known. Future research can investigate the severity of violations of the assumption and potential mitigation techniques. Appendix A of the contributing article in Chapter 6 analyzed a technique by Gao and Hastie (2022) against misspecifications but found no improvement in performance in a simulation study when omitting a prognostic variable.

Counterfactual & Semi-factual Explanations: Exploring Synergies

As seen in Chapter 4, CFEs and SFEs are both based on causal counterfactuals, and they have multiple desired properties in common (Section 4.2.1 and 4.3.1). Consequently, there might be synergies between their generation, and future work can evaluate whether CFE methods can be adapted to generate SFEs. Such investigations are valuable because many CFE methods were proposed in the last few years, while only a few were proposed for generating SFEs. Current SFE methods have disadvantages, as seen in Section 4.3.2: the majority only return a single SFE, and the only method that generates a set (Artelt and Hammer, 2022) neglects the plausibility property and does not consider trade-offs between the objectives.

For selecting suitable CFE methods, other desired properties beyond the ones of Section 4.2.2 exist, which have received less attention in research so far. For example, the methods should be robust (such that small changes in the inputs, underlying data, or hyperparameters lead to similar CFEs), and efficient (in the sense that few calls to \hat{f} and a low computational time are required). Optimally, these methods generate CFEs for multiple input data points simultaneously or reuse knowledge from previous runs. MOC, which was introduced in the contribution of Chapter 8, also has room for improvement in these aspects.

Adapting CFE methods to SFEs requires some further considerations w.r.t. the distance property: On the one hand, SFEs should be similar to \mathbf{x}^* because, otherwise, the SFE would no longer display a reachable, alternative world. On the other hand, SFEs that largely differ from \mathbf{x}^* in a few features are more convincing (see Section 4.3). To my knowledge, no previous work properly formalized this property. A suitable requirement can be that the SFE maximally differs in a few (selected) features to \mathbf{x}^* while being part of the level set of \mathbf{x}^* . An observation is part of the local level set if itself and all intermediate points on the path between \mathbf{x} and \mathbf{x}^* have the same prediction as \mathbf{x}^* (Definition 2 in the contribution of Chapter 11). All SFEs in an Interpretable Regional Descriptor (proposed in the contributing article of Chapter 11) are, by design, part of the local level set. However, due to the hyperbox shape, the IRD might not cover the whole local level set, so maximal distances to \mathbf{x}^* in a few features cannot be guaranteed. Therefore, further research is required to formalize and methodically implement the distance property.

Applying Interpretation Methods to Treatment Effect Estimators

As stated in Section 4.2.2, some methods for CFEs consider causal relations denoted in a (partially known) causal graph to derive more realistic explanations. This principle has also been applied to other interpretation methods like Shapley values (Heskes *et al.*, 2020), surrogate models (Cinquini and Guidotti, 2023) or partial dependence plots (Loftus *et al.*, 2023). What has been less discussed in research so far is the reverse: applying interpretation methods to obtain insights into treatment effect estimators. This would be especially valuable since one of the most prominent application fields of HTE estimation is the sensitive domain of medicine.

Most proposed interpretation methods for HTE estimators are model-specific: Crabbé *et al.* (2022) applied feature importance methods only to neural networks. Likewise, the implemented variable importance methods in the **grf** and **model4you** R packages are tailored to forest-based methods (Tibshirani *et al.*, 2023; Seibold *et al.*, 2021). A model-agnostic interpretation method is the dependence plot, which was also applied in the contributions of Chapters 5 and 6 based on the work of Seibold *et al.* (2018). It plots the estimated (out-of-bag) treatment effects against the feature values of the training data. A smooth curve calculated by a generalized additive model with a single smooth term displays the estimated conditional mean effect. This curve does not display how the effect changes over a single feature (the marginal effect) – this would only be the case if the feature is not correlated with other features. Instead, it displays a *combined* effect, including the effect of other correlated features (Molnar *et al.*, 2020). The accumulated local effect (ALE) method by Apley and Zhu (2020) can remove the effect of other correlated features but has yet to be applied to HTE estimators.

Applications of post-hoc interpretation methods to HTE estimators seem to be straightforward given that the \hat{f} is replaced by $\hat{\tau}$, but many open questions exist: Since features are correlated, and the HTE estimators allow for a non-parametric structure that can include interactions, many of the pitfalls described in the contribution of Chapter 7 hold. For indirect estimators (introduced in Section 3.2), the question is if the interpretation methods should be applied to the estimators of the mean expected outcome ($\hat{\eta}_1$ and $\hat{\eta}_0$) or to its difference, the treatment effect function τ . For MOB, a particular challenge is the computational time: predicting on new data points is rather costly. Since most interpretation methods are based on the SIPA framework, which consists of sampling new data and predicting the outcome (Scholbeck *et al.*, 2020), time-efficient variants of MOB or approximations of the interpretation methods are required. Furthermore, some interpretation methods require ground-truth knowledge, but the actual treatment effect is not observable. Another issue is a proper, concise summary and visualization of the results of interpretation methods, which can be easily understood, for example, by medical doctors. All of these raised points offer exciting opportunities for future research.

References

- Aalen OO, Cook RJ, Røysland K (2015). “Does Cox Analysis of a Randomized Survival Study Yield a Causal Treatment Effect?” *Lifetime Data Analysis*, **21**(4), 579–593. doi:10.1007/s10985-015-9335-y.
- Angrist JD, Imbens GW, Rubin DB (1996). “Identification of Causal Effects Using Instrumental Variables.” *Journal of the American statistical Association*, **91**(434), 444–455. doi:10.2307/2291629.
- Apley DW, Zhu J (2020). “Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models.” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **82**(4), 1059–1086. doi:10.1111/rssb.12377.
- Artelt A, Hammer B (2022). “”Even if ...” - Diverse Semifactual Explanations of Reject.” In H Ishibuchi, C Kwoh, A Tan, D Srinivasan, C Miao, A Trivedi, KA Crockett (eds.), *IEEE Symposium Series on Computational Intelligence, SSCI 2022, Singapore*, pp. 854–859. IEEE. doi:10.1109/SSCI51031.2022.10022139.
- Aryal S, Keane MT (2023). “Even If Explanations: Prior Work, Desiderata & Benchmarks for Semi-Factual XAI.” In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 6526–6535. International Joint Conferences on Artificial Intelligence Organization, Macau, SAR China. doi:10.24963/ijcai.2023/732.
- Athey S, Tibshirani J, Wager S (2019). “Generalized Random Forests.” *The Annals of Statistics*, **47**(2), 1148–1178. doi:10.1214/18-aos1709.
- Belson WA (1959). “Matching and Prediction on the Principle of Biological Classification.” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **8**(2), 65–75. doi:10.2307/2985543.
- Bennett J (1982). “Even If.” *Linguistics and Philosophy*, **5**(3), 403–418. doi:10.1007/bf00351461.
- Bezanson J, Edelman A, Karpinski S, Shah VB (2017). “Julia: A Fresh Approach to Numerical Computing.” *SIAM Review*, **59**(1), 65–98. doi:10.1137/141000671.
- Breiman L (1996). “Bagging Predictors.” *Machine Learning*, **24**(2), 123–140. doi:10.1007/bf00058655.
- Breiman L (2001a). “Random Forests.” *Machine Learning*, **45**(1), 5–32. doi:10.1023/a:1010933404324.
- Breiman L (2001b). “Statistical Modeling: The Two Cultures (with Comments and a Rejoinder by the Author).” *Statistical Science*, **16**(3), 199–231. doi:10.1214/ss/1009213726.
- Breiman L, Friedman JH, Olshen RA, Stone CJ (1984). *Classification And Regression Trees*. Chapman and Hall/CRC. doi:10.1201/9781315139470.

- Buri M, Hothorn T (2020). “Model-based Random Forests for Ordinal Regression.” *International Journal of Biostatistics*, **16**(2), 20190063. doi:10.1515/ijb-2019-0063.
- Byrne RM (2002). “Mental Models and Counterfactual Thoughts About What Might Have Been.” *Trends in Cognitive Sciences*, **6**(10), 426–431. doi:10.1016/s1364-6613(02)01974-5.
- Carvalho DV, Pereira EM, Cardoso JS (2019). “Machine Learning Interpretability: A Survey on Methods and Metrics.” *Electronics*, **8**(8). doi:10.3390/electronics8080832.
- Cinelli C, Forney A, Pearl J (2022). “A Crash Course in Good and Bad Controls.” *Sociological Methods & Research*. doi:10.1177/00491241221099552.
- Cinquini M, Guidotti R (2023). “CALIME: Causality-Aware Local Interpretable Model-Agnostic Explanations.” *arXiv 2212.05256 v2*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2212.05256.
- Clarivate (2023). “Web of Science.” Last accessed: 11.09.2023, URL <https://webofscience.com>.
- Colnet B, Mayer I, Chen G, Dieng A, Li R, Varoquaux G, Vert JP, Josse J, Yang S (2023). “Causal Inference Methods for Combining Randomized Trials and Observational Studies: A Review.” To appear in *Statistical Science*.
- Cook CE, Thigpen CA (2019). “Five Good Reasons to Be Disappointed with Randomized Trials.” *Journal of Manual & Manipulative Therapy*, **27**(2), 63–65. doi:10.1080/10669817.2019.1589697.
- Crabbé J, Curth A, Bica I, van der Schaar M (2022). “Benchmarking Heterogeneous Treatment Effect Models through the Lens of Interpretability.” In S Koyejo, S Mohamed, A Agarwal, D Belgrave, K Cho, A Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 12295–12309. Curran Associates, Inc.
- Crump RK, Hotz VJ, Imbens GW, Mitnik OA (2009). “Dealing with Limited Overlap in Estimation of Average Treatment Effects.” *Biometrika*, **96**(1), 187–199. doi:10.1093/biomet/asn055.
- Cui Y, Kosorok MR, Sverdrup E, Wager S, Zhu R (2023). “Estimating Heterogeneous Treatment Effects With Right-censored Data Via Causal Survival Forests.” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **85**(2), 179–211. doi:10.1093/jrssi/bqkac001.
- Cummins L, Bridge D (2012). “KLEOR: A Knowledge Lite Approach to Explanation Oriented Retrieval.” *Computing and Informatics*, **25**(2-3), 173–193. URL <https://www.cai.sk/ojs/index.php/cai/article/view/338>.
- Curth A, van der Schaar M (2021). “Nonparametric Estimation of Heterogeneous Treatment Effects: From Theory to Learning Algorithms.” In A Banerjee, K Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1810–1818. PMLR.
- Dandl S, Bender A, Hothorn T (2022a). “Heterogeneous Treatment Effect Estimation for Observational Data using Model-based Forests.” *arXiv 2210.02836*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2210.02836. To appear in *Statistical Methods in Medical Research*.

References

- Dandl S, Casalicchio G, Bischl B, Bothmann L (2023a). “Interpretable Regional Descriptors: Hyperbox-Based Local Explanations.” In D Koutra, C Plant, M Gomez Rodriguez, E Baralis, F Bonchi (eds.), *Machine Learning and Knowledge Discovery in Databases: Research Track (ECML PKDD 2023)*, pp. 479–495. Springer Nature Switzerland, Cham. doi: 10.1007/978-3-031-43418-1_29.
- Dandl S, Haslinger C, Hothorn T, Seibold H, Sverdrup E, Wager S, Zeileis A (2023b). “What Makes Forest-Based Heterogeneous Treatment Effect Estimators Work?” *arXiv 2206.10323 v2*, arXiv.org E-Print Archive. doi: 10.48550/arXiv.2206.10323. To appear in *The Annals of Applied Statistics*.
- Dandl S, Hofheinz A, Binder M, Bischl B, Casalicchio G (2023c). “**counterfactuals**: An R Package for Counterfactual Explanation Methods.” *arXiv 2304.06569 v2*, arXiv.org E-Print Archive. doi: 10.48550/arXiv.2304.06569.
- Dandl S, Molnar C, Binder M, Bischl B (2020). “Multi-Objective Counterfactual Explanations.” In T Bäck, M Preuss, A Deutz, H Wang, C Doerr, M Emmerich, H Trautmann (eds.), *Parallel Problem Solving from Nature – PPSN XVI*, pp. 448–469. Springer International Publishing, Cham. doi: 10.1007/978-3-030-58112-1_31.
- Dandl S, Pfisterer F, Bischl B (2022b). “Multi-Objective Counterfactual Fairness.” In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO ’22*, p. 328–331. Association for Computing Machinery, New York, NY, USA. doi: 10.1145/3520304.3528779.
- Deaton A, Cartwright N (2018). “Understanding and Misunderstanding Randomized Controlled Trials.” *Social Science & Medicine*, **210**, 2–21. doi: 10.1016/j.socscimed.2017.12.005. Randomized Controlled Trials and Evidence-based Policy: A Multidisciplinary Dialogue.
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002). “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II.” *IEEE Transactions on Evolutionary Computation*, **6**(2), 182–197. doi: 10.1109/4235.996017.
- Doshi-Velez F, Kim B (2017). “Towards A Rigorous Science of Interpretable Machine Learning.” *arXiv 1702.08608 v2*, arXiv.org E-Print Archive. doi: 10.48550/arXiv.1702.08608.
- Doyle D, Cunningham P, Bridge D, Rahman Y (2004). “Explanation Oriented Retrieval.” In *Lecture Notes in Computer Science*, pp. 157–168. Springer Berlin Heidelberg. doi: 10.1007/978-3-540-28631-8_13.
- D’Amour A, Ding P, Feller A, Lei L, Sekhon J (2021). “Overlap in Observational Studies with High-dimensional Covariates.” *Journal of Econometrics*, **221**(2), 644–654. doi: 10.1016/j.jeconom.2019.10.014.
- European Parliament, Council of the European Union (2016). “Regulation (EU) 2016/679.” OJ L 119, 4.5.2016, p. 1–88. Online access: <https://data.europa.eu/eli/reg/2016/679/oj>.
- Fisher A, Rudin C, Dominici F (2019). “All Models are Wrong, but Many are Useful: Learning a Variable’s Importance by Studying an Entire Class of Prediction Models Simultaneously.” *Journal of Machine Learning Research*, **20**(177), 1–81. URL <http://jmlr.org/papers/v20/18-760.html>.

- Fokkema M, Smits N, Zeileis A, Hothorn T, Kelderman H (2018). “Detecting Treatment-Subgroup Interactions in Clustered Data with Generalized Linear Mixed-Effects Model Trees.” *Behavior Research Methods*, **50**(6), 2016–2034. doi:10.3758/s13428-017-0971-x.
- Foster JC, Taylor J, Ruberg S (2011). “Subgroup Identification from Randomized Clinical Trial Data.” *Statistics in Medicine*, **30**(24), 2867–2880. doi:10.1002/sim.4322.
- Frost N, Hinton G (2017). “Distilling a Neural Network Into a Soft Decision Tree.” *arXiv 1711.09784*, arXiv.org E-Print Archive. doi:10.48550/arXiv.1711.09784.
- Gao Z, Hastie T (2022). “Estimating Heterogeneous Treatment Effects for General Responses.” *arXiv 2103.04277 v4*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2103.04277.
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014). “Generative Adversarial Nets.” In Z Ghahramani, M Welling, C Cortes, N Lawrence, K Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.
- Goodman N (1947). “The Problem of Counterfactual Conditionals.” *The Journal of Philosophy*, **44**(5). doi:10.2307/2019988.
- Gower JC (1971). “A General Coefficient of Similarity and Some of Its Properties.” *Biometrics*, **27**(4), 857–871. doi:10.2307/2528823.
- Greenland S, Pearl J, Robins JM (1999). “Confounding and Collapsibility in Causal Inference.” *Statistical Science*, **14**(1), 29–46. doi:10.1214/ss/1009211805.
- Guidotti R (2022). “Counterfactual Explanations and How to Find Them: Literature Review and Benchmarking.” *Data Mining and Knowledge Discovery*. doi:10.1007/s10618-022-00831-6.
- Hada SS, Carreira-Perpiñán MÁ (2021). “Exploring Counterfactual Explanations for Classification and Regression Trees.” In M Kamp, I Koprinska, A Bibal et al (eds.), *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 489–504. Springer International Publishing, Cham. doi:10.1007/978-3-030-93736-2_37.
- Hariton E, Locascio JJ (2018). “Randomised Controlled Trials - the Gold Standard for Effectiveness Research.” *BJOG: An International Journal of Obstetrics & Gynaecology*, **125**(13), 1716–1716. doi:10.1111/1471-0528.15199.
- Haslinger C, Korte W, Hothorn T, Brun R, Greenberg C, Zimmermann R (2020). “The Impact of Parturient Factor XIII Activity on Postpartum Blood Loss.” *Journal of Thrombosis and Haemostasis*, **18**, 1310–1319. doi:10.1111/jth.14795.
- Hastie T, Tibshirani R, Friedman J (2009). *The Elements of Statistical Learning*. Springer New York. doi:10.1007/978-0-387-84858-7.
- Henderson NC, Louis TA, Rosner GL, Varadhan R (2018). “Individualized Treatment Effects with Censored Data via Fully Nonparametric Bayesian Accelerated Failure Time Models.” *Biostatistics*, **21**(1), 50–68. doi:10.1093/biostatistics/kxy028.
- Hernán MA, Robins JM (2020). *Causal Inference: What If*. Boca Raton: Chapman & Hall/CRC.

References

- Heskes T, Sijben E, Bucur IG, Claassen T (2020). “Causal Shapley Values: Exploiting Causal Knowledge to Explain Individual Predictions of Complex Models.” In H Larochelle, M Ranzato, R Hadsell, M Balcan, H Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 4778–4789. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/32e54441e6382a7fbacbbbf3c450059-Paper.pdf.
- Holland PW (1986). “Statistics and Causal Inference.” *Journal of the American Statistical Association*, **81**(396), 945–960. doi:10.2307/2289064.
- Hothorn T, Hornik K, Zeileis A (2006). “Unbiased Recursive Partitioning: A Conditional Inference Framework.” *Journal of Computational and Graphical Statistics*, **15**(3), 651–674. doi:10.1198/106186006x133933.
- Hothorn T, Zeileis A (2021). “Predictive Distribution Modelling Using Transformation Forests.” *Journal of Computational and Graphical Statistics*, **14**, 144–148. doi:10.1080/10618600.2021.1872581.
- Hu L, Ji J, Li F (2021). “Estimating Heterogeneous Survival Treatment Effect in Observational Data using Machine Learning.” *Statistics in Medicine*, **40**, 4691–4713. doi:10.1002/sim.9090.
- Hudgens MG, Halloran ME (2008). “Toward Causal Inference With Interference.” *Journal of the American Statistical Association*, **103**(482), 832–842. doi:10.1198/016214508000000292.
- Hume D (1748). “An Enquiry Concerning Human Understanding.”
- Humphries G, Magness DR, Huettmann F (eds.) (2018). *Machine Learning for Ecology and Sustainable Natural Resource Management*. Springer International Publishing. doi:10.1007/978-3-319-96978-7.
- Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS (2008). “Random Survival Forests.” *The Annals of Applied Statistics*, **2**(3), 841–860. doi:10.1214/08-aos169.
- Japkowicz N, Shah M (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press. doi:10.1017/CBO9780511921803.
- Karimi AH, Schölkopf B, Valera I (2021). “Algorithmic Recourse: From Counterfactual Explanations to Interventions.” In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 353–362. doi:10.1145/3442188.3445899.
- Kennedy EH (2022). “Towards Optimal Doubly Robust Estimation of Heterogeneous Causal Effects.” *arXiv 2004.14497 v3*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2004.14497.
- Knaus MC, Lechner M, Strittmatter A (2020). “Machine Learning Estimation of Heterogeneous Causal Effects: Empirical Monte Carlo Evidence.” *The Econometrics Journal*, **24**(1), 134–161. doi:10.1093/ectj/utaa014.
- Korepanova N, Seibold H, Steffen V, Hothorn T (2020). “Survival Forests under Test: Impact of the Proportional Hazards Assumption on Prognostic and Predictive Forests for ALS Survival.” *Statistical Methods in Medical Research*, **29**(5), 1403–1419. doi:10.1177/0962280219862586.
- Künzel SR, Sekhon JS, Bickel PJ, Yu B (2019). “Metalearners for Estimating Heterogeneous Treatment Effects Using Machine Learning.” *Proceedings of the National Academy of Sciences*, **116**(10), 4156–4165. doi:10.1073/pnas.1804597116.

- Kusner M, Loftus J, Russell C, Silva R (2017). “Counterfactual Fairness.” In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, pp. 4069–4079. Curran Associates Inc., Red Hook, NY, USA. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.
- Lewis DK (1973). *Counterfactuals*. Malden, Massachusetts: Blackwell.
- Loftus JR, Bynum LEJ, Hansen S (2023). “Causal Dependence Plots.” *arXiv 2303.04209 v2*, arXiv.org E-Print Archive. doi: 10.48550/arXiv.2303.04209.
- Loscul C, Schmitz T, Blanc-Petitjean P, Goffinet F, Ray CL (2019). “Risk of Cesarean after Induction of Labor in Twin Compared to Singleton Pregnancies.” *European Journal of Obstetrics & Gynecology and Reproductive Biology*, **237**, 68–73. doi: 10.1016/j.ejogrb.2019.04.005.
- Lu M, Sadiq S, Feaster DJ, Ishwaran H (2018). “Estimating Individual Treatment Effect in Observational Data Using Random Forest Methods.” *Journal of Computational and Graphical Statistics*, **27**(1), 209–219. doi: 10.1080/10618600.2017.1356325.
- MacDorman MF, Declercq E, Cabral H, Morton C (2016). “Recent Increases in the U.S. Maternal Mortality Rate: Disentangling Trends From Measurement Issues.” *Obstetrics & Gynecology*, **128**(3), 447–455. doi: 10.1097/AOG.0000000000001556.
- MacEachern SJ, Forkert ND (2021). “Machine Learning for Precision Medicine.” *Genome*, **64**(4), 416–425. doi: 10.1139/gen-2020-0131.
- McCloy R, Byrne RM (2002). “Semifactual “Even If” Thinking.” *Thinking & Reasoning*, **8**(1), 41–67. doi: 10.1080/13546780143000125.
- McNamee R (2005). “Regression Modelling and Other Methods to Control Confounding.” *Occupational and Environmental Medicine*, **62**(7), 500–506. doi: 10.1136/oem.2002.001115.
- Mitchell TM (1997). *Machine Learning*. McGraw-Hill New York.
- Molnar C, Casalicchio G, Bischl B (2020). “Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges.” In I Koprinska, M Kamp, A Appice et al (eds.), *ECML PKDD 2020 Workshops*, volume 1323 of *Communications in Computer and Information Science*, pp. 417–431. Springer International Publishing, Cham. doi: 10.1007/978-3-030-65965-3_28.
- Molnar C, König G, Herbringer J, Freiesleben T, Dandl S, Scholbeck CA, Casalicchio G, Grosse-Wentrup M, Bischl B (2022). “General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models.” In A Holzinger, R Goebel, R Fong, T Moon, KR Müller, W Samek (eds.), *xxAI - Beyond Explainable AI*, volume 13200 of *Lecture Notes in Artificial Intelligence*, pp. 39–68. Springer, Cham. doi: 10.1007/978-3-031-04083-2_4.
- Nelder JA, Mead R (1965). “A Simplex Method for Function Minimization.” *The Computer Journal*, **7**(4), 308–313. doi: 10.1093/comjnl/7.4.308.
- Nie X, Wager S (2020). “Quasi-oracle Estimation of Heterogeneous Treatment Effects.” *Biometrika*, **108**(2), 299–319. doi: 10.1093/biomet/asaa076.
- Nugent C, Doyle D, Cunningham P (2009). “Gaining Insight Through Case-based Explanation.” *Journal of Intelligent Information Systems*, **32**(3), 267–295. doi: 10.1007/s10844-008-0069-0.

References

- OpenAI (2023). “ChatGPT (August 25 Version) [Large Language Model].” URL <https://chat.openai.com>.
- Pearl J (1995). “Causal Diagrams for Empirical Research.” *Biometrika*, **82**(4), 669–688. doi: 10.1093/biomet/82.4.669.
- Pearl J (2022). “Interview with Judea Pearl.” *Observational Studies*, **8**(2), 23–36. doi:10.1353/obs.2022.0007.
- Pearl J, Glymour M, Jewel NP (2016). “Causal Inference in Statistics: A Primer.” *John Wiley & Sons*, **88**(1), 256–258.
- Powers S, Qian J, Jung K, Schuler A, Shah NH, Hastie T, Tibshirani R (2018). “Some Methods for Heterogeneous Treatment Effect Estimation in High Dimensions.” *Statistics in Medicine*, **37**(11), 1767–1787. doi: 10.1002/sim.7623.
- R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Ribeiro MT, Singh S, Guestrin C (2016). “Why Should I Trust You?: Explaining the Predictions of Any Classifier.” In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, pp. 1135–1144. Association for Computing Machinery, New York, NY, USA. doi:10.1145/2939672.2939778.
- Robins JM, Rotnitzky A (1995). “Semiparametric Efficiency in Multivariate Regression Models with Missing Data.” *Journal of the American Statistical Association*, **90**(429), 122–129. doi: 10.1080/01621459.1995.10476494.
- Robinson PM (1988). “Root-N-Consistent Semiparametric Regression.” *Econometrica*, **56**(4), 931–954. doi:10.2307/1912705.
- Rosenbaum PR, Rubin DB (1983). “The Central Role of the Propensity Score in Observational Studies for Causal Effects.” *Biometrika*, **70**(1), 41–55. doi:10.1093/biomet/70.1.41.
- Rosenblatt F (1957). “The Perceptron: A Perceiving and Recognizing Automaton.” *Technical Report 85-460-1*, Cornell Aeronautical Laboratory, Ithaca, New York.
- Rubin DB (1974). “Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies.” *Journal of Educational Psychology*, **66**(5), 688–701.
- Rubin DB (1980). “Randomization Analysis of Experimental Data: The Fisher Randomization Test Comment.” *Journal of the American Statistical Association*, **75**(371), 591–593. doi:10.2307/2287653.
- Rubin DB (2005). “Causal Inference Using Potential Outcomes: Design, Modeling, Decisions.” *Journal of the American Statistical Association*, **100**(469), 322–331. URL <https://www.jstor.org/stable/27590541>.
- Schapire RE (1990). “The Strength of Weak Learnability.” *Machine Learning*, **5**(2), 197–227. doi:10.1007/bf00116037.

- Scholbeck CA, Molnar C, Heumann C, Bischl B, Casalicchio G (2020). “Sampling, Intervention, Prediction, Aggregation: A Generalized Framework for Model-Agnostic Interpretations.” In P Cellier, K Driessens (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 205–216. Springer International Publishing, Cham. doi:10.1007/978-3-030-43823-4_18.
- Schölkopf B, Locatello F, Bauer S, Ke NR, Kalchbrenner N, Goyal A, Bengio Y (2021). “Toward Causal Representation Learning.” *Proceedings of the IEEE*, **109**(5), 612–634. doi:10.1109/JPROC.2021.3058954.
- Seibold H, Zeileis A, Hothorn T (2016). “Model-Based Recursive Partitioning for Subgroup Analyses.” *International Journal of Biostatistics*, **12**(1), 45–63. doi:10.1515/ijb-2015-0032.
- Seibold H, Zeileis A, Hothorn T (2018). “Individual Treatment Effect Prediction for Amyotrophic Lateral Sclerosis Patients.” *Statistical Methods in Medical Research*, **27**(10), 3104–3125. doi:10.1177/0962280217693034.
- Seibold H, Zeileis A, Hothorn T (2021). **model4you**: *Stratified and Personalised Models Based on Model-Based Trees and Forests*. R package version 0.9-7, URL <https://CRAN.R-project.org/package=model4you>.
- Shalit U, Johansson FD, Sontag D (2017). “Estimating Individual Treatment Effect: Generalization Bounds and Algorithms.” In D Precup, YW Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3076–3085. PMLR. URL <https://proceedings.mlr.press/v70/shalit17a/shalit17a.pdf>.
- Splawa-Neyman J, Dabrowska DM, Speed TP (1923). “Sur les Applications de la Théorie des Probabilités aux Expériences Agricoles: Essai des Principes.” *Roczniki Nauk Rolniczych*, **10**.
- Spooner T, Dervovic D, Long J, Shepard J, Chen J, Magazzeni D (2021). “Counterfactual Explanations for Arbitrary Regression Models.” *International Conference on Machine Learning (ICML) Workshop on Algorithmic Recourse*.
- Sutton RS, Barto AG (2018). *Reinforcement Learning: An Introduction*. Second edition. The MIT Press. URL <https://mitpress.mit.edu/9780262039246/reinforcement-learning/>.
- Tibshirani J, Athey S, Sverdrup E, Wager S (2023). **grf**: *Generalized Random Forests*. R package version 2.3.0, URL <https://CRAN.R-project.org/package=grf>.
- Van Rossum G, Drake Jr FL (1995). “Python Tutorial.” *Technical Report CS-R9526*, Centrum voor Wiskunde en Informatica.
- VanderWeele TJ, Hernán MA (2013). “Causal Inference Under Multiple Versions of Treatment.” *Journal of Causal Inference*, **1**(1), 1–20. doi:10.1515/jci-2012-0002.
- VanderWeele TJ, Shpitser I (2013). “On the Definition of a Confounder.” *The Annals of Statistics*, **41**(1). doi:10.1214/12-AOS1058.
- Vapnik V (1982). *Estimation of Dependences Based on Empirical Data, Addendum 1*. Springer New York. doi:10.1007/0-387-34239-7.
- Verma S, Boonsanong V, Hoang M, Hines KE, Dickerson JP, Shah C (2022). “Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review.” *arXiv 2010.10596 v3*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2010.10596.

References

- Vowels MJ, Camgoz NC, Bowden R (2022). “D’ya Like DAGs? A Survey on Structure Learning and Causal Discovery.” *ACM Computing Surveys*, **55**(4). doi:10.1145/3527154.
- Wachter S, Mittelstadt B, Russell C (2018). “Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR.” *Harvard Journal of Law & Technology*, **31**(2), 841–887. doi:10.2139/ssrn.3063289.
- Warin T, Stojkov A (2021). “Machine Learning in Finance: A Metadata-Based Systematic Review of the Literature.” *Journal of Risk and Financial Management*, **14**(7). doi:10.3390/jrfm14070302.
- Wexler J, Pushkarna M, Bolukbasi T, Wattenberg M, Viégas F, Wilson J (2019). “The What-If Tool: Interactive Probing of Machine Learning Models.” *IEEE transactions on visualization and computer graphics*, **26**(1), 56–65. doi:10.1109/TVCG.2019.2934619.
- Wright MN, Ziegler A (2017). “ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R.” *Journal of Statistical Software*, **77**(1), 1–17. doi:10.18637/jss.v077.i01.
- Wright S (1921). “Correlation and Causation.” *Journal of Agricultural Research*, **20**, 557–585.
- Zeileis A, Hothorn T, Hornik K (2008). “Model-Based Recursive Partitioning.” *Journal of Computational and Graphical Statistics*, **17**(2), 492–514. doi:10.1198/106186008x319331.

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, §8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 18.09.2023

Susanne Dandl

