Tessa Steensgaard

Transparent and Fair Statistical Learning in Non-Life Insurance

PHD THESIS

Department of Mathematical Sciences
University of Copenhagen

August 2026

This thesis has been submitted to the PhD school of
the faculty of science, University of Copenhagen

PhD thesis by:

Tessa Steensgaard
tes@math.ku.dk
Department of Mathematical Sciences
University of Copenhagen
Universitetsparken 5
DK-2100 Copenhagen Ø

Thesis title:
Transparent and Fair Statistical Learning in Non-Life Insurance

Assessment committee:

Associate Professor Martin Bladt (Chairperson)
University of Copenhagen

Professor Montserrat Guillén
University of Barcelona

Associate Professor Mathias Lindholm
Stockholm University

Principal supervisor:

Associate Professor Munir Hiabu
University of Copenhagen

Co-supervisor:

Professor Mogens Steffensen
University of Copenhagen

Co-supervisor:

Associate Professor Christian Furrer
University of Copenhagen

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen on August 13 2026.

Christian Furrer supervised until January 2025.

Chapter 1: © Tessa Steensgaard.
Chapter 2: © Munir Hiabu, Jinyang Liu, Niklas Pfister, Tessa Steensgaard, & Marvin Wright.
Chapter 3: © Munir Hiabu, Niklas Pfister, & Tessa Steensgaard.
Chapter 4: © Munir Hiabu, Fei Huang, Patrick J. Laub, & Tessa Steensgaard.

ISBN: 978-87-7629-249-2

Preface

This thesis has been prepared in partial fulfillment of the requirements for the PhD degree at the Department of Mathematical Sciences, Faculty of Science, University of Copenhagen. The work was carried out between August 2023 and August 2026 in association with the project frame InterAct, an extensive agreement of collaboration within research and education between the Department of Mathematical Sciences at the University of Copenhagen and, as of August 2026, the seven Danish insurance and pension companies AP Pension, Danica Pension, Industriens Pension, PFA Pension, PensionDanmark, Tryg, and Velliv (listed in alphabetical order). The topics covered by this thesis were determined jointly by the industrial partners, the university, and myself.

During my studies, I was supervised by Associate Professor Munir Hiabu and co-supervised by Professor Mogens Steffensen and Associate Professor Christian Furrer. Christian Furrer supervised until January 2025.

The thesis consists of three manuscripts, each presenting stand-alone scientific contributions. As such notation and terminology might differ between chapters. An introductory chapter is included to provide context of the interplay between the manuscripts and to the current international state-of-the-art.

Declaration on the use of generative AI: Large language models, specifically GPT-5.6 Sol and earlier models distributed by OpenAI, assisted in the preparation of this thesis through information searches, code generation, and proofreading. All AI-generated output was critically reviewed and edited, and I take full responsibility for the final content.

Acknowledgments

I owe the existence of this PhD thesis above all to the wonderful people who helped me along the way.

First and foremost, I would like to thank Munir for your interest in starting this research project with me, for three years of inspiring supervision, and for always keeping your door open.

I am grateful to Christian for guiding me through all the parts of academia that exist beyond research, and to Mogens for your help and encouragement even before the project began.

I extend my gratitude to my co-authors, Munir, Jin, Niklas, Marvin, Patrick, and Fei, for generously giving your time and for our engaging discussions and rewarding collaborations.

Thank you, Patrick and Fei, for hosting me and taking care of me during my time on the other side of the world. I am also grateful to the staff at UNSW for their warm welcome and to the PhD students for including me in university life. A special thank you to Gayani for always finding new and exciting places to take me.

I am grateful to the InterAct partners for the discussions and input that inspired and shaped this thesis. The passion and insight you brought to every meeting were a constant source of inspiration. I would also like to thank my fellow InterAct PhD students: Philipp, with whom I paved the way, and later Rasmus and Tobias. I greatly value the support we have been able to give one another.

Thank you to my former colleagues at Tryg, in particular Vinnie and Anders, for encouraging me to apply for the PhD. I would not be here without your support.

I would also like to thank my colleagues at the department. To the seniors, thank you for the ever-open invitations to join your lunches full of life; to the administration, for fixing any problem within minutes. And to my fellow PhD students Phillip, Margherita, Ulises, Cecilie, Nena, Jiali, Philipp, Jin, Christoffer, Malthe, Karl, Theodor, Marcus, Jon, Gabriel, Rasmus, Francesco, Marco, Rasmus, Alessandro, and Tobias, thank you for for making the last three years an experience rather than a job.

To my friends and family, thank you for supporting me along the way. To Christina and Jim, for being there for me in so many ways. A Serena e Eugenio, for welcoming me into your family. And to Mor og Far, for your unconditional love and for accepting every choice I make, with the only condition that it brings me happiness.

And finally, thank you, Gabriele. For being there from the very first day, and for the support, devotion, and love I know you will show me until the last. I wrote this thesis for you.

Tessa Steensgaard
August 2026

Abstract

This thesis consists of three independent manuscripts and an introductory chapter that places their contributions in the context of non-life insurance. The manuscripts address model interpretability, fairness, and customer portfolio evolution. All contributions are illustrated using relevant simulated and real-world data sets.

First, we consider partial dependence (PD) functions, which form the basis of widely used model interpretation methods such as PD plots and SHAP values. We discuss how these methods can obscure interaction effects and build on recent work that uses PD functions to construct a functional decomposition separating main and higher-order interaction effects. We propose an efficient and consistent algorithm, `FastPD`, for estimating arbitrary PD functions for tree-based models. At a complexity comparable to literature benchmarks estimating SHAP values, `FastPD` estimates PD functions from which PD plots, SHAP values, and the functional decomposition can be obtained with little additional computational effort.

Second, we consider the problem of predicting a response Y using features X in the presence of an auxiliary attribute A, when the distribution of (A, X) should not determine the prediction target. We call a prediction functional AX-invariant if it depends on the underlying distribution P only through $P^{Y|A,X}$ and is therefore unaffected by the joint distribution of (A, X). Under suitable causal assumptions, we show that AX-invariance is a necessary condition for functional-level path-specific and interventional fairness and, under stronger assumptions, is equivalent to these fairness notions. We characterise AX-invariant estimands and develop an asymptotic test for falsifying AX-invariance from observed data.

Finally, we consider the evolution of an insurance customer's product portfolio, where products are purchased and cancelled over time until the customer fully churns. Motivated by the assumption that these events cluster in time, we propose a fully parametric bivariate Hawkes process in which purchases and cancellations are modelled as mutually exciting counting processes. Customer and portfolio characteristics enter through covariates, while the specific products added or dropped are represented by marks. We further implement a simulation algorithm based on thinning that generates event histories from the fitted model parameters, allowing entire future portfolio trajectories to be simulated and evaluated.

Sammenfatning

Denne afhandling består af tre selvstændige manuskripter samt et indledende kapitel, der sætter deres bidrag ind i en skadesforsikringsmæssig kontekst. Manuskripterne omhandler modelfortolkning, fairness og udviklingen i kunders produktporteføljer. Alle bidrag illustreres ved hjælp af relevante simulerede og virkelige datasæt.

Først betragter vi partial dependence-funktioner (PD-funktioner), som danner grundlag for udbredte metoder til modelfortolkning såsom PD-plots og SHAP-værdier. Vi diskuterer, hvordan disse metoder kan skjule interaktionseffekter, og bygger videre på nyere forsking, hvor PD-funktioner anvendes til at konstruere en funktionel dekomponering, der adskiller hovedeffekter fra interaktionseffekter af højere orden. Vi foreslår en effektiv og konsistent algoritme, `FastPD`, til estimering af vilkårlige PD-funktioner for træbaserede modeller. Med en beregningsmæssig kompleksitet, der er sammenlignelig med eksisterende metoder i litteraturen til estimering af SHAP-værdier, estimerer `FastPD` PD-funktioner, hvorfra PD-plots, SHAP-værdier og den funktionelle dekomponering kan udledes med kun en begrænset yderligere beregningsindsats.

Dernæst beskæftiger vi os med problemstillingen om at prædiktere en respons Y ud fra kovariater X i tilstedeværelsen af en supplerende attribut A, hvor fordelingen af (A, X) ikke må have indflydelse på prædiktionsmålet. Vi kalder en prædiktionsfunktional AX-invariant, hvis den kun afhænger af den underliggende fordeling P gennem $P^{Y|A,X}$ og dermed er upåvirket af den simultane fordeling af (A, X). Under passende kausale antagelser viser vi, at AX-invarians er en nødvendig betingelse for path-specific og interventionel fairness på funktionalniveau og, under stærkere antagelser, er ækvivalent med disse fairnessbegreber. Vi karakteriserer AX-invariante estimander og udvikler en asymptotisk test til at falsificere AX-invarians ud fra observerede data.

Endelig betragter vi udviklingen i en forsikringskundes produktportefølje, hvor produkter købes og opsiges over tid, indtil kunden har opsagt alle sine produkter. Motiveret af antagelsen om, at disse hændelser har tendens til at optræde tæt på hinanden i tid, foreslår vi en fuldt parametrisk bivariat Hawkes-proces, hvor køb og opsigelser modelleres som gensidigt selvforstærkende tælleprocesser. Kunde- og porteføljekarakteristika indgår som kovariater, mens de specifikke produkter, der tilføjes eller fjernes, repræsenteres ved hjælp af mærker. Vi implementerer desuden

en simulationsalgoritme baseret på udtynding, som genererer hændelsesforløb ud fra de estimerede modelparametre. Dette gør det muligt at simulere og evaluere hele fremtidige forløb for kundens produktportefølje.

List of Papers

This thesis consists of four separate chapters. Chapter 1 provides an introduction to the thesis, collecting the contributions of the following chapters. The final three chapters are self-contained manuscripts whereof, at the time of submission, Chapter 2 presents a peer-reviewed, published work and Chapters 3 and 4 are yet-to-be-submitted working manuscripts.

Chapter 2: Liu, J., T. Steensgaard, M. N. Wright, N. Pfister, and M. Hiabu (2025). Fast estimation of partial dependence functions using trees. In *Proceedings of the 42nd International Conference on Machine Learning*, Volume 267 of *Proceedings of Machine Learning Research*, pp. 39496–39534. PMLR.

Chapter 3: Hiabu, M., N. Pfister, and T. Steensgaard. AX-Invariance. *(Working manuscript).*

Chapter 4: Hiabu, M., F. Huang, P. J. Laub, and T. Steensgaard. Modelling Non-Life Insurance Customer Portfolio Changes via Hawkes Processes. *(Working manuscript).*

Contents

Chapter 1

Introduction

This thesis consists of two main research streams. The first considers explainability and fairness for models, often fitted using machine learning, while the second focuses on modelling churn and portfolio evolution in non-life insurance under an assumption of self-excitation. The first stream is treated in Chapters 2 and 3, and the second in Chapter 4. While the methods presented have broader applications, this chapter introduces the background for the two research streams separately within the context of non-life insurance and describes the contributions of the manuscripts relative to the current state-of-the-art.

1.1 Insurance pricing with machine learning

Within non-life insurance, one often considers the regression problem of estimating the expected value of a response variable $Y \in \mathbb{R}$ given a multidimensional set of features $X \in \mathcal{X}$, with true underlying distribution $P_0 \in \mathcal{P}$:

$$m(x) = \mathbb{E}_{P_0}[Y \mid X = x]. \tag{1.1}$$

Since P_0 is unknown, we estimate m from a sample of $n \in \mathbb{N}$ observations and denote the resulting fitted model by $\hat{m}$. For example, Y could represent the claim cost associated with an automobile insurance product. In this case, $\hat{m}(x)$ corresponds to the price, also referred to as the risk premium, paid by a policyholder in exchange for coverage, given their specific characteristics X. By determining the premium from the policyholder's individual characteristics, the policyholder is charged according to their own estimated risk, in which case the premium is referred to as 'actuarially fair'.

Classically, the risk premium is modelled using generalised linear models (GLMs), for example by using the Tweedie distribution or by decomposing the claim cost into frequency and severity components and modelling these using Poisson and Gamma distributions, respectively. With the development of machine learning methods, such

as random forests, gradient boosting machines, and neural networks, more flexible models have become available that can outperform classical parametric approaches in terms of predictive performance. This increased flexibility, however, introduces a new challenge. Whereas GLMs and other parametric models can often be understood directly from their estimated parameters, machine learning models may be highly intricate, making it difficult, if not impossible, for humans to understand how the observed input features x are mapped to the model output $\hat{m}(x)$.

Within insurance, model interpretability is important for several reasons. The conclusions drawn from a model should be understandable to the actuary developing and validating it, but also to the business offering the product and to the customer receiving the resulting premium. In the following chapters, we focus on two branches of research addressing different problems arising from the black-box nature of machine learning models. First, we consider interpretable machine learning, where the aim is to apply methods that make the behaviour of a fitted model more transparent. Second, we consider fairness, where a protected feature is present whose effects, for ethical or legal reasons, should not influence the model $\hat{m}$, either directly or indirectly.

1.1.1 Some methods in explainable AI

In this section, we present some commonly used methods for model interpretability. Let $x \in \mathcal{X} \subseteq \mathbb{R}^d$ denote a d-dimensional vector of observed features. We consider a subset of features x_S, indexed by $S \subseteq [d]$, and let $\bar{S} = [d] \setminus S$ denote the indices of the remaining features. Our objective is to understand how the features in x_S affect the fitted model $\hat{m}$.

In insurance applications, we are often specifically interested in the effect of x_S on the model $\hat{m}$. If, instead, the objective is to understand the effect on the underlying real-world regression function m, interpretation may require extrapolation beyond the support of the observed data. We briefly discuss this distinction in Chapter 2.

Define the partial dependence (PD) function as

$$\mathrm{PD}_S(x_S) = \mathbb{E}_{P_0}\left[\hat{m}(x_S, X_{\bar{S}})\right].$$

Here, we assume that the pair $(x_S, X_{\bar{S}})$ is ordered consistently with the input indexing of $\hat{m}$. The PD function evaluates $\hat{m}$ while keeping the features indexed by S fixed and averaging over the remaining features. When a single feature x_k, $k \in [d]$, is considered, the one-dimensional function $\mathrm{PD}_k(x_k)$ is commonly visualised as a PD plot. It illustrates the dependence of $\hat{m}$ on the kth feature, x_k, while averaging over all remaining features.

Inspired by cooperative game theory, another popular interpretability tool is the

Shapley value, defined as

$$\phi_k(x) = \frac{1}{d} \sum_{S \subseteq [d] \setminus \{k\}} \binom{d-1}{|S|}^{-1} \left(v_{S \cup \{k\}}(x_{S \cup \{k\}}) - v_S(x_S) \right),$$

where v is referred to as the value function (Štrumbelj and Kononenko, 2010). We interpret $\phi_k(x)$ as the contribution of the kth feature to the prediction $\hat{m}(x)$. We focus on the case where the value function is given by the PD function, that is,

$$v_S(x_S) = \mathrm{PD}_S(x_S),$$

in which case we refer to $\phi_k(x)$ as the SHAP value.

When the model includes interaction effects, as is predominantly the case for machine learning models, neither PD functions nor SHAP values provide a complete description of the behaviour of $\hat{m}$; see Example 2.2.2. Instead, we may consider a functional decomposition of the model,

$$\begin{aligned} \hat{m}(x) &= \hat{m}_0 + \sum_{k=1}^{d} \hat{m}_k(x_k) + \sum_{k<l} \hat{m}_{kl}(x_{k,l}) + \cdots + \hat{m}_{1,\ldots,d}(x) \\ &= \sum_{S \subseteq [d]} \hat{m}_S(x_S), \end{aligned}$$

for all $x \in \mathcal{X}$. Such a decomposition allows us to distinguish between main effects and higher-order interaction effects within the model. However, the decomposition above is not unique. We therefore introduce the identification proposed by Hiabu et al. (2023), which is based on the PD functions:

$$\sum_{U \subseteq S} \hat{m}_U(x_U) = \mathrm{PD}_S(x_S).$$

This identification yields the unique solution

$$\hat{m}_S(x_S) = \sum_{U \subseteq S} (-1)^{|*|S \setminus U} \mathrm{PD}_U(x_U).$$

1.1.2 Contributions of Chapter 2

As discussed in the previous section, PD functions can be used to construct several interpretability tools commonly employed in explainable AI. Efficient computation of PD functions is therefore important, particularly for increasingly large and complex models. In Chapter 2, we focus on tree-based machine learning methods.

One popular method for estimating SHAP values is TreeSHAP (Lundberg et al., 2020), a version of which is implemented in the widely used XGBoost and LightGBM packages for fitting gradient-boosted tree models (Chen and Guestrin, 2016; Ke et al., 2017). We show that this version, known as path-dependent TreeSHAP, can

yield inconsistent estimates of SHAP values; see Proposition 2.3.1. In contrast, interventional TreeSHAP provides consistent estimates but has a substantially higher computational cost and is typically applied using only a small background sample. More recently, Zern et al. (2023) proposed an efficient algorithm for consistently estimating SHAP values.

We introduce an algorithm, which we call FastPD, that similarly exploits the tree structure to efficiently and consistently estimate PD functions. FastPD computes PD functions with a runtime comparable to that of Zern et al. (2023), while additionally allowing PD plots and a functional decomposition to be obtained with little additional computational effort.

Since the publication of the manuscript, the computational complexity of the FastPD algorithm has been further improved by Wettenstein et al. (2026).

1.1.3 Fairness in insurance

We now assume that, in addition to the features X, there exists a protected feature $A \in \mathcal{A}$. In an insurance context, A could, for example, represent the gender of the policyholder. The conditional mean in (1.1) determines the expected response Y given only the features X. However, excluding A from the model does not necessarily imply that the resulting pricing rule is fair. If A and X are dependent, then X may contain information about A, so that the conditional mean may depend indirectly on the protected feature. This can be seen from the representation

$$\mathbb{E}_{P_0}[Y \mid X = x] = \int_{\mathcal{A}} \mathbb{E}_{P_0}[Y \mid A = a, X = x] \mathrm{d}P_0^{A|X=x}(a),$$

which shows that the conditional mean depends on the conditional distribution $P_0^{A|X=x}$. Consequently, even when A is not included explicitly in the model, information about A may still enter the pricing rule through its dependence with X.

For gender specifically as a protected feature in insurance pricing, the European Union, through EIOPA, states that "The use of risk factors which might be correlated with gender [...] remains possible, as long as they are true risk factors in their own right." Moreover, "In contrast to direct discrimination, indirect discrimination can be justified if the aim is legitimate and the means of achieving it are appropriate and necessary" (European Commission, 2012; EIOPA Consultative Expert Group on Digital Ethics in Insurance, 2021). Thus, the use of a feature correlated with gender may be permissible when the feature constitutes a genuine risk factor in its own right and any resulting indirect discrimination is justified as necessary and appropriate for achieving a legitimate objective.

Pricing according to the conditional mean in (1.1) using X but not A is commonly referred to as fairness through unawareness. Many alternative notions of fairness have been proposed in the literature. A causal framework is particularly useful in

this context, as it provides a high-level and often visually interpretable representation of the relationships between protected features, other covariates, and the outcome. Within such frameworks, several fairness criteria have been proposed, including counterfactual, interventional, and path-specific notions of fairness (Kilbertus et al., 2017; Kusner et al., 2017; Nabi and Shpitser, 2018; Wu et al., 2019).

Closely related to the ideas developed in Chapter 3, Lindholm et al. (2024) introduce a criterion for proxy discrimination based on invariance of the pricing functional. In particular, a pricing functional avoids proxy discrimination if, for any $P \in \mathcal{P}$, it remains unchanged under changes to the joint distribution that leave the conditional distribution $P^{Y|A,X}$ and the marginal distributions P^A and P^X unchanged.

1.1.4 Contributions of Chapter 3

Let $\mathcal{G}$ denote the set of measurable functions $g : \mathcal{X} \to \mathbb{R}$. We are particularly interested in functions in $\mathcal{G}$ because they take as input values from the feature space $\mathcal{X}$, where X takes values, and map them to a real-valued response intended to resemble Y. We therefore consider functionals $\Phi : \mathcal{P} \to \mathcal{G}$, which map a distribution P to a prediction function.

For a given $P \in \mathcal{P}$, define the set of distributions that share its conditional distribution of Y given (A, X) by

$$\mathcal{Q}(P) = \left\{ Q \in \mathcal{P} \mid Q^{Y|A,X} = P^{Y|A,X} \right\}.$$

We say that the functional Φ is AX-invariant if, for every $P \in \mathcal{P}$ and every $Q', Q'' \in \mathcal{Q}(P)$,

$$\Phi_{Q'}(x) = \Phi_{Q''}(x) \quad \text{for } \mu^X\text{-almost every } x \in \mathcal{X}. \tag{1.2}$$

Thus, an AX-invariant functional depends on P only through the conditional distribution $P^{Y|A,X}$ and is unaffected by changes to the joint distribution of (A, X). Here, μ^A and μ^X are σ-finite dominating measures such that $P^{A,X} \sim \mu^A \otimes \mu^X$.

Although AX-invariance is not itself defined as a fairness criterion, it acquires a fairness interpretation when A is a protected attribute. Under the causal comparison classes considered in Chapter 3, we show that functional level path-specific and interventional fairness imply AX-invariance. Thus, failure of AX-invariance rules out these causal fairness requirements. Under stronger causal identification assumptions, AX-invariance is equivalent to the corresponding fairness notions.

Chapter 3 provides a detailed study of the construction of AX-invariant functionals. Focusing on the squared loss, we present a selected subset of these constructions under the regularity conditions imposed in Chapter 3. Let ρ_x be a probability kernel from $\mathcal{X}$ to $\mathcal{A}$ satisfying

$$\rho_x \ll \mu^A \qquad \text{and} \qquad \operatorname{supp}(\rho_x) = \mathcal{A},$$

and define the conditional mean and variance by

$$m_P(a,x) = \mathbb{E}_P[Y \mid A=a, X=x] \quad \text{and} \quad v_P(a,x) = \mathrm{Var}_P(Y \mid A=a, X=x).$$

One AX-invariant functional is obtained by averaging the conditional mean over $\mathcal{A}$ according to ρ_x:

$$\Phi_P^{0,\rho}(x) = \int_{\mathcal{A}} m_P(a,x)\, \rho_x(\mathrm{d}a).$$

Another is the worst-case target

$$\begin{aligned}\Phi_P^{\infty}(x) &= \arg\min_{w\in\mathbb{R}} \operatorname{ess\,sup}_{\rho_x} E_P[(Y-w)^2 \mid A=a, X=x] \\ &= \arg\min_{w\in\mathbb{R}} \max_{a\in\mathcal{A}} \left[v_P(a,x) + \{m_P(a,x)-w\}^2 \right],\end{aligned}$$

for which explicit simplifications are available in several special cases; see Corollary 3.4.3.

Finally, we consider how AX-invariance can be assessed for estimators fitted to observed data. Since it is infeasible to verify (1.2) for every pair $Q', Q'' \in \mathcal{Q}(P_0)$, we develop a testing procedure that can falsify AX-invariance by considering a particular choice of distributions.

Suppose that P admits a density p. We may factorise it as

$$p(y,a,x) = p^{Y|A,X}(y \mid a,x) p^{A|X}(a \mid x) p^X(x),$$

We define the dependence-removing shift

$$\tau(P)(\mathrm{d}y, \mathrm{d}a, \mathrm{d}x) = P^{Y|A=a,X=x}(\mathrm{d}y)\, u^A(\mathrm{d}a)\, P^X(\mathrm{d}x),$$

where the reference distribution $u^A \sim \mu^A$ is uniform for finite categorical A. The transformation τ therefore removes the dependence between A and X by replacing $P^{A|X}$ with u^A, while leaving $P^{Y|A,X}$ unchanged. Consequently, $\tau(P) \in \mathcal{Q}(P)$, and AX-invariance implies

$$\Phi_{P_0} - \Phi_{\tau(P_0)} = 0.$$

A violation of this equality can therefore be used to reject AX-invariance.

Given observations indexed by $i = 1, \ldots, n$, we replace the unknown distribution P_0 by the empirical distribution P_n. It is, however, not immediate how to generate observations under the shift τ. We use the resampling procedure proposed by Thams et al. (2023), which permits sampling m observations from $\tau(P_0)$ when m is of order $o(\sqrt{n})$. For our case, we improve on this by exploiting the fact that testing AX-invariance does not require the shifted distribution to be exactly $\tau(P_0)$. We show that taking m beyond the $\sqrt{n}$ guarantee still yields a sample from a distribution $Q_{n,m} \in \mathcal{Q}(P_0)$.

Based on this, we construct a covariance test of the necessary condition

$$\mathrm{cov}_{P_0}(\Phi_{P_0}(X) - \Phi_{Q_{n,m}}(X), A) = 0,$$

and establish its asymptotic normality under suitable conditions on the independent evaluation sample and the fitting error.

1.2 Customer churn and portfolio evolution

Once the risk premium has been determined, the insurance company can, after accounting for operational costs, margin, etc., offer a final price to the customer. We now take a step back from pricing individual products and consider the overall behaviour of a policyholder throughout their customer lifetime. For example, a customer enters the company upon purchasing their first policy. From this point onwards, the customer may purchase additional covers or products or, conversely, cancel existing ones. This process continues until the customer no longer holds any coverage with the company, at which point the customer is said to have fully churned.

Understanding the full customer lifetime is useful for several purposes. Identifying early warning signals of cancellation allows the company to take action, for example by contacting the customer or offering discounts. Conversely, knowing when a customer is likely to increase their coverage can facilitate targeted cross- or up-selling. More generally, insights into customer behaviour make it possible to time outgoing communication to periods when the customer is most receptive.

Customer churn has been studied extensively across several areas. Commonly, the problem is formulated as a classification task, with the objective of identifying customers who will churn within some fixed time horizon. Such models can work well for estimating churn probabilities from relatively simple data. However, some information may be lost by disregarding the richer event history available to the insurance company.

One approach that expands on this idea is proposed by Dong et al. (2022), who construct a discrete-time multi-state model describing the current state of the customer as 'fully covered', 'partially covered', or 'churned'. Among other features, they incorporate the customer's history by imposing a second-order Markov assumption, thereby allowing the current transition probabilities to depend on the states occupied at previous time points.

In another research stream, Brockett et al. (2008), Guillén et al. (2009), and Guillén et al. (2012) study churn within a survival-analysis framework. They model the time from the cancellation of one product to the cancellation of another using an extended Cox model. More specifically, given a set of time-dependent covariates X, they model the intensity of the next cancellation as

$$\lambda(u) = Y(u)\lambda_0(u)\exp\{X(D+u)^\top \beta(u)\}, \qquad u > 0.$$

Here, Y is an at-risk indicator, D denotes the time of the first cancellation, and β represents the time-dependent covariate effects.

A common feature of these two approaches is that they exploit information contained in the customer's previous behaviour. Both recognise that the probability of churn may depend on earlier observed cancellations or purchases. Related work studies insurance recommendations by including life events as predictors through so-called Hawkes processes (Lesage et al., 2024). We use a related modelling framework to study the broader evolution of an insurance customer's portfolio.

1.2.1 Self-exciting counting processes

Define the counting process $(N(t))_{t\geq 0}$ with arrival times $0 < T_1 < T_2 < \cdots$ by

$$N(t) = \#\{\ell : T_\ell \leq t\}.$$

We assume that N admits an intensity process, λ, defined as

$$\lambda(t) = \lim_{\Delta t \to 0^+} \frac{\mathbb{E}[N(t+\Delta t) - N(t)|\mathcal{F}(t)]}{\Delta t},$$

where $\mathcal{F}(t)$ denotes the filtration generated by the counting process up to time t and can be interpreted as the complete history of N prior to time t.

We call $(N(t))_{t\geq 0}$ a Hawkes process if its intensity takes the form

$$\lambda(t) = \mu + \eta \sum_{\ell: T_\ell < t} \nu(t - T_\ell). \tag{1.3}$$

Here, μ is the background rate, while the second term represents the self-exciting component. Each arrival causes an immediate increase in the intensity, after which its contribution decays according to the excitation function $\nu : (0,\infty) \to (0,\infty)$, which is assumed to be a probability density. The parameter $\eta > 0$ is referred to as the branching ratio and controls the overall magnitude of the self-excitation.

The specification in (1.3) is highly flexible. For example, the background rate may be generalised using more complex parametric or non-parametric models, while different choices of the excitation function allow for different patterns of decay. Given observations over the interval $[0, T]$, the log-likelihood takes the familiar point-process form

$$\ell(\theta) = \sum_{\ell: T_\ell \leq T} \log\big\{\lambda(T_\ell)\big\} - \int_0^T \lambda(s)\,\mathrm{d}s.$$

For particular choices of the excitation function, such as the density of an exponential distribution, the integral term can be evaluated explicitly.

The Hawkes process can furthermore be extended to a multivariate setting in which several counting processes are allowed to mutually excite one another. In this case, the intensity of each process contains, in addition to its background rate, a

contribution from each process $j = 1, \ldots, d$, describing how arrivals in process j affect its future intensity.

Finally, the counting process may be equipped with marks. Let $M_\ell \in \mathcal{M}$ denote the mark associated with arrival time T_ℓ. If the mark space $\mathcal{M}$ is finite, the marked intensity may be written as

$$\lambda(t, m) = \lambda(t)p(m \mid t), \qquad \sum_{m \in \mathcal{M}} p(m \mid t) = 1,$$

where the filtration is now generated by $(T_\ell, M_\ell) : T_\ell < t$.

1.2.2 Contributions of Chapter 4

In Chapter 4, we propose to model portfolio evolution, in terms of purchases and cancellations, using a bivariate Hawkes process indexed by $j \in \{b, d\}$. One process counts the number of purchases ($j = b$), while the other counts the number of drops ($j = d$). The intuition behind modelling customer behaviour using self-exciting processes is the assumption that policyholders' actions tend to cluster in time. When a customer is already interacting with their insurance company, they may be more likely to consider purchasing additional coverage or to reconsider whether their existing policies are necessary. Conversely, customers may often go for long periods without interacting with their insurance company in their day-to-day lives.

Further, inspired by Brockett et al. (2008), Guillén et al. (2009), and Guillén et al. (2012), we model the background rate as

$$\mu^j\{t, X^j(t-)\} = \exp\{\beta^j_{\mu,0} + X^j(t-)^\top \beta^j_\mu\},$$

where X^j and β^j_μ denote the covariates and corresponding covariate effects for process j, respectively, and $\beta^j_{\mu,0}$ is the baseline log-rate.

Assuming exponential decay of the excitation effects, this gives the intensity

$$\lambda^j(t \mid \mathcal{F}_{t-}) = Y^j(t)\Bigg[\exp\{\beta^j_{\mu,0} + X^j(t-)^\top \beta^j_\mu\} + \sum_{k \in \{b,d\}} \eta^{jk} \sum_{\ell : T^k_\ell < t} \gamma^j \exp(-\gamma^j (t - T^k_\ell))\Bigg],$$

for $j \in \{b, d\}$. Here, η^{jk} determines the excitation of process j induced by jumps in process k, occurring at times T^k_ℓ. The at-risk indicators are given by

$$Y^{\mathrm{b}}(t) = \mathbf{1}\{t \leq C, \text{not yet churned and not fully covered}\}, \quad \text{and}$$
$$Y^{\mathrm{d}}(t) = \mathbf{1}\{t \leq C, \text{not yet churned}\},$$

where C denotes the right-censoring time, corresponding to the end of the observation period.

The specific products added or dropped at an event can be incorporated into the model by assigning marks to the counting processes. Together with the components described above, this gives rise to a broad range of possible model specifications. Model selection can be performed using classical likelihood-based methods. Alternatively, we implement a simulation algorithm that uses thinning to generate trajectories from a Hawkes process with the fitted parameters. This allows us to evaluate fitted models using a wide range of non-parametric scoring metrics. More generally, the simulator facilitates a range of further analyses, including, for example, what-if scenarios and bootstrap procedures.

The proposed model is fully parametric, which provides a transparent and interpretable description of the mechanisms driving portfolio evolution. With sufficiently rich data, however, parts of the specification may be made more flexible. In particular, the background rate can be modelled non-parametrically to accommodate nonlinear effects and interactions that are difficult to capture within a fixed parametric form. This increased flexibility comes at the cost of reduced transparency and may also introduce fairness concerns through the use of complex relationships between policyholder characteristics and predicted behaviour. In such settings, the methods developed in Chapters 2 and 3 provide complementary tools for analysing the resulting model in terms of interpretability and fairness.

Chapter 2

Fast Estimation of Partial Dependence Functions using Trees

This chapter contains the paper Liu et al. (2025).

Abstract

Many existing interpretation methods are based on Partial Dependence (PD) functions that, for a pre-trained machine learning model, capture how a subset of the features affects the predictions by averaging over the remaining features. Notable methods include Shapley additive explanations (SHAP) which computes feature contributions based on a game theoretical interpretation and PD plots (i.e., 1-dim PD functions) that capture average marginal main effects. Recent work has connected these approaches using a functional decomposition and argues that SHAP values can be misleading since they merge main and interaction effects into a single local effect. However, a major advantage of SHAP compared to other PD-based interpretations has been the availability of fast estimation techniques, such as `TreeSHAP`. In this paper, we propose a new tree-based estimator, `FastPD`[1], which efficiently estimates arbitrary PD functions. We show that `FastPD` consistently estimates the desired population quantity – in contrast to path-dependent `TreeSHAP` which is inconsistent when features are correlated. For moderately deep trees, `FastPD` improves the complexity of existing methods from quadratic to linear in the number of observations. By estimating PD functions for arbitrary feature subsets, `FastPD` can be used to extract PD-based interpretations such as SHAP, PD plots and higher-order interaction effects.

[1]The implementation is available as an R-package on GitHub: `https://github.com/PlantedML/glex`

2.1 Introduction

As machine learning models become increasingly complex and widely deployed in mission-critical applications, interpretability has become essential for ensuring fairness and transparency (Adadi and Berrada, 2018). One popular model explanation method is Shapley additive explanations (SHAP) (Lundberg et al., 2020) – a post-hoc explanation method that has gained traction for its game-theoretic approach of attributing feature importance based on Shapley values. A value function must be specified for the Shapley value. In this paper we refer to SHAP as the Shapley values that use the partial dependence (PD) functions as the value function. Others (Chen et al., 2020; Taufiq et al., 2023) have termed it interventional Shapley values.

PD plots, i.e. one-dimensional PD functions (Friedman, 2001; Hastie et al., 2009; Molnar et al., 2023) quantify the effect of individual features by averaging predictions while holding a target feature at fixed values. However, as noted by Hiabu et al. (2023), both PD plots and SHAP do not provide a complete model interpretation. SHAP merges main effects and interactions while PD plots ignore interactions thereby limiting insight into feature contributions.

Alternatively, a functional decomposition allows for greater insight by clearly separating main effects and higher-order interactions. This idea has previously been investigated in Stone (1994); Hooker (2007); Chastaing et al. (2012); Lengerich et al. (2020); Herren and Hahn (2022) under the functional ANOVA identification constraint and more generally in Bordt and von Luxburg (2023); Fumagalli et al. (2025). In this paper, we consider an identification constraint based on PD functions for which we develop a fast algorithm.

Computational efficiency improvements for SHAP have been proposed in several contexts: model-agnostic methods such as FastSHAP Jethani et al. (2022); deep-network adaptations (Ancona et al., 2019; Wang et al., 2022); and tree-based approaches via TreeSHAP (Lundberg et al., 2020). The popular path-dependent variant of TreeSHAP—used in XGBoost (Chen and Guestrin, 2016) and LightGBM (Ke et al., 2017)—is derived from approximating PD functions (Friedman, 2001). Recent work has further optimized these algorithms (Yang, 2022; Yu et al., 2022) and extended them to efficient SHAP interaction computations (Muschalik et al., 2024).

2.1.1 Contribution

We propose `FastPD`, a novel and efficient tree-based algorithm for consistently estimating arbitrary PD functions. `FastPD` can be used to obtain well-known PD-based explanations such as SHAP and PD plots. In addition it can be used to extract a functional decomposition that fully characterizes the target function with little computational cost. Finally, we show that path-dependent `TreeSHAP` can be an inconsistent estimate of the population SHAP value and demonstrate in a simula-

tions study that there can be substantial differences between the path-dependent estimates and `FastPD` estimates. The remainder of this work is structured as follows. Section 2.2 introduces PD functions and their role in functional decomposition. Section 2.3 discusses estimation methods, computational complexity, and presents `FastPD`. Section 3.6 compares `FastPD` with existing PD-based explanation techniques.

2.1.2 Notation

For all $k \in \mathbb{N}$, we let $[k] \coloneqq \{1, \dots, k\}$, and for any subset $S \subseteq [d]$, define $\overline{S} \coloneqq [d] \setminus S$. For $x \in \mathcal{X} \subseteq \mathbb{R}^d$, we use the notation $x_S \in \mathcal{X}^S$ to represent the coordinates of x corresponding to the indices in S. Random variables are denoted by capital letters. Lastly for a d-dimensional function $m : \mathcal{X} \longrightarrow \mathbb{R}$, with a slight abuse of notation, we will write $m(x_S, x_{\overline{S}})$, nevertheless with the interpretation that the coordinates are permuted into the right order before applying m.

2.2 Motivation: PD-Based Explanations

Consider a real multivariate function $m : \mathbb{R}^d \supseteq \mathcal{X} \longrightarrow \mathbb{R}$ and a distribution P_X on $\mathcal{X}$ with full support. For example, m could be a black-box machine learning model for estimating the credit score of a customer and P_X a distribution describing the customer base (see also Section 2.2.1). Our goal is to understand how changes to individual coordinates affect the function value. To this end, we consider a functional decomposition $\{m_S \mid S \subseteq [d]\}$ of m such that for all $x \in \mathcal{X}$

$$\begin{aligned} m(x) = \; & m_0 + \sum_{k=1}^{d} m_k(x_k) \\ & + \sum_{k<l} m_{kl}(x_{k,l}) + \cdots + m_{1,\dots,d}(x) \\ = \; & \sum_{S \subseteq [d]} m_S(x_S). \end{aligned} \tag{2.1}$$

Unfortunately, without further assumptions such a decomposition is not unique. We will consider an identification strategy due to Hiabu et al. (2023), and assume that for all $S \subseteq [d]$ and $x \in \mathcal{X}$

$$\sum_{U \subseteq S} m_U(x_U) = v_S(x_S), \tag{2.2}$$

where $v_S : \mathcal{X}^S \longrightarrow \mathbb{R}$ are the PD functions defined as

$$v_S(x_S) \coloneqq \mathbb{E}_{P_X}[m(x_S, X_{\overline{S}})]. \tag{2.3}$$

The identification constraint (2.2) leads to the unique solution

$$m_S(x_S) = \sum_{U \subseteq S} (-1)^{|*|S \setminus U} v_U(x_U), \tag{2.4}$$

which is known as the Möbius inverse (Rota, 1964) in combinatorics and as the Harsanyi dividend (Harsanyi, 1963) in cooperative game theory.

The PD function v_S can be interpreted as the expected value of the function m if the coordinates x_k for all $k \in S$ are kept fixed while the remaining coordinates $k \in \overline{S}$ vary according to the distribution $P_{X_{\overline{S}}}$. This interpretation of PD functions directly carries over to sums of the components in the functional decomposition of the form $\sum_{U \subseteq S} m_U(x_U)$ via (2.2). Furthermore, the functional decomposition decomposes the function m into additive contributions that together make up the whole function m. To illustrate this, consider the following two-dimensional example.

Example 2.2.1 (Functional decomposition). Assume that for $x_1, x_2 \in \mathbb{R}$ the PD functions take values

$$v_0 = 5,\ v_1(x_1) = 10,\ v_2(x_2) = 3,\ v_{1,2}(x_1, x_2) = 12.$$

Then from (2.4) the functional decomposition can be obtained as

$$m_0 = 5,\ m_1(x_1) = 5,\ m_2(x_2) = -2,\ m_{1,2}(x_1, x_2) = 4,$$

with the interpretation that fixing the x_1-value adds $m_1(x_1) = 5$ to the baseline prediction $v_0 = m_0 = 5$, leading to $v_1(x_1) = 10$. On the other hand, fixing both x_1 and x_2 and assuming no interaction, we would expect an output of $m_0 + m_1(x_1) + m_2(x_2) = 5 + 5 - 2 = 8$, but since the actual expectation is $v_{1,2}(x_1, x_2) = 12$, we have an interaction effect of $m_{1,2}(x_1, x_2) = 4$.

Two popular quantities used to capture feature contribution are PD plots and SHAP values. The latter can be defined for all $k \in [d]$ and all $x \in \mathcal{X}$ via a game theoretical motivation (Štrumbelj and Kononenko, 2010) as

$$\begin{aligned} \Delta(S, k, x) &:= v_{S\cup\{k\}}(x_{S\cup\{k\}}) - v_S(x_S), \\ \phi_k(x) &:= \frac{1}{d} \sum_{S \subseteq [d]\setminus\{k\}} \binom{d-1}{|S|}^{-1} \cdot \Delta(S, k, x). \end{aligned} \tag{2.5}$$

Hiabu et al. (2023) showed that a functional decomposition can represent both the PD plots and the SHAP values as

$$v_k(x_k) = m_0 + m_k(x_k), \tag{PD plot}$$

$$\phi_k(x) = m_k(x_k) + \tfrac{1}{2} \sum_j m_{kj}(x_{kj}) + \cdots + \tfrac{1}{d} m_{1,\ldots,d}(x_{1,\ldots,d}). \tag{SHAP value}$$

These expansions illustrate that PD plots capture the main effects in the functional decomposition at a specific coordinate value x_k, while SHAP values aggregate main effects and interaction effects of all orders. As a result, neither PD plots nor SHAP values fully capture the behavior of m, leading to potential misinterpretations as illustrated in Example 2.2.2 – a similar argument is made by Hiabu et al. (2023).

Example 2.2.2 (PD plots and SHAP values do not fully capture m). Consider the function $m : \mathbb{R}^2 \longrightarrow \mathbb{R}$ defined for all $x \in \mathbb{R}^d$ by $m(x_1, x_2) \coloneqq x_1 + 2x_1x_2$, and let P_X be a distribution with mean zero. Then, $\{m_0, m_1, m_2, m_{1,2}\}$ defined for all $x \in \mathbb{R}^2$ by

$$\begin{aligned} m_0 &= 2\mathbb{E}_{P_X}[X_1X_2], \\ m_1(x_1) &= x_1 - 2\mathbb{E}_{P_X}[X_1X_2], \\ m_2(x_2) &= -2\mathbb{E}_{P_X}[X_1X_2], \\ m_{1,2}(x_1, x_2) &= 2x_1x_2 + 2\mathbb{E}_{P_X}[X_1X_2], \end{aligned}$$

is the unique functional decomposition satisfying (2.1) and (2.2). For the SHAP value ϕ_1 we get

$$\phi_1(x_1, x_2) = x_1 + x_1x_2 - \mathbb{E}_{P_X}[X_1X_2].$$

The PD plot $m_0 + m_1$ only captures the average dependence on x_1, which does not provide any insight on the interaction between x_1 and x_2. Similarly, the SHAP value ϕ_1 only captures part of m as it down-weights the interaction contribution between x_1 and x_2.

Instead of focusing on PD plots and SHAP values, we therefore advocate to estimate the full functional decomposition $\{m_S \mid S \subseteq [d]\}$ defined via (2.1) and (2.2). However, this entails estimation of all PD functions $\{v_S \mid S \subseteq [d]\}$. Unlike for SHAP values, where many fast estimation techniques are available (e.g., `TreeSHAP`), no fast algorithms have been proposed to estimate all PD functions. In Section 2.3 we propose such an algorithm that efficiently estimates all PD functions which can then be further used to extract PD-based explanations such as SHAP values and the functional decomposition with no significant extra cost.

2.2.1 What Is The Target m?

It is worth differentiating two use-cases of PD-based explanations: (i) When the function m represents a pre-trained black-box model $\hat{m}$, such as a neural network or tree ensemble, and the sole focus is to understand the model without drawing conclusions about the underlying data-generating process. (ii) When the emphasis is on the relationship between input features X and a response Y. In this scenario, the machine learning model $\hat{m}$ is used to approximate a target function m^* (e.g., the conditional mean $m^* : x \mapsto \mathbb{E}[Y \mid X = x]$ or the conditional quantile $m^* : x \mapsto \inf\{t \in \mathbb{R} \mid \mathbb{P}(Y \leq t \mid X = x) \geq \alpha\}$) which is the actual function we wish to explain.

It has been argued (e.g. Chen et al., 2020) that for the case (ii), when m^* is the conditional expectation function, it can make sense to consider the functions $x^S \mapsto \mathbb{E}_{P_X}[Y \mid X_S = x_s] = \mathbb{E}_{P_X}[m^*(x_s, X_{\overline{S}}) \mid X^S = x^s]$ instead of PD functions. Janzing et al. (2020) argue from a causal perspective that PD functions are generally

preferable to this approach and easier to interpret. We tend to agree with their arguments and only consider PD functions as defined in (2.3).

In this paper, we consider both cases $m = \hat{m}$ and $m = m^*$ as possible targets. We formally distinguish them by considering PD functions in two settings:

(i) The model PD function

$$v_S^{\hat{m}}(x_S) = \mathbb{E}_{P_X}[\hat{m}(x_S, X_{\overline{S}})]. \tag{2.6}$$

(ii) The ground truth PD function

$$v_S^{m^*}(x_S) = \mathbb{E}_{P_X}[m^*(x_S, X_{\overline{S}})]. \tag{2.7}$$

In both cases, PD-explanations are applied to a trained machine learning model $\hat{m}$. However, if the target is the ground truth PD function (ii), then additional assumptions are required to ensure valid explanations. An important point is that the ground truth PD function is only identified in settings in which P^X is a product measure or regularity conditions are made that avoid unidentifiability due to extrapolation (e.g., assuming a parametric model). In contrast, such assumptions are not necessary in case (i), as it can, in fact, be of interest to understand the behaviour of $\hat{m}$ outside the training support.

Most algorithms for PD-based explanations rely on $\hat{m}$ being a specific machine learning model. This is also the case for `FastPD`, which supposes that $\hat{m}$ is a tree-based model. A possible work-around to this is to train a new (surrogate) tree-based model on $(X^{(1)}, \hat{m}(X^{(1)})), \ldots, (X^{(n)}, \hat{m}(X^{(n)}))$ and afterwards apply `FastPD` to this model. However, the additional approximation steps then lead to the same potential difficulties as in case (ii).

2.3 Estimation of PD Functions

The most direct approach to computing PD-based explanations is to directly compute the PD functions v_S, defined in (2.3), for all $S \subseteq [d]$ required to compute the desired PD-based explanation. In most practical examples one does not have access to the data generating mechanism P_X directly and therefore need to estimate it. For example, if m is a prediction model for fraud detection and P_X is the future distribution of customers' features the algorithm will be applied on, one may use (parts) of the current customer base as background data to approximate P_X. Here, we consider the case in which we observe a background sample $\mathcal{D}_{n_b} = \{X^{(1)}, \ldots, X^{(n_b)}\}$ consisting of n_b iid samples from P_X based on which we want to estimate the PD-based explanations. An obvious estimator in this setting is the empirical PD function, defined for all $S \subseteq [d]$ and all $x \in \mathcal{X}$ by

$$\hat{v}_S(x_S) = \frac{1}{n_b} \sum_{i=1}^{n_b} m(x_S, X_{\overline{S}}^{(i)}). \tag{2.8}$$

Table 2.1: Comparison of the algorithmic complexity to estimate either the SHAP values or PD functions for all features and n_e evaluation samples in a single decision tree using n_b background samples. Here d denotes the total number of features, D the depth of the tree, F the number of features the tree splits on and R the operations required for a single model evaluation (for a single tree this is $O(D)$). The method of Friedman (2001) can estimate SHAP values from the PD functions (2.5).

Method	**Complexity (SHAP)**	$v_S(x)$?	**Details**
`VanillaPD`	$O(R2^d n_\mathrm{e} n_\mathrm{b})$	Yes	Slow if d is large, but applicable to all models
Friedman (2001)	$O(2^d 2^D n_e)$	Yes	Only approximates PD functions, with same inconsistency as `TreeSHAP-path`
`TreeSHAP-path`	$O(D^2 2^D n_\mathrm{e})$	No	Fast but inconsistent if features are correlated
`TreeSHAP-int`	$O(D2^D n_\mathrm{e} n_\mathrm{b})$	No	Estimates are based on ≤ 100 background samples[2]
Zern et al. (2023)	$O(2^D n_b + 3^D D n_e)$	No	Fast and consistent with any number of background samples
`FastPD`	$O(2^{D+F}(n_\mathrm{e} + n_\mathrm{b}))$	Yes	Fast and consistent for any PD-based explanations

In a model-agnostic setting, using (2.8) to estimate the PD function v_S for a fixed S at a single evaluation point x has a complexity of $O(Rn_b)$ where R is the number of operations needed to evaluate m at a single point. We usually refrain from evaluating the PD functions at all points in the domain $\mathcal{X}$, and instead only evaluate it at a set of n_e evaluation points that depend on the precise application. Consequently, computing (2.8) for all n_e evaluation points and all sets $S \subseteq [d]$ results in a complexity of $O(R2^d n_e n_b)$, which in many applications is intractable. We call this baseline approach of estimating PD functions `VanillaPD`. In Table 2.1 we compare its complexity to tree-based alternatives which we discuss next.

2.3.1 Tree-Based Methods

If m is a decision tree, the complexity of obtaining various PD-based explanations can be substantially reduced. The earliest example we are aware of is an algorithm proposed by Friedman (2001) which proposed to compute PD functions by traversing the tree weighting predictions based on the coverage of each node. Assuming that $2^D < Rn_b$, where D denotes the depth of the tree, it has a reduced complexity of

[2]See GitHub issue: `https://github.com/shap/shap/issues/3461`

$O(2^d 2^D n_e)$, compared to `VanillaPD`. The tree-based algorithm of Friedman (2001) is one of the only implementations we found that attempts to estimate the PD functions, however, by construction it only roughly approximates (2.8). As with `TreeSHAP-path` (discussed next) this can lead to inconsistent estimates (see Proposition 2.3.1). More Recent approaches have focused on estimating SHAP directly and hence do not provide estimates of the PD functions. The most notable algorithm is TreeSHAP (Lundberg et al., 2020), which comes in two variants: Path-dependent TreeSHAP (`TreeSHAP-path`) and interventional TreeSHAP (`TreeSHAP-int`). We first examine `TreeSHAP-path`: By exploiting the tree structure, `TreeSHAP-path` reduces the factor $2^d n_b$ in the complexity of `VanillaPD` to $2^D D$. While this method is computationally efficient, the SHAP values it computes rely on the same approximation of the PD functions as the algorithm proposed by Friedman (2001). This leads to the undesirable property that for two distinct trees that have the exact same predictions, the estimated SHAP values may differ when the features are not independent. In particular for post-hoc explanations of a black box model, this dependence on the internals of the model is concerning. Moreover, even in the limit of infinite (background) data the SHAP values estimated by `TreeSHAP-path` do not necessarily converge to the model SHAP value. A formal statement of this inconsistency is provided in the following Proposition 2.3.1. A proof is given in the Appendix.

Proposition 2.3.1 (Inconsistency of `TreeSHAP-path`)**.** *There exists a distribution P_X on $\mathcal{X}$, evaluation point $x' \in \mathcal{X}$ and distinct trees $\hat{m}^A$ and $\hat{m}^B$ such that*

(i) $\forall x \in \mathcal{X} : \hat{m}^A(x) = \hat{m}^B(x)$, *but* $\hat{\phi}_k^{\hat{m}^A}(x', \mathcal{D}_{n_b}) \neq \hat{\phi}_k^{\hat{m}^B}(x', \mathcal{D}_{n_b})$,

(ii) $\lim_{n_b \to \infty} \left| \widehat{\phi}_k^{\hat{m}^A}(x', \mathcal{D}_{n_b}) - \phi_k^{\hat{m}^A}(x') \right| > 0 \qquad a.s.$,

where $\mathcal{D}_{n_b} \coloneqq \{X^{(i)}, \ldots, X^{(n_b)}\}$ consists of iid samples from P_X, $\phi_k^{\hat{m}}(x')$ denotes the population SHAP value computed via the model PD function and $\widehat{\phi}_k^{\hat{m}}(x', \mathcal{D}_{n_b})$ denotes the `TreeSHAP-path` explanation of feature k at x', where $\mathcal{D}_{n_b}$ is used to compute the coverage probability of $\hat{m}$.

`TreeSHAP-int` was proposed as a method that consistently estimates the SHAP values. However, it has a rather high complexity of $O(D 2^D n_e n_b)$ that scales with the product $n_e n_b$. Recently, Zern et al. (2023) have reduced the complexity by considering all background samples simultaneously when traversing the tree, yielding an improved complexity of $O(2^D n_b + 3^D D n_e)$. The algorithm by Zern et al. (2023) is an improvement on `TreeSHAP-int` which however does not estimate PD functions and instead only estimates the differences $\Delta(S, k, x)$ in (2.5). While this approach reduces the complexity of estimating SHAP values it does not allow us to efficiently extract estimates of the PD functions.

In the following section, we propose `FastPD` which estimates all PD functions at a similar complexity to Zern et al. (2023). From this, the SHAP values and the

complete functional decomposition $\{m_S \mid S \subseteq [d]\}$ can be extracted with practically no additional cost, providing a more complete explanation of m.

2.3.2 `FastPD` Algorithm

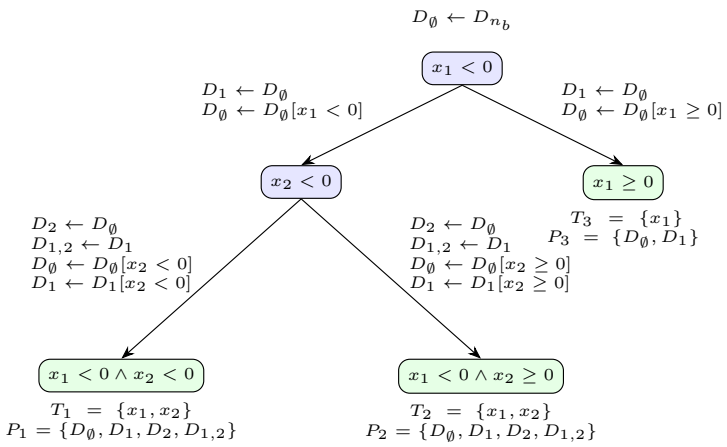


Figure 2.1: Augmentation step in `FastPD`. Augmentation entails calculating sets $D_S^{(j)}$ which contain observations that would have reached leaf j if splits on features in S were ignored. Starting from the full background sample D_{n_b} and setting $D_\emptyset \leftarrow D_{n_b}$, each subsequent split creates new sets and updates existing ones. Here, $D_S[x_j < c] \coloneqq \{i \in D_S : X_j^{(i)} < c\}$.

In this section, we introduce `FastPD`, which substantially reduces the complexity of computing $\hat{v}_S(x_S)$ for an evaluation point x compared with `VanillaPD`. It operates in two main steps: (1) a tree augmentation step that precomputes partial dependence information using the n_b background samples in D_{n_b}; and (2) an evaluation step that retrieves partial dependence functions for any desired feature subsets on n_e evaluation points.

The naïve estimator `VanillaPD` performs step 1 and step 2 for each evaluation point and background sample together, which leads to a complexity of $O(n_e n_b)$ in the number of the samples. The main observation used by `FastPD` is that the two steps can be separated entirely by exploiting the tree structure, thus resulting in a complexity of $O(n_e + n_b)$. To motivate the `FastPD` algorithm, we begin by noting that a decision-tree $\hat{m}$ can be expressed as a weighted sum of indicator functions. More concretely, there exists $L \in \mathbb{N}$, $c_1, \ldots, c_L \in \mathbb{R}$ and $A_1, \ldots, A_L \subseteq \mathbb{R}^d$ such that for all $x \in \mathcal{X}$

$$\hat{m}(x) = \sum_{j=1}^{L} c_j \mathbb{1}_{A_j}(x).$$

Furthermore, the leaves A_j are rectangles, such that for each $j \in [L]$, $A_j := [a_{j1}, b_{j1}] \times [a_{j2}, b_{j2}] \times \cdots \times [a_{jd}, b_{jd}]$ determines the bounds of a leaf node in the tree. For all $S \subseteq [d]$ and $j \in [L]$ define the bounds of features S on leaf j as $A_j(S) := \prod_{s \in S} [a_{js}, b_{js}]$. By substituting m with $\hat{m}$ in (2.8) and simplifying, we obtain

$$\hat{v}_S(x_S) = \sum_{j=1}^{L} c_j \underbrace{\mathbb{1}_{A_j(S)}(x_S)}_{\text{(i)}} \cdot \underbrace{\hat{P}\left(X_{\overline{S}} \in A_j(\overline{S})\right)}_{\text{(ii)}}, \tag{2.9}$$

where $\hat{P}$ denotes the empirical distribution of the background data $\mathcal{D}_{n_b}$, that is, $\hat{P}\left(X_{\overline{S}} \in A_j(\overline{S})\right) = {n_b}^{-1} \sum_{i=1}^{n_b} \mathbb{1}(X_{\overline{S}}^{(i)} \in A_j(\overline{S}))$. The factor (i) in (2.9) identifies the leaves in which x would have landed if the splits corresponding to $\overline{S}$ were ignored during traversal, while the factor (ii) in (2.9) represents the proportion of observations for which the features in $\overline{S}$ fall within the bounds of the leaf.

A naïve computation of (ii) in (2.9) loops over all observations and checks whether the feature in $\overline{S}$ lie within $A_j(\overline{S})$. The idea of `FastPD` is to separate the computation of partial dependence information on the background samples from evaluating on evaluation points into two phases:

Augmentation (Algorithm 2.1): For each leaf $j \in [L]$, we identify the set of features T_j encountered along the path to that leaf. For every subset $S \subseteq T_j$, we save a corresponding list $D_S^{(j)}$ that contains the observations that would have reached leaf j if splits on features in S were ignored. For any sample i it follows that $i \in D_S^{(j)}$ if and only if $X_{\overline{S}}^{(i)} \in A_j(\overline{S})$. Consequently, factor (ii) in (2.9) can be computed efficiently as the ratio $|D_S^{(j)}|/n_b$.

Evaluation (Algorithm 2.2, Appendix 2.B): Once the tree is augmented, the PD function $\hat{v}_S(x_S)$ at any fixed evaluation point x and set S can be computed quickly. Due to the augmentation step, every subset $S \subseteq T_j$ has a corresponding list $D_S^{(j)}$ saved in P_j on leaf j. To compute $\hat{v}_S(x_S)$ for a point x and features S, we can first intersect S with the tree's split features to obtain a subset U of features that is actively used by the tree. Afterwards, we traverse the tree to every leaf j in which $x_U \in A_j(U)$.

The procedure guarantees that we arrive at the leaves j in which the indicator (i) in (2.9) is equal to one. Once we are at leaf j, we can compute the empirical probability (ii) in (2.9) by counting the samples in $D_{U_j}^{(j)}$ where $U_j = U \cap T_j$. Note that $D_{U_j}^{(j)}$ exists in P_j since $U_j \subseteq T_j$.

Lastly, because different subsets S may lead to the same U when intersected with the split features, redundant computations are avoided by saving $\hat{v}_U(x_U)$ (see lines 3 and 22 in Algorithm 2.2). For example, if the tree-depth is less than the number of features, then $U = \emptyset$ may occur often and the computed PD functions can be saved.

The consistency of `FastPD` follows from the consistency of the empirical estimator (2.8) which it computes exactly. A proof can be found in the Appendix.

Complexity of `FastPD`

The main reduction in complexity of `FastPD` stems from separating the computation of the partial dependence information on D_{n_b} from the evaluation on n_e evaluation points, which breaks a product into a sum.

To explicitly bound the algorithmic complexity of `FastPD` we can proceed as follows. Let F be the number of unique features the tree has split. It will always hold that $F \leq d$ and usually the inequality is strict. We start at the tree root with a list containing all observations and assign that to the set $S = \emptyset$, these lists are recursively passed to the child nodes. New lists $D_{S\cup\{k\}} := D_S$ are created for all S when a new feature, k, is encountered on the path. This means that the total number of lists to keep track of will be at most 2^F on each node if every split feature was unique. For every node, every list incurs a maximum of n_b operations, yielding a very rough bound of $O(2^{D+F}n_b)$ for the worst-case complexity of augmenting the whole tree. Once the tree has been augmented, the complexity of traversing all nodes is $O(2^D)$, and doing it for all 2^F subsets and evaluation points will result in a complexity of $O(2^{D+F}n_e)$.

Thus, if D and therefore also F, are not too big – which usually is the case for gradient boosted trees as in as XGBoost (Chen and Guestrin, 2016) and LightGBM (Ke et al., 2017) – even with a large number of features, the complexity of computing the PD function for all subsets S is reduced. This is because the PD functions are calculated separately for every tree and then summed together. Hence for trees with moderate depth D, the main complexity does not stem from the number of subsets $S \subseteq [d]$ for which v_S needs to be estimated, but in the traversal of all background samples for every new point that needs to be explained. If $n = n_b = n_e$, then both `VanillaPD` and `TreeSHAP-int` will scale proportionally to n^2. We hence gain a significant speed-up by reusing the computed empirical probabilities at the leaves whenever new points are explained.

Proposition 2.3.2 (Consistency of `FastPD`)**.** *Let $m : \mathcal{X} \to \mathbb{R}$ be a bounded target function and P_X a distribution on $\mathcal{X}$. Then, for a sequence of iid background samples $X^{(1)}, X^{(2)}, \ldots \sim P_X$, it holds for all $S \subseteq [d]$ that $\lim_{n_b\to\infty} \hat{v}^m_{S,n_b}(x_S) = v^m_S(x_S)$ a.s., where $\hat{v}^m_{S,n_b}$ is the estimate for v^m_S from `FastPD` applied with background data $\mathcal{D}_{n_b} = \{X^{(1)}, \ldots, X^{(n_b)}\}$. Moreover, if $\hat{m}_{n_b}$ is a uniformly consistent estimate of m trained on $\mathcal{D}_{n_b}$, i.e., $\lim_{n_b\to\infty} \sup_{x\in\mathcal{X}} |\hat{m}_{n_b}(x) - m(x)| = 0, a.s.$ then*

$$\lim_{n_b\to\infty} \hat{v}^{\hat{m}_{n_b}}_{S,n_b}(x_S) = v^m_S(x_S) \quad a.s..$$

2.4 Experiments

We consider a supervised learning setup where training data $\mathcal{D}^{\text{train}}_n := \{(Y^{(i)}, X^{(i)})\}_{i\in[n]}$ are iid sampled from a distribution P with correlated features

Algorithm 2.1 `FastPD` augmentation step. Nodes are indexed by j, where l_j and r_j represent the indices of the left and right child nodes, respectively. The feature used to split at node j is denoted by d_j, and t_j is the split threshold and v_j the value.

input Tree structure $= \{v, l, r, t, d\}$ and dataset D_{n_b}
output Augmented data $T = \{T_j \mid j \text{ is leaf}\}, P = \{P_j \mid j \text{ is leaf}\}$

function RECURSE (j, T, P)
  if j is leaf **then**
    $T_j \leftarrow T$
    $P_j \leftarrow P$
    return
  end if
  for all $(S, D_S) \in P$ **do**
    if $d_j \in S$ **then**
      $P_{\text{yes}} \leftarrow P_{\text{yes}} \cup \{(S, D_S)\}$
      $P_{\text{no}} \leftarrow P_{\text{no}} \cup \{(S, D_S)\}$
    else
      $P_{\text{yes}} \leftarrow P_{\text{yes}} \cup \{(S, D_S[x_{d_j} < t_j])\}$
      $P_{\text{no}} \leftarrow P_{\text{no}} \cup \{(S, D_S[x_{d_j} \geq t_j])\}$
    end if
  end for
  $T_{\text{new}} \leftarrow T$
  if $d_j \notin T$ **then**
    $T_{\text{new}} \leftarrow T \cup \{d_j\}$
    for all $(S, D_S) \in P$ **do**
      $P_{\text{yes}} \leftarrow P_{\text{yes}} \cup \{(S \cup \{d_j\}, D_S)\}$
      $P_{\text{no}} \leftarrow P_{\text{no}} \cup \{(S \cup \{d_j\}, D_S)\}$
    end for
  end if
  RECURSE$(l_j, T_{\text{new}}, P_{\text{yes}})$
  RECURSE$(r_j, T_{\text{new}}, P_{\text{no}})$
end function
Initialize for all leaf nodes j: $T_j \leftarrow \emptyset$, $P_j \leftarrow \emptyset$ RECURSE$(0, T = \emptyset, P = \{(\emptyset, D_{n_b})\})$

$X^{(i)}$. For all experiments, an XGBoost estimator, $\hat{m}$, was trained on $\mathcal{D}_n^{\text{train}}$ and subsequently explained using the same training samples as background data.

Specific simulation settings varied depending on the analysis. For the inconsistency and MSE analyses (Figures 2.2, 2.3 and 2.5), we used $d = 2$ covariates sampled from a bivariate Gaussian distribution with correlation 0.3 and variance of 1. XGBoost hyperparameters (`nrounds` $\in \{1, \ldots, 1000\}$, `eta` $\in [0.01, 0.3]$, `max_depth` $\in \{2, \ldots, 6\}$) were tuned via 5-fold cross-validation with 50 random search evaluations. For the runtime comparison (Figure 2.4), we used $d = 7$ covariates sampled from a multivariate Gaussian distribution (details in Appendix) and a single fixed XGBoost model configuration with 20 trees and a maximum depth of $D = 5$. All other hyperparameters were left as default, except for `eta`, which was drawn uniformly

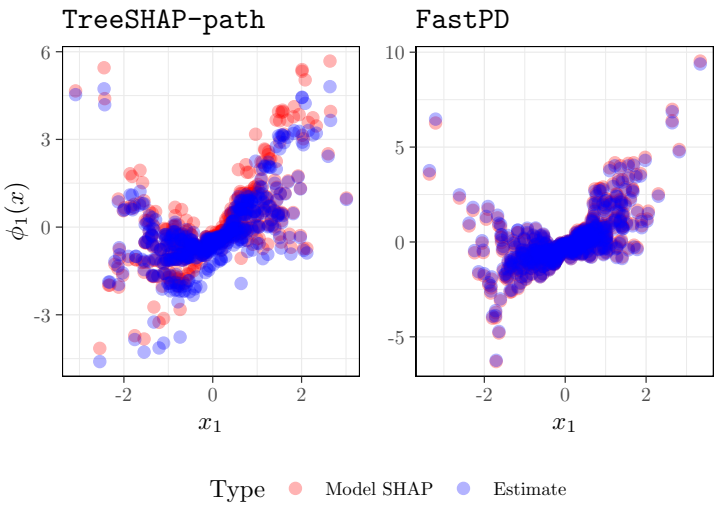


Figure 2.2: Comparison between `FastPD` (for SHAP) and `TreeSHAP-path` on simulated data of two covariates with correlation 0.3 and $n_b = 500$ background samples that are also used as evaluation points. For the two methods, the simulation run that achieved the median MSE was selected. We see that the model SHAP is captured well by the `FastPD` SHAP estimates (equivalent to `VanillaPD`) which is not the case for `TreeSHAP-path`.

from $[0.01, 0.3]$.

All simulations were conducted on a dedicated compute cluster (2 Intel Xeon Gold 6230 @ 2.1 GHz CPUs, 192 GB RAM). Our implementation of the `FastPD` algorithm is available as an R package on GitHub[3].

Inconsistency of `TreeSHAP-path` Figure 2.2 illustrates the SHAP explanations of $\hat{m}$ for X_1 in 500 observations of $(Y, X) \in \mathbb{R} \times \mathbb{R}^2$. We observe that `TreeSHAP-path` inconsistently estimates the model SHAP obtained via the model PD function. In contrast, `FastPD`, which estimates the PD function based on the same 500 samples used as the background data, is consistent and lies close to the model SHAP. In the setting of Figure 2.2, X_1 and X_2 have a correlation of 0.3 further correlations of $0, 0.1$, and 0.7 are considered in the Appendix.

Non-Decreasing MSE of `TreeSHAP-path`

Figure 2.3 depicts the mean squared errors (MSEs) of the different methods when the model SHAP is taken as the target. We observe that the MSE of `TreeSHAP-path` does not shrink with increasing number of background samples, whereas the MSE of `FastPD` decreases substantially, demonstrating that it is consistent towards the model SHAP. Lastly, we observe that accurate estimation might require more than

[3]R implementation of the `FastPD` algorithm: `https://github.com/PlantedML/glex`

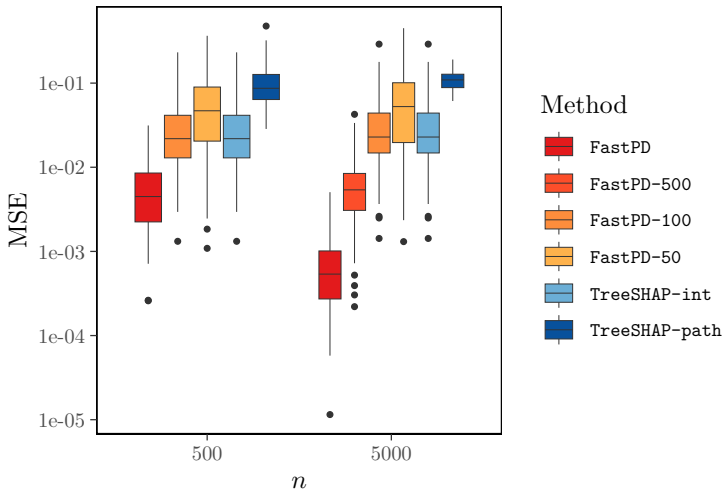


Figure 2.3: Comparsion of Mean Squared Error (MSE) of `TreeSHAP` versus `FastPD` over $B = 100$ simulations (log-scale). Each boxplot summarizes the results of $B = 100$ independent simulation runs. In each run, an XGBoost model $\hat{m}$ was trained using n observations, and the same training samples were resampled as background data to estimate the partial dependence (PD) function. For `FastPD`, different numbers of background samples were used: $\{50, 100, 500\}$ for $n = 500$ and $\{50, 100, 500, 5000\}$ for $n = 5000$. SHAP values were estimated using the training observations as evaluation points, and the mean squared error (MSE) was computed as $\frac{1}{n}\sum_{i=1}^{n}(\phi_1(x^{(i)}) - \hat{\phi}_1(x^{(i)}))^2$, where ϕ_1 denotes the model SHAP of feature X_1.

100 background samples (The Python package `shap` e.g. does not use more than 100 background samples). In the setting of Figure 2.3 X_1 and X_2 have a correlation of 0.3 further correlations of 0 and 0.7 are considered in the Appendix. Notably even when the correlation is zero, `FastPD` turns out to be more accurate then `TreeSHAP-path`. This is because `TreeSHAP-path` does not leverage all information available in the background samples whereby at each inner node, it conditions on the path taken by the evaluation point.

Comparison of Computational Runtime Figure 2.4 compares the runtime of extracting the PD functions for all S using `FastPD` with computing the interventional SHAP values as implemented in the SHAP Python package. An XGBoost model was pre-fitted with 20 trees and a max-depth of 5 on a fixed dataset of 8 000 observations. The number of background samples, n_b, was selected to be $1\,000, 2\,000, \ldots, 8\,000$. The same samples were used as evaluation points. `VanillaPD` and `TreeSHAP-int` scale quadratically in comparison to Zern et al. (2023) and `FastPD`, which has a linear complexity in the number of samples. While Zern et al. (2023) and `FastPD` have a comparable runtime, `FastPD` calculates all PD functions from which not only SHAP values but a complete functional decomposition can be derived.

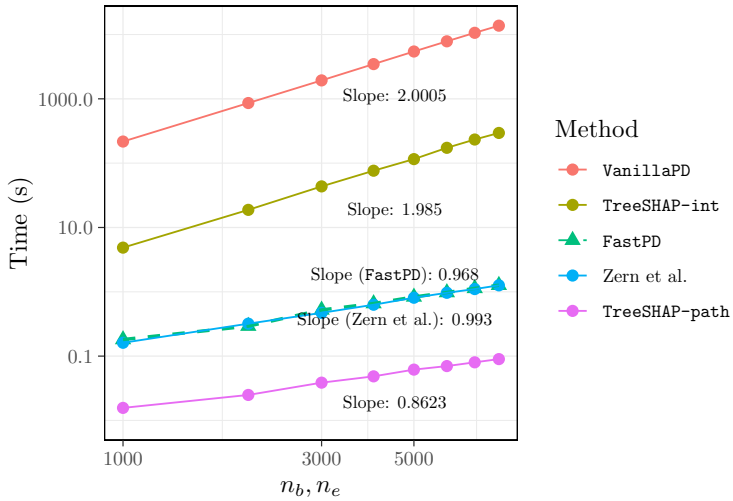


Figure 2.4: Runtime in seconds (s) on log-log scale between `FastPD` and other methods as a function of the number of background samples and evaluation points which are taken to be the same. Measurements are the mean times over $B = 100$ runs where an XGBoost model was fitted on the data with 20 trees and a depth of 5 each. The mean times for $n_b = n_e = 1000$ are (217s, 4.86s, 0.181s, 0.161s, 0.0157s) respectively and (13722s, 295s, 1.26s, 1.26s, 0.0899s) for $n_b = n_e = 8000$. The runtime of `TreeSHAP-path` depends only on the number of evaluation points n_e.

Obtaining Functional Components The functional components can be recovered from the estimated PD functions via (2.4). Figure 2.5 compares $m_1(x_1)$ from Example 2.2.2 with the functional component computed using `FastPD` and with the method proposed in Friedman (2001). The estimate of `FastPD` lies close to the component $\hat{m}_1(x_1)$ as one would have obtained via the model PD function, while the path-dependent Friedman (2001) suffers in areas outside the center. We also see that the component estimated by `FastPD-100` has a slope that is slightly off, which highlights the need to approximate the PD function with more background samples. In the setting of Figure 2.5, X_1 and X_2 have a correlation of 0.3 and a setting with correlation of 0.7 is given in the Appendix.

2.5 Real Data Experiments

2.5.1 Comparison on Benchmark Datasets OpenML-CTR23 and OpenML-CC18

We have conducted experiments on a significant number of curated datasets in both regression and classification settings. We used the OpenML-CTR23 (Fischer et al.,

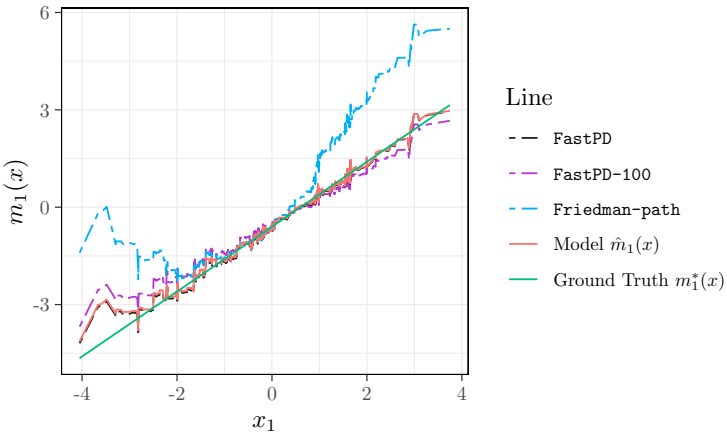


Figure 2.5: Comparison of estimated functional component m_1 between `FastPD` and the path-dependent method of Friedman (2001) (`Friedman-path`). The components are extracted from an XGBoost model trained on 5000 samples from the data-generating distribution. The model component ($\hat{m}_1$, red line) was computed by weighting the leaves using the true probabilities, while the ground truth component (m^*, green line) was computed analytically similar to Example 2.2.2.

2023) regression datasets and the OpenML-CC18 (Bischl et al., 2021) classification datasets. In these experiments, we explained the predictions of a XGBoost model via functional decomposition as done in Figure 2.5. The hyperparameters `nrounds` $\in \{10, \ldots, 200\}$, `max_depth` $\in \{1, \ldots, 5\}$, `eta` $\in [0, 0.5]$, `colsample_bytree` $\in [0.5, 1]$ and `subsample` $\in [0.5, 1]$ were tuned via random search with 5-fold cross-validation over 100 random search evaluations. For each dataset, we computed the following measure of variable importance

$$\text{Importance}(\hat{m}_S) = \frac{1}{n} \sum_{i=1}^{n} \left| \hat{m}_S(X_S^{(i)}) \right|. \tag{2.10}$$

Components $\hat{m}_S$ appearing in the top five of any method (`FastPD`, `FastPD-50`, `FastPD-100`, or `Friedman-path`) were compared in their importance attributions. Taking `FastPD` as the reference, the path-dependent algorithm exhibited relative differences exceeding $\pm 30\%$ in 33 of the 96 datasets. Applying `FastPD` with only 50 background samples (`FastPD-50`) reduced this to 18 datasets, and using 100 samples (`FastPD-100`) further lowered it to just 9. The full attribution results for each dataset are provided in the Appendix.

These experiments show that PD estimation accuracy improves with larger background samples. Our results suggest that a smaller background sample size may often suffice. Ideally, one would leverage the full dataset to compute PD functions; when this is not feasible, a practical strategy is to subsample (e.g. $n_b = 100$) and then checking whether the resulting PD functions closely match those from a larger

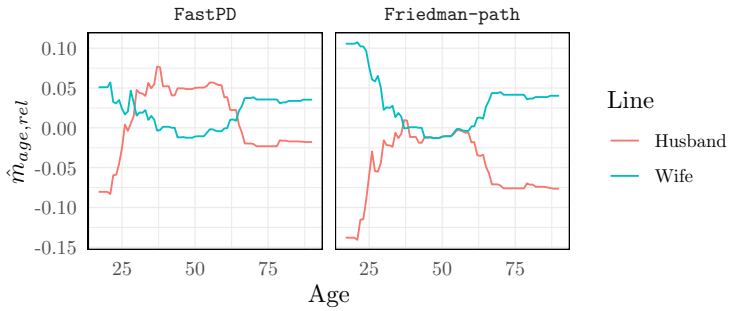


Figure 2.6: The component $\hat{m}_{age,rel}$ identified by `FastPD` and `Friedman-path` for the `adult` dataset. We see that `FastPD` estimates a positive effect for husbands during the prime working age, while `Friedman-path` estimates a zero effect for both husbands and wives.

sample (e.g. $n_b = 200$). Such a comparison provides a simple yet effective check on the adequacy of the reduced background set.

2.5.2 The Adult Dataset

A particularly illustrative additional example is the `adult` dataset which contains data on whether an individual's income exceeds $\$50,000$ per year (Becker and Kohavi, 1996). We ran randomized grid search with 5-fold CV to tune XGBoost hyperparameters (`max_depth` $\in \{3, \ldots, 7\}$, `eta` $\in \{0.01, 0.05, 0.1\}$, `nrounds` $\in \{100, 200, 300\}$, `min_child_weight` $\in \{1, 3, 5\}$, `subsample` $\in \{0.6, 0.8, 1.0\}$, `colsample_bytree` $\in \{0.6, 0.8, 1.0\}$) over 25 trials. We then visualized the `age`–`relationship` interaction component (Figure 2.6) and noticed that `FastPD` and `Friedman-path` offered conflicting interpretations: With `FastPD`, we notice that at prime working age there is a slight positive effect on income for husbands, while the effect is close to zero and/or slightly negative for wives. In contrast, the path-dependent algorithm `Friedman-path` estimates the effect to be zero for both wives and husbands at working age; suggesting that age has the same effect on a husband's and wife's income in this range.

2.6 Discussion

In this paper, we have proposed `FastPD`, an algorithm that consistently estimates model PD functions for tree-based models $\hat{m}$ with linear time complexity in the number of observations. Under additional assumptions, it also consistently estimates the ground truth PD function of the conditional expectation m^*. One limitation of Algorithm 2.2 is its space complexity, which increases with the number of lists at each leaf. However, once the tree is augmented, storing only the sample counts

instead of the full lists can improve memory usage, but this problem also motivates limiting the tree depth. Fortunately, existing gradient boosting algorithms such as XGBoost perform well with shallow trees[4], as increasing depth is likely to cause the model to overfit.

It is important to distinguish the *model* PD function from the *ground-truth* PD function. If the target is the ground-truth PD, our recommendation is to use `FastPD` as an exploratory visualization tool, to be supplemented by semiparametric or doubly-robust estimators (Kennedy et al., 2016; Chernozhukov et al., 2018).

Although `FastPD` can be used to recover functional components of any order, in practice higher-order interaction components may contribute negligibly; therefore one may choose to truncate the intitial black-box model at the main effects and pairwise interactions to improve interpretability. Additionally one can report the average discrepancy between the truncated model and the initial black-box model.

Finally, if the target is the ground-truth PD, a key challenge with PD-based explanations mentioned in Section 2.2.1 is that they can be distorted by extrapolation outside the support of P_X. Future work should investigate strategies to limit or correct for extrapolation, or consider alternative summaries such as average local effects $\mathbb{E}_{P_X}[\frac{\partial}{\partial x^j} m(X)]$ (ALE) (Apley and Zhu, 2020), which avoid these issues.

Acknowledgments

Niklas Pfister was supported by a research grant (0069071) from Novo Nordisk Fonden. Marvin Wright is supported by the German Research Foundation (DFG) under the grants 437611051 and 459360854. Tessa Steensgaard and Munir Hiabu are supported by the project framework InterAct.

Impact Statement

This paper contributes to the field of explainable AI (XAI) by providing a novel algorithm, `FastPD`, for estimating Partial Dependence (PD)-based explanations of tree-based models. XAI is a critical field focused on making complex machine learning models more transparent, trustworthy, and accountable. Our work addresses a key challenge in XAI: the need for computationally efficient methods that provide *consistent* and comprehensive explanations. While methods like SHAP are popular, some widely-used implementations can yield inconsistent results, potentially undermining trust. By enabling fast and consistent estimation of PD functions, `FastPD` allows for more reliable application of PD-based exaplanations, including SHAP and full functional decompositions. This enhanced reliability and the ability to disentangle main effects from interactions contribute directly to core XAI goals:

[4]XGBoost uses a default `max_depth` of 6, warning that deeper trees may increase memory usage aggressively: `https://xgboost.readthedocs.io/en/stable/parameter.html`

fostering user trust, enabling robust model debugging, facilitating the identification and mitigation of bias for improved fairness, and supporting scientific discovery by providing deeper insights into model behavior.

2.A Additional Details on Simulations

In this section we provide additional details on the numerical experiments shown in Figures 2-5 in the main text. We first state the two data generating processes (DGPs) we used in the experiments. Afterwards, we will give details on the figures and specify which of the two DGPs has been used in each figure.

DGP 1: We consider the covariate distribution $P_X = \mathcal{N}(0, \Sigma)$ with covariance matrix $\Sigma = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}$ and the target function $m : \mathbb{R}^2 \longrightarrow \mathbb{R}$ is defined for all $x \in \mathbb{R}^2$ as

$$m(x) = x_1 + x_2 + 2x_1x_2.$$

We generate independent samples from (X, Y) by first sampling $X \sim P_X$ and then $Y \sim \mathcal{N}(m(X), 1)$.

DGP 2: We consider the covariate distribution $P_X = \mathcal{N}(0, \Sigma)$ with $\Sigma = 3 \cdot I_7 + \frac{3}{5} \cdot J_7$, where $I_7 \in \mathbb{R}^{7\times7}$ denotes the identity matrix and $J_7 \in \mathbb{R}^{7\times7}$ denotes the antidiagonal identity matrix (entries of ones going from lower left corner to upper right corner, rest being zero). The target function $m : \mathbb{R}^7 \longrightarrow \mathbb{R}$ is defined for all $x \in \mathbb{R}^7$ by

$$m(x) = 3\sin(x_1) + 2.5\cos(0.3x_2) + 1.12x_3 + \sin(x_4x_5) + 0.7x_6x_7.$$

We generate independent samples from (X, Y) by first sampling $X \sim P_X$ and then $Y \sim \mathcal{N}(m(X), 0.1)$. In Figure 2.3, this DGP is modified to have 6 additional covariates that are not used by the target function.

All numerical experiments were conducted using R-4.4.1 or Python-3.12 on a dedicated cluster with 2 Intel Xeon Gold 6302@2.1 GHz CPUs and 192 GB of memory. We modified the existing R package `glex` to compute the PD functions using `FastPD` and the path-dependent algorithm – which is due to Friedman (2001) but also reproduced as Algorithm 1 in Lundberg et al. (2020). The code of the experiments can be found in the attached Git repository, `codeforsimulations`. Finally, we used `FastPD-100` to emulate the SHAP values that would have been computed by `TreeSHAP-int` since they are equivalent.

2.A.1 Estimation Error Comparison - Figure 2.3

For this numerical experiment we generated iid datasets $\{(X^{(1)}, Y^{(1)}), \ldots, (X^{(n)}, Y^{(n)})\}$ over $B = 100$ repetitions with sample sizes $n = 500$ and $n = 5000$ from DGP 1. Multiple XGBoost models $(\hat{m}_n)$ were trained in each of the 100 repetitions with 5-fold cross-validation and their out-of-fold mean squared prediction error (MSPE) was computed. We ran the cross-validation with random search and 50 evaluations to tune the hyperparameters: `nrounds` $\in \{1, 2, \ldots, 1000\}$, `eta` $\in [0.01, 0.3]$ and `max_depth` $\in \{2, 3, \ldots, 6\}$. Following optimization, the best hyperparameter configuration was used to fit an XGBoost model on all n observations, and the

SHAP value ϕ_1 was estimated for all n observations using the different methods. The n generated samples were also used as background samples. The SHAP MSEs were then computed as $\frac{1}{n}\sum_{i=1}^{n}(\phi_1^{\hat{m}_n}(X^{(i)}) - \hat{\phi}_1^{\hat{m}_n}(X^{(i)}))^2$, where $\hat{\phi}_1^{\hat{m}_n}$ is the estimate of the SHAP value $\phi_1^{\hat{m}_n}$ for the target function $\hat{m}_n$ for each method.

Additional Simulations on Figure 2.3

We have furthermore conducted the same experiment with 8 covariates with pairwise correlation of 0 and 0.7, respectively. We notice that `TreeSHAP-path` performs noticeably worse in high-correlation scenarios, while our `FastPD` method is robust.

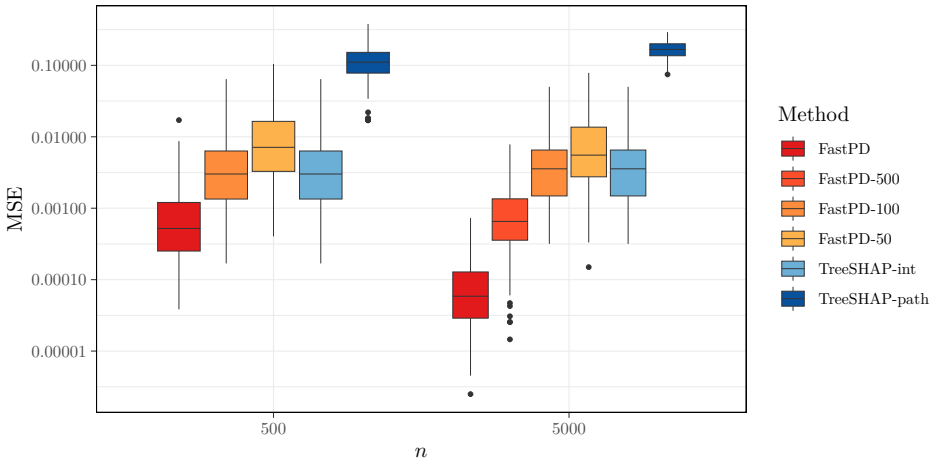


Figure 2.7: Reproduced Figure 2.3 with 8 covariates under the same target function as DGP 1, except that the covariates have a pairwise correlation of 0.7.

`TreeSHAP-path` can reliably estimate SHAP values when covariates are uncorrelated. However, its error remains higher than `FastPD` because it does not leverage all available samples. At each inner node, it conditions on the path taken by the evaluation point, significantly reducing the effective sample size. This effect is evident in the Figure 2.8 for $n = 500$.

2.A.2 Inconsistency of `TreeSHAP-path` - Figure 2.2

For this numerical experiment, we followed the same procedure as in Section 2.A.1. We selected the repetition where the MSE of `FastPD` matched its median MSE across all trials for the `FastPD` plot, and similarly, the repetition where `TreeSHAP-path` had median MSE for its plot.

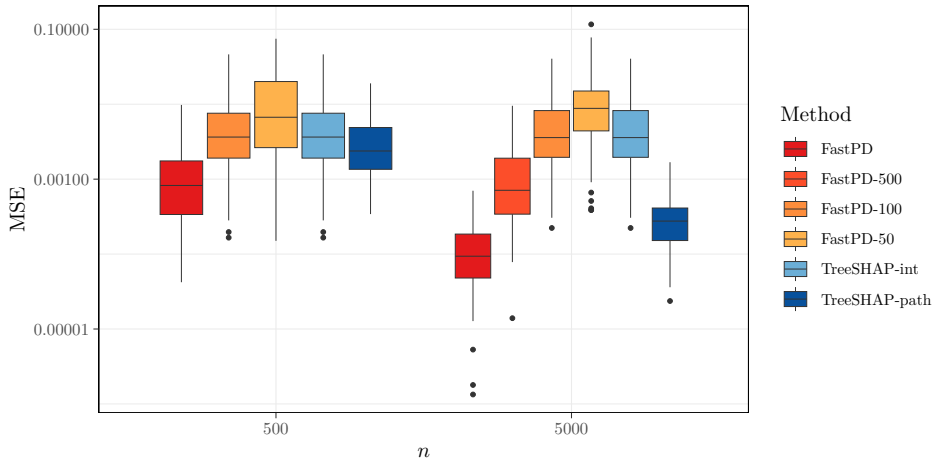


Figure 2.8: Reproduced Figure 2.3 with 8 covariates under the same target function as DGP 1, except that the covariates are uncorrelated.

Additional Simulations on Figure 2.2

We also conducted the same experiment under DGP 2 with pairwise correlations of $\{0, 0.1, 0.3, 0.7\}$. The results indicate that `TreeSHAP-path` produces biased estimates of Model SHAP, whereas `FastPD` does not.

2.A.3 Inconsistency of `Friedman-path` - Figure 2.5

For this numerical experiment we followed the same procedure as in Section 2.A.1. We selected the single repetition for which the MSE of the `FastPD` m_1-component corresponded to the median MSE across all trials. In order to reproduce the `Friedman-path` plot, we used the R package `glex` which provided the path-dependent functional decomposition that was computed using the algorithm by Friedman (2001). Furthermore, we have modified the package to implement our FastPD algorithm, which we then used to compute the functional decomposition for the `FastPD` plot.

Additional Simulations on Figure 2.5

We also conducted the same experiment under DGP 1 with pairwise correlations of 0.3 and 0.7. The median performing run is shown in Figure 2.10. We see that `FastPD` performs well in estimating the model functional component but by doing so it differs (and more so with increasing correlation) from the ground truth functional component.

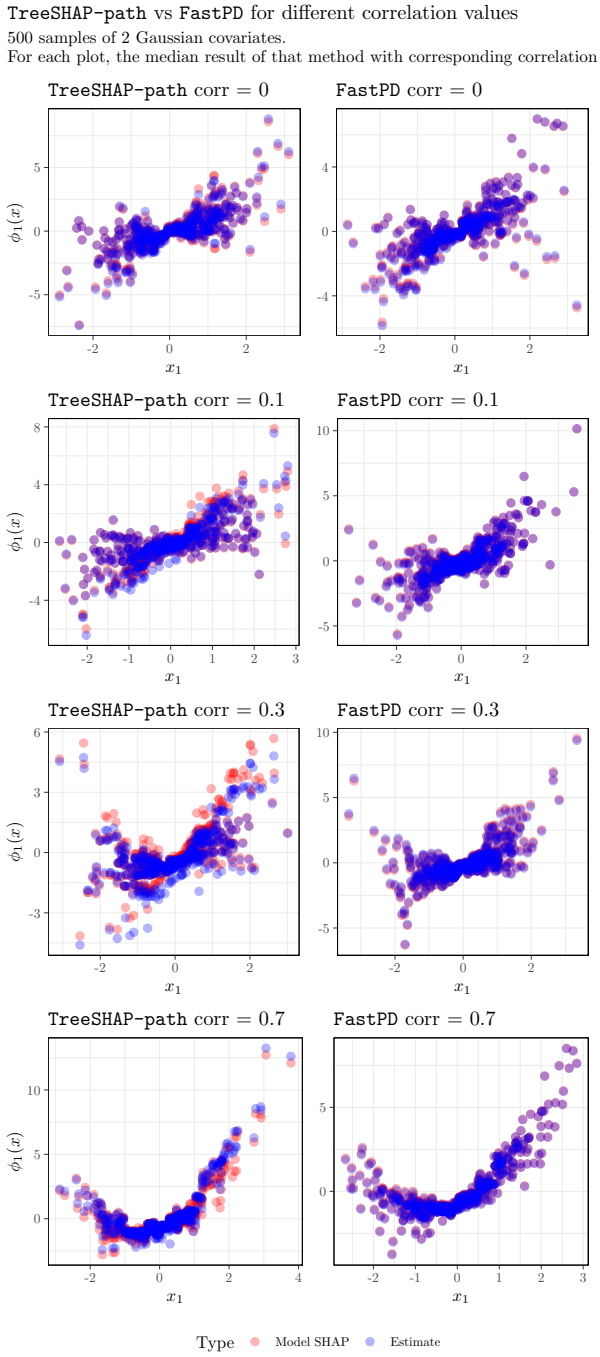


Figure 2.9: Reproduced Figure 2.2 with different pairwise correlations under DGP 2.

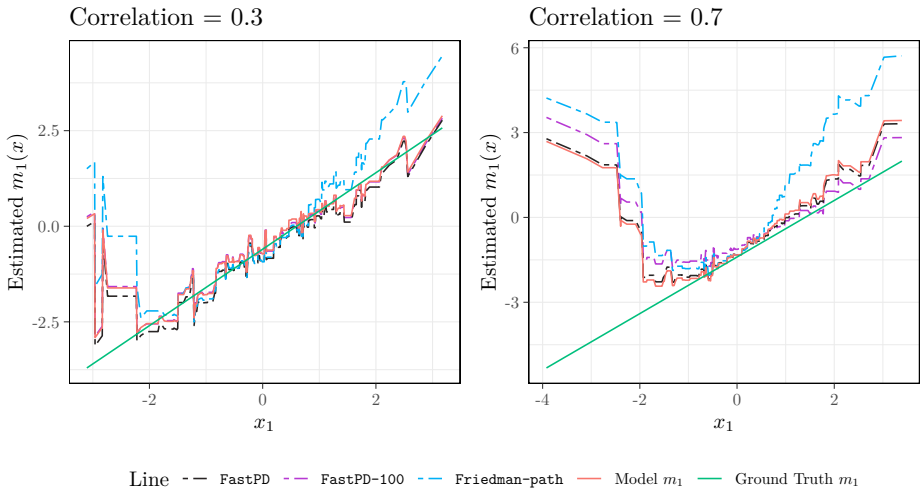


Figure 2.10: Plot of median `FastPD` and `Friedman-path` estimated component over 150 runs for correlations 0.3 and 0.7.

Figure 2.10 shows that in high-correlation settings the machine learning model can struggle to recover the true regression function; consequently, even though FastPD accurately estimates the model component, it does not target the underlying ground truth.

2.A.4 Runtime Comparison - Figure 2.4

For this numerical experiment, we generated a single dataset of size $n = 8000$ using DGP 2 and fitted an XGBoost model with 20 trees and max-depth of 5, with the rest being the default XGBoost parameters. We performed no hyperparameter tuning here as we only wish to examine the runtime when n is varied. Both the model $\hat{m}$ and the dataset was saved, we then evaluated the runtime as follows

1. For computing the functional components using `FastPD`: We implemented the algorithm in R (4.4.1) with Rcpp bindings to compute the all PD functions using `FastPD`.

2. For computing the SHAP values using `TreeSHAP-int`: We used Python-3.12 and modified the `shap` package[5] to compute the SHAP explanations for all features using arbitrary many background samples.

[5]GitHub: `https://github.com/shap/shap/`

3. For computing the SHAP values using Zern et al. (2023): We used Python 3.12 and the `pltreeshap` package [6] to compute the SHAP explanations for all features using arbitrary many background samples

4. For computing the SHAP values using `VanillaPD`: We wrote a simple script in Python 3.12 which repeatedly calls the predict function of XGBoost on the synthetic samples created from the background samples and the evaluation points, the predictions are then averaged to obtain the partial dependence functions.

For all $k \in \{1000, 2000, \ldots, 8000\}$, we took a subset $\mathcal{D}$ of the original dataset of size k and used it both as background and evaluation data (i.e., $n_b = n_f = k$). We then ran all methods 100 times to obtain SHAP values for n_e evaluation points using n_b background samples.

2.B Evaluation Algorithm

2.C Proofs

2.C.1 Proof of Proposition 2.3.1

Proof. We show (i) and (ii) via an example, let P_X be a distribution over $X = (X_1, X_2) \in \mathbb{R}^2$ such that

$$P_X(X = x) = \begin{cases} 500/2500 = 0.2 & \text{if } x = (0, 0), \\ 250/2500 = 0.1 & \text{if } x = (0, 0.4), \\ 250/2500 = 0.1 & \text{if } x = (0.7, 0), \\ 1500/2500 = 0.6 & \text{if } x = (0.7, 0.4), \\ 0 & \text{otherwise.} \end{cases}$$

Next, assume $n_b = 2500$ observations sampled from from P_X. There is a non-zero probability that $\mathcal{D}_{n_b}$ satisfies

$$\sum_{i=1}^{2500} \mathbb{1}(X^{(i)} = x) = \begin{cases} 500 & \text{if } x = (0, 0), \\ 250 & \text{if } x = (0, 0.4), \\ 250 & \text{if } x = (0.7, 0), \\ 1500 & \text{if } x = (0.7, 0.4) \\ 0 & \text{otherwise.} \end{cases}$$

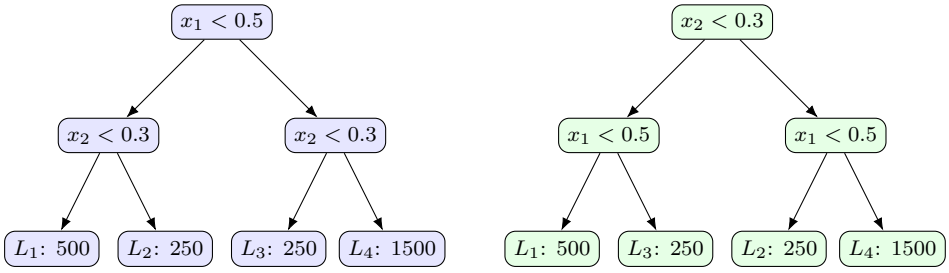
[6] GitHub: `https://github.com/schufa-innovationlab/pltreeshap/tree/main`

Algorithm 2.2 `FastPD` evaluation step to calculate $\hat{v}_S(x_S)$. To be applied after augmenting the tree as in Algorithm 2.1.

input Query point x, feature set S, tree structure, path features T, path data P
output $\hat{v}_S(x_S)$ – PD-function evaluated on x_S

$U \leftarrow S \cap \left(\bigcup_j T_j\right)$
if $\hat{v}_U(x_U)$ calculated before **then**
  return $\hat{v}_U(x_U)$
end if
function G(j)
  if j is leaf **then**
    $U_j \leftarrow U \cap T_j$
    Extract $D_{U_j}^{(j)}$ from P_j
    $\hat{P} \leftarrow \text{length}(D_{U_j}^{(}j))/n_b$
    return $v_j \cdot \hat{P}$
  end if
  if $d_j \in U$ **then**
    if $x_{d_j} \leq t_j$ **then**
      return G(l_j)
    else
      return G(r_j)
    end if
  else
    return G(l_j) + G(r_j)
  end if
end function
$\hat{v}_U(x_U) \leftarrow G(1)$
$\hat{v}_S(x_S) \leftarrow \hat{v}_U(x_U)$

We now consider two decision trees $\hat{m}^A$ and $\hat{m}^B$ as depicted in Figure 2.11. The leaves, going from left to right, are labeled L_1 to L_4 and are identical for both trees, implying that they are functionally equivalent, i.e., $\hat{m}^A(x) = \hat{m}^B(x)$ for all x.



(a) First Decision Tree $\hat{m}^A$ (b) Second Decision Tree $\hat{m}^B$

Figure 2.11: The two trees have the same leaves hence predict the same values, but their explanations differ when obtained via `TreeSHAP-path`. The number on each leaf is the number of observations landing in that leaf. The left branch is followed when the split condition is true.

We first prove (i) by showing that their SHAP values differ when they are computed using the path-dependent algorithm on $\mathcal{D}_{n_b}$. Indeed, let $x = (0.1, 0.2)$ be the observation to be explained and V_N be the value of leaf N. We follow the left branch if the split condition is satisfied. Assume that $V_1 = 10, V_2 = -5, V_3 = -5$ and $V_4 = 10$. In the following denote by $\tilde{v}^{\hat{m}^A}$ and $\tilde{v}^{\hat{m}^B}$ the estimates of the path-dependent PD functions (used in both `TreeSHAP-path` and `Friedman-path`). For the first tree, $\tilde{v}_S^{\hat{m}^A}(x_S)$, using $\mathcal{D}_{n_b}$ equals

$$\begin{aligned}
\tilde{v}_{\emptyset}^{\hat{m}^A} &= \frac{1}{2500}\left(500 \cdot V_1 + 250 \cdot V_2 + 250 \cdot V_3 + 1500 \cdot V_4\right) = 7, \\
\tilde{v}_1^{\hat{m}^A}(x_1) &= \frac{1}{750}(500 \cdot V_1 + 250 \cdot V_2) = 5, \\
\tilde{v}_2^{\hat{m}^A}(x_2) &= \frac{1}{2500}(750 \cdot V_1 + 1750 \cdot V_3) = -0.5, \\
\tilde{v}_{1,2}^{\hat{m}^A}(x_1, x_2) &= V_1 = 10.
\end{aligned}$$

For the second tree, $\tilde{v}_S^{\hat{m}^B}(x_S)$, using $\mathcal{D}_{n_b}$ equals

$$\begin{aligned}
\tilde{v}_{\emptyset}^{\hat{m}^B} &= \frac{1}{2500}\left(500 \cdot V_1 + 250 \cdot V_3 + 250 \cdot V_2 + 1500 \cdot V_4\right) = 7, \\
\tilde{v}_1^{\hat{m}^B}(x_1) &= \frac{1}{2500}(750 \cdot V_1 + 1750 \cdot V_2) = -0.5, \\
\tilde{v}_2^{\hat{m}^B}(x_2) &= \frac{1}{750}(500 \cdot V_1 + 250 \cdot V_3) = 5, \\
\tilde{v}_{1,2}^{\hat{m}^B}(x_1, x_2) &= V_1 = 10.
\end{aligned}$$

Finally, the `TreeSHAP-path` estimates of the SHAP value for feature x_1 in both trees are given as

$$\begin{aligned}
\hat{\phi}_1^{\hat{m}^A} &= \frac{1}{2}(\tilde{v}_{1,2}^{\hat{m}^A}(x_1, x_2) - \tilde{v}_2^{\hat{m}^A}(x_2) + \tilde{v}_1^{\hat{m}^A}(x_1) - \tilde{v}_{\emptyset}^{\hat{m}^A}) = 4.25, \\
\hat{\phi}_1^{\hat{m}^B} &= \frac{1}{2}(\tilde{v}_{1,2}^{\hat{m}^B}(x_1, x_2) - \tilde{v}_2^{\hat{m}^B}(x_2) + \tilde{v}_1^{\hat{m}^B}(x_1) - \tilde{v}_{\emptyset}^{\hat{m}^B}) = -1.25.
\end{aligned}$$

The computed SHAP values not only differ but have opposite signs! We observe that for $\hat{m}^A$, feature x_1 has a positive attribution, whereas the attribution is negative for $\hat{m}^B$. We can also compute the empirical PD functions for both trees as follows

$$\begin{aligned}
\hat{v}_\emptyset &= \frac{1}{2500}\left(500 \cdot V_1 + 250 \cdot V_2 + 250 \cdot V_3 + 1500 \cdot V_4\right) = 7, \\
\hat{v}_1(x_1) &= \frac{1}{2500}(750 \cdot V_1 + 1750 \cdot V_2) = -0.5, \\
\hat{v}_2(x_2) &= \frac{1}{2500}(750 \cdot V_1 + 1750 \cdot V_3) = -0.5, \\
\hat{v}_{1,2}(x_1, x_2) &= V_1 = 10.
\end{aligned}$$

And thus, the empirical SHAP estimate is given as

$$\hat{\phi}_1 = \frac{1}{2}(\hat{v}_{1,2}^{\hat{m}^A}(x_1, x_2) - \hat{v}_2^{\hat{m}^A}(x_2) + \hat{v}_1^{\hat{m}^A}(x_1) - \hat{v}_\emptyset^{\hat{m}^A}) = 1.5,$$

which is the same for both $\hat{m}^A$ and $\hat{m}^B$. Furthermore, by construction, the empirical SHAP estimate is equal to the population SHAP value, $\hat{\phi}_1 = \phi_1^{\hat{m}^A} = \phi_1^{\hat{m}^B}$.

We now show (ii). Let $\#L_N$ denote the number of observations that fall in leaf N. The path-dependent approximations of the PD functions for the first tree can be alternatively written as

$$\begin{aligned}
\tilde{v}_\emptyset^{\hat{m}^A} &= \frac{1}{n_b}\left(\#L_1 \cdot V_1 + \#L_2 \cdot V_2 + \#L_3 \cdot V_3 + \#L_4 \cdot V_4\right), \\
\tilde{v}_1^{\hat{m}^A}(x_1) &= \frac{1}{\#L_1 + \#L_2}(\#L_1 \cdot V_1 + \#L_2 \cdot V_2), \\
\tilde{v}_2^{\hat{m}^A}(x_2) &= \frac{1}{n_b}((\#L_1 + \#L_2) \cdot V_1 + (\#L_3 + \#L_4) \cdot V_3), \\
\tilde{v}_{1,2}^{\hat{m}^A}(x_1, x_2) &= V_1 = 10.
\end{aligned}$$

By the strong law of large numbers it holds that $\#L_N/n_b \xrightarrow{a.s.} P_X(X \in L_N)$ as $n_b \longrightarrow \infty$, so therefore $\tilde{v}_\emptyset^{\hat{m}^A} \xrightarrow{a.s.} 7$ and $\tilde{v}_2^{\hat{m}^A}(x_2) \xrightarrow{a.s.} -0.5$. However, since $(\#L_1 + \#L_2)/n_b \xrightarrow{a.s.} P_X(X \in L_1) + P_X(X \in L_2)$ we have the following

$$\begin{aligned}
\frac{\#L_1}{\#L_1 + \#L_2} &\xrightarrow{a.s.} \frac{P_X(X \in L_1)}{P_X(X \in L_1) + P_X(X \in L_2)} = 0.2/0.3 = 500/750, \\
\frac{\#L_2}{\#L_1 + \#L_2} &\xrightarrow{a.s.} \frac{P_X(X \in L_2)}{P_X(X \in L_1) + P_X(X \in L_2)} = 0.1/0.3 = 250/750.
\end{aligned}$$

Hence $\tilde{v}_1^{\hat{m}^A}(x_1) \xrightarrow{a.s.} 5$, implying that $\hat{\phi}_1^{\hat{m}^A} \xrightarrow{a.s.} 4.25$, which is not the same as the population SHAP, which was 1.5. □

2.C.2 Proof of Proposition 2.3.2

Proof. First, observe that the PD function $v_S^m(x_S) = \mathbb{E}_{P_X}[m(x_S, X_{\bar{S}})]$ of m exists with respect to any $S \subseteq [d]$ since m is bounded.

For the first part of the statement, we fix $S \subseteq [d]$. Since `FastPD` exactly evaluates the empirical PD function, it holds for all $x \in \mathcal{X}$ that

$$\hat{v}^m_{S,n_b}(x_S) = \frac{1}{n_b} \sum_{i=1}^{n_b} m(x_S, X^{(i)}_{\bar{S}}).$$

Therefore, since m is bounded the strong law of large numbers implies that

$$\hat{v}^m_{S,n_b} \xrightarrow{a.s.} v^m_S(x_S) \qquad \text{for } n_b \longrightarrow \infty.$$

For the second part of the statement, again fix $S \subseteq [d]$ and let $\hat{m}_{n_b}$ be a uniformly consistent estimate of m trained on $\mathcal{D}_{n_b}$ observations. Then by applying the triangle inequality it readily follows for all $x \in \mathcal{X}$ that

$$\begin{aligned}
\left| v^{\hat{m}_{n_b}}_{S,n_b}(x_S) - v^m_S(x_S) \right| &= \left| \frac{1}{n_b} \sum_{i=1}^{n_b} \hat{m}_{n_b}(x_S, X^{(i)}_{\bar{S}}) - \mathbb{E}_{P_X}[m(x_S, X_{\bar{S}})] \right| \\
&= \left| \frac{1}{n_b} \sum_{i=1}^{n_b} \hat{m}_{n_b}(x_S, X^{(i)}_{\bar{S}}) - \frac{1}{n_b} \sum_{i=1}^{n_b} m(x_S, X^{(i)}_{\bar{S}}) \right. \\
&\quad \left. + \frac{1}{n_b} \sum_{i=1}^{n_b} m(x_S, X^{(i)}_{\bar{S}}) - \mathbb{E}_{P_X}[m(x_S, X_{\bar{S}})] \right| \\
&\leq \frac{1}{n_b} \sum_{i=1}^{n_b} \left| \hat{m}_{n_b}(x_S, X^{(i)}_{\bar{S}}) - m(x_S, X^{(i)}_{\bar{S}}) \right| \\
&\quad + \left| \frac{1}{n_b} \sum_{i=1}^{n_b} m(x_S, X^{(i)}_{\bar{S}}) - \mathbb{E}_{P_X}[m(x_S, X_{\bar{S}})] \right| \\
&\leq \frac{1}{n_b} \sum_{i=1}^{n_b} \sup_x \left| \hat{m}_{n_b}(x) - m(x) \right| \\
&\quad + \left| \frac{1}{n_b} \sum_{i=1}^{n_b} m(x_S, X^{(i)}_{\bar{S}}) - \mathbb{E}_{P_X}[m(x_S, X_{\bar{S}})] \right| \\
&\xrightarrow{a.s.} 0,
\end{aligned}$$

where the convergence follows from using uniform consistency of $\hat{m}_{n_b}$ and the consistency of the empirical PD function which follows from the strong law of large numbers as above. □

2.D Experiments on Real Data

2.D.1 Adult Dataset - Figure 2.6

To reproduce Figure 2.6, we used the `Adult` dataset where we first removed all observations with missing values and applied ordinal encoding to every categorical feature. We then performed a randomized grid search (25 evaluations) with 5-fold cross-validation to tune the following XGBoost hyperparameters:

$$\begin{aligned}
&\texttt{max_depth} \in \{3,4,5,6,7\}, \quad \texttt{eta} \in \{0.01, 0.05, 0.1\},\\
&\texttt{nrounds} \in \{100, 200, 300\}, \quad \texttt{min_child_weight} \in \{1,3,5\},\\
&\texttt{subsample} \in \{0.6, 0.8, 1.0\}, \quad \texttt{colsample_bytree} \in \{0.6, 0.8, 1.0\}.
\end{aligned}$$

The best-performing model had the following hyperparameters:

$$\begin{aligned}
\texttt{subsample} &= 0.8, & \texttt{min_child_weight} &= 1,\\
\texttt{max_depth} &= 6, & \texttt{eta} &= 0.05,\\
\texttt{colsample_bytree} &= 0.6, & \texttt{nrounds} &= 100
\end{aligned}$$

2.D.2 FastPD vs Path-dependent (Regression)

We summarize the results of the experiments comparing `FastPD` with path-dependent methods in the regression setting below. We used the OpenML-CTR23 Regression Task Collection, which contains 35 varied regression datasets. On each dataset, we ran 5-fold cross-validation, tuning the following hyperparameters for the XGBoost model over 100 randomly sampled settings:

$$\begin{aligned}
\texttt{max_depth} &\in [1,5], & \texttt{eta} &\in [0, 0.5],\\
\texttt{nrounds} &\in [10, 200], & \texttt{colsample_bytree} &\in [0.5, 1],\\
\texttt{subsample} &\in [0.5, 1].
\end{aligned}$$

The best-performing model was then selected based on the cross-validated MSE, and used by `FastPD`, `FastPD (50)`, `FastPD (100)` and the path-dependent method (`Friedman--path`) to compute the functional components (see Figure 2.5). We then applied the variable-importance measure for each component S as

$$\text{Importance}\big(\widehat{m}_S\big) = \frac{1}{n}\sum_{i=1}^{n}\big|\widehat{m}_S\big(X_S^{(i)}\big)\big|.$$

This enables us to rank the importance of each component. We present the importances of all components that were ranked in the top five by *any* method for each dataset. Using `FastPD` as the reference method, the relative differences

$$\frac{\text{Importance(other)} - \text{Importance(}\texttt{FastPD}\text{)}}{\text{Importance(}\texttt{FastPD}\text{)}}$$

of each method with respect to `FastPD` are reported in parentheses. Finally, for each dataset we report in parentheses the number of observations n, the feature dimension p and the maximum depth of the XGBoost model D. Note that `FastPD` and the path-dependent method are trivially equivalent for those datasets where the tuned XGBoost model had $D = 1$.

Dataset	Variable/Interaction	FastPD (Reference)	FastPD (50)	FastPD (100)	Path-dependent
Moneyball (n: 1232, p: 15, D: 2)	Intercept	**713.8**	694.4 (−2.7%)	715.5 (+0.2%)	713.8 (0%)
	SLG	**35.96**	35.91 (−0.1%)	36.12 (+0.4%)	37.11 (+3.2%)
	OBP	**28.36**	28.45 (+0.3%)	28.2 (−0.6%)	28.5 (+0.5%)
	W	**15.26**	15.94 (+4.5%)	15.52 (+1.7%)	13.03 (−14.6%)
	RA	**10.95**	11.05 (+0.9%)	11.4 (+4.1%)	10.51 (−4%)
QSAR_fish_toxicity (n: 908, p: 7, D: 5)	Intercept	**4.066**	4.143 (+1.9%)	4.125 (+1.5%)	4.066 (0%)
	MLOGP	**0.4468**	0.4085 (−8.6%)	0.4514 (+1%)	0.5795 (+29.7%)
	SM1_Dz	**0.4279**	0.4339 (+1.4%)	0.4323 (+1%)	0.4099 (−4.2%)
	GATS1i	**0.2743**	0.2865 (+4.4%)	0.2869 (+4.6%)	0.2568 (−6.4%)
	CIC0	**0.1907**	0.1681 (−11.9%)	0.2047 (+7.3%)	0.1703 (−10.7%)
abalone (n: 4177, p: 9, D: 5)	Intercept	**9.934**	10.08 (+1.5%)	10.17 (+2.4%)	9.934 (0%)
	shell_weight	**1.388**	1.33 (−4.2%)	1.358 (−2.2%)	1.206 (−13.1%)
	shucked_weight	**1.26**	1.349 (+7.1%)	1.311 (+4%)	0.8631 (−31.5%)
	shucked_weight:whole_weight	**0.8577**	0.8526 (−0.6%)	0.8585 (+0.1%)	0.4454 (−48.1%)
	whole_weight	**0.7164**	0.7108 (−0.8%)	0.6996 (−2.3%)	0.4368 (−39%)
airfoil_self_noise (n: 1503, p: 6, D: 5)	Intercept	**124.8**	124.9 (+0.1%)	125.1 (+0.2%)	124.8 (0%)
	frequency	**3.335**	3.389 (+1.6%)	3.554 (+6.6%)	2.734 (−18%)
	displacement_thickness:frequency	**2.985**	2.841 (−4.8%)	2.719 (−8.9%)	2.521 (−15.5%)
	chord_length:frequency	**1.837**	1.963 (+6.9%)	1.93 (+5.1%)	1.821 (−0.9%)
	chord_length	**1.678**	1.637 (−2.4%)	1.998 (+19.1%)	1.543 (−8%)
auction_verification (n: 2043, p: 8, D: 5)	Intercept	**7337**	7932 (+8.1%)	8331 (+13.5%)	7337 (0%)
	process.b1.capacity	**4712**	5548 (+17.7%)	4625 (−1.8%)	4571 (−3%)
	property.product.2	**3318**	3479 (+4.9%)	3374 (+1.7%)	4126 (+24.4%)
	process.b1.capacity:property.product.2	**2329**	2642 (+13.4%)	2176 (−6.6%)	2627 (+12.8%)
	property.product.6	**2219**	2029 (−8.6%)	1960 (−11.7%)	1667 (−24.9%)
brazilian_houses (n: 10692, p: 10, D: 1)	Intercept	**5487**	5821 (+6.1%)	4867 (−11.3%)	5487 (0%)
	hoa	**1718**	1791 (+4.2%)	1608 (−6.4%)	1718 (0%)
	area	**1320**	1364 (+3.3%)	1282 (−2.9%)	1320 (0%)
	bathroom	**620.8**	576.2 (−7.2%)	595 (−4.2%)	620.8 (0%)
	furniture.not.furnished	**312.1**	336.5 (+7.8%)	314.8 (+0.9%)	312.1 (0%)
california_housing (n: 20640, p: 9, D: 5)	Intercept	**206800**	207700 (+0.4%)	218700 (+5.8%)	206800 (0%)
	latitude	**93530**	90120 (−3.6%)	90150 (−3.6%)	53530 (−42.8%)
	latitude:longitude	**82130**	85870 (+4.6%)	83530 (+1.7%)	48730 (−40.7%)

	longitude	**75310**	77420 (+2.8%)	80700 (+7.2%)	52040 (−30.9%)
	medianIncome	**42210**	40590 (−3.8%)	43190 (+2.3%)	48430 (+14.7%)
cars (n: 804, p: 18, D: 5)	Intercept	**21330**	19870 (−6.8%)	23490 (+10.1%)	21330 (0%)
	Cylinder	**2701**	2688 (−0.5%)	3154 (+16.8%)	2869 (+6.2%)
	Cadillac	**2696**	1775 (−34.2%)	3285 (+21.8%)	2604 (−3.4%)
	Saab	**2271**	2251 (−0.9%)	2231 (−1.8%)	2406 (+5.9%)
	Mileage	**1166**	1152 (−1.2%)	1213 (+4%)	1164 (−0.2%)
concrete_compressive_strength (n: 1030, p: 9, D: 4)	Intercept	**35.82**	38.19 (+6.6%)	36.2 (+1.1%)	35.82 (0%)
	age	**8.087**	8.365 (+3.4%)	7.96 (−1.6%)	7.641 (−5.5%)
	cement	**6.263**	6.089 (−2.8%)	6.238 (−0.4%)	5.942 (−5.1%)
	water	**4.093**	4.145 (+1.3%)	4.104 (+0.3%)	3.914 (−4.4%)
	blast_furnace_slag	**2.702**	2.623 (−2.9%)	2.722 (+0.7%)	2.429 (−10.1%)
cps88wages (n: 28155, p: 7, D: 4)	Intercept	**604**	546.8 (−9.5%)	628 (+4%)	604 (0%)
	education	**122.3**	109 (−10.9%)	128.4 (+5%)	125.5 (+2.6%)
	experience	**119.3**	121.1 (+1.5%)	123.8 (+3.8%)	114.5 (−4%)
	parttime.yes	**45.72**	65.15 (+42.5%)	43.05 (−5.8%)	46.77 (+2.3%)
	smsa.yes	**36.99**	33.19 (−10.3%)	39.36 (+6.4%)	37.16 (+0.5%)
cpu_activity (n: 8192, p: 22, D: 5)	Intercept	**83.97**	83.81 (−0.2%)	84.37 (+0.5%)	83.97 (0%)
	freeswap	**5.133**	4.076 (−20.6%)	4.725 (−7.9%)	5.126 (−0.1%)
	vflt	**2.359**	2.58 (+9.4%)	2.325 (−1.4%)	3.304 (+40.1%)
	scall	**1.597**	1.808 (+13.2%)	1.708 (+7%)	1.687 (+5.6%)
	pflt	**0.8853**	0.9213 (+4.1%)	0.959 (+8.3%)	0.8976 (+1.4%)
diamonds (n: 53940, p: 10, D: 5)	Intercept	**3933**	3949 (+0.4%)	4278 (+8.8%)	3933 (0%)
	carat	**2329**	2322 (−0.3%)	2400 (+3%)	2250 (−3.4%)
	y	**1242**	1351 (+8.8%)	1289 (+3.8%)	1616 (+30.1%)
	clarity.VS2	**407.1**	418.2 (+2.7%)	374.4 (−8%)	300.5 (−26.2%)
	clarity.VS1	**363.3**	303.3 (−16.5%)	294 (−19.1%)	274.3 (−24.5%)
	x:y	**308.6**	363.6 (+17.8%)	421.5 (+36.6%)	129.2 (−58.1%)
	carat:y	**293.8**	318.2 (+8.3%)	308.1 (+4.9%)	644.9 (+119.5%)
	clarity.VVS1	**243.8**	410.6 (+68.4%)	248.8 (+2.1%)	181 (−25.8%)
energy_efficiency (n: 768, p: 9, D: 5)	Intercept	**22.31**	21.34 (−4.3%)	22.67 (+1.6%)	22.31 (0%)
	overall_height	**10.07**	10.05 (−0.2%)	10.56 (+4.9%)	8.534 (−15.3%)
	glazing_area	**2.17**	2.326 (+7.2%)	2.044 (−5.8%)	2.164 (−0.3%)
	overall_height:relative_compactness	**1.592**	1.585 (−0.4%)	1.386 (−12.9%)	1.6 (+0.5%)

	relative_compactness	**1.495**	1.494 (−0.1%)	1.152 (−22.9%)	1.561 (+4.4%)
fifa (n: 19178, p: 29, D: 3)	Intercept	**9020**	15910 (+76.4%)	9309 (+3.2%)	9020 (0%)
	overall	**7661**	11650 (+52.1%)	7544 (−1.5%)	7775 (+1.5%)
	skill_ball_control	**884.3**	955.9 (+8.1%)	1112 (+25.7%)	899.5 (+1.7%)
	nationality_name.England	**522.7**	968.5 (+85.3%)	497.6 (−4.8%)	494.1 (−5.5%)
	attacking_heading_accuracy	**426.7**	448.7 (+5.2%)	377.7 (−11.5%)	399.3 (−6.4%)
	skill_dribbling	**250.7**	441.7 (+76.2%)	412.6 (+64.6%)	229.7 (−8.4%)
	defending_standing_tackle	**237.1**	710.9 (+199.8%)	334.9 (+41.2%)	262.1 (+10.5%)
forest_fires (n: 517, p: 13, D: 2)	Intercept	**3.536**	5.088 (+43.9%)	4.289 (+21.3%)	3.536 (0%)
	temp	**1.184**	2.376 (+100.7%)	1.748 (+47.6%)	1.218 (+2.9%)
	DMC	**0.3103**	0.3302 (+6.4%)	0.3246 (+4.6%)	0.2826 (−8.9%)
	day.sat:temp	**0.1736**	0.1705 (−1.8%)	0.1967 (+13.3%)	0.1701 (−2%)
	Y	**0.1575**	0.1124 (−28.6%)	0.1384 (−12.1%)	0.1575 (0%)
	day.sat	**0.1539**	0.1403 (−8.8%)	0.1786 (+16%)	0.1499 (−2.6%)
fps_benchmark (n: 24624, p: 44, D: 4)	Intercept	**123.6**	113 (−8.6%)	121.5 (−1.7%)	123.6 (0%)
	GameSetting.max	**15.59**	15.16 (−2.8%)	15.67 (+0.5%)	15.59 (0%)
	GameName.battlefield4	**9.474**	11.48 (+21.2%)	8.28 (−12.6%)	9.267 (−2.2%)
	GameName.worldOfTanks	**9.219**	6.357 (−31%)	5.871 (−36.3%)	8.987 (−2.5%)
	GameName.totalWar3Kingdoms	**8.719**	10.05 (+15.3%)	7.265 (−16.7%)	8.491 (−2.6%)
	GameName.grandTheftAuto5	**8.142**	6.011 (−26.2%)	11.18 (+37.3%)	7.957 (−2.3%)
	GpuNumberOfTransistors	**7.022**	6.575 (−6.4%)	7.271 (+3.5%)	8.611 (+22.6%)
	GameName.apexLegends	**6.837**	16.53 (+141.8%)	9.649 (+41.1%)	6.594 (−3.6%)
	GameName.destiny2	**5.488**	11.93 (+117.4%)	6.69 (+21.9%)	5.311 (−3.2%)
geographical_origin_of_music (n: 1059, p: 117, D: 4)	Intercept	**26.68**	26.11 (−2.1%)	29.12 (+9.1%)	26.68 (0%)
	V32	**2.66**	2.762 (+3.8%)	2.486 (−6.5%)	2.921 (+9.8%)
	V92	**2.016**	2.229 (+10.6%)	2.013 (−0.1%)	1.627 (−19.3%)
	V4	**1.53**	1.787 (+16.8%)	1.46 (−4.6%)	1.288 (−15.8%)
	V90	**1.392**	1.287 (−7.5%)	1.311 (−5.8%)	1.469 (+5.5%)
	V91	**1.173**	1.358 (+15.8%)	0.9099 (−22.4%)	1.141 (−2.7%)
grid_stability (n: 10000, p: 13, D: 5)	Intercept	**0.01574**	0.01094 (−30.5%)	0.01687 (+7.2%)	0.01574 (0%)
	tau2	**0.01023**	0.01183 (+15.6%)	0.009302 (−9.1%)	0.01034 (+1.1%)
	tau4	**0.01021**	0.008786 (−13.9%)	0.008845 (−13.4%)	0.01017 (−0.4%)
	tau3	**0.01014**	0.008657 (−14.6%)	0.0108 (+6.5%)	0.01014 (0%)
	tau1	**0.01008**	0.01042 (+3.4%)	0.01186 (+17.7%)	0.01015 (+0.7%)

	g3	**0.009318**	0.009932 (+6.6%)	0.009094 (−2.4%)	0.009422 (+1.1%)
	g2	**0.009225**	0.00834 (−9.6%)	0.01086 (+17.7%)	0.009254 (+0.3%)
	g4	**0.009084**	0.009958 (+9.6%)	0.009448 (+4%)	0.009025 (−0.6%)
health_insurance (n: 22272, p: 12, D: 5)	Intercept	**25.5**	24.86 (−2.5%)	27.17 (+6.5%)	25.5 (0%)
	whi.yes	**8.041**	8.252 (+2.6%)	7.856 (−2.3%)	8.687 (+8%)
	experience	**3.469**	3.612 (+4.1%)	3.239 (−6.6%)	3.092 (−10.9%)
	kidslt6	**2.557**	2.594 (+1.4%)	2.497 (−2.3%)	2.142 (−16.2%)
	husby	**1.637**	1.711 (+4.5%)	1.434 (−12.4%)	1.636 (−0.1%)
kin8nm (n: 8192, p: 9, D: 5)	Intercept	**0.714**	0.7176 (+0.5%)	0.7396 (+3.6%)	0.714 (0%)
	theta3	**0.1264**	0.1068 (−15.5%)	0.1331 (+5.3%)	0.1253 (−0.9%)
	theta5	**0.05947**	0.06424 (+8%)	0.05806 (−2.4%)	0.05835 (−1.9%)
	theta4:theta7	**0.04003**	0.04405 (+10%)	0.03762 (−6%)	0.0403 (+0.7%)
	theta4:theta6	**0.03859**	0.04127 (+6.9%)	0.03474 (−10%)	0.03879 (+0.5%)
	theta6	**0.03624**	0.02494 (−31.2%)	0.04615 (+27.3%)	0.03586 (−1%)
	theta7	**0.03544**	0.03077 (−13.2%)	0.04927 (+39%)	0.03534 (−0.3%)
kings_county (n: 21613, p: 22, D: 5)	Intercept	**540100**	548000 (+1.5%)	586000 (+8.5%)	540100 (0%)
	lat	**106400**	104800 (−1.5%)	109600 (+3%)	105700 (−0.7%)
	sqft_living	**64580**	65630 (+1.6%)	67860 (+5.1%)	74220 (+14.9%)
	grade	**61040**	59060 (−3.2%)	70030 (+14.7%)	75760 (+24.1%)
	long	**28700**	27900 (−2.8%)	35610 (+24.1%)	27830 (−3%)
miami_housing (n: 13932, p: 16, D: 4)	Intercept	**399900**	436400 (+9.1%)	410000 (+2.5%)	399900 (0%)
	TOT_LVG_AREA	**66800**	69390 (+3.9%)	67920 (+1.7%)	82580 (+23.6%)
	OCEAN_DIST	**39470**	42860 (+8.6%)	36060 (−8.6%)	48600 (+23.1%)
	CNTR_DIST	**37680**	49020 (+30.1%)	42160 (+11.9%)	39730 (+5.4%)
	SUBCNTR_DI	**35880**	37150 (+3.5%)	36230 (+1%)	34710 (−3.3%)
	LONGITUDE	**34240**	37200 (+8.6%)	32940 (−3.8%)	25080 (−26.8%)
naval_propulsion_plant (n: 11934, p: 15, D: 5)	Intercept	**0.975**	0.9752 (0%)	0.9745 (−0.1%)	0.975 (0%)
	hp_turbine_exit_pressure	**0.01705**	0.01745 (+2.3%)	0.01698 (−0.4%)	0.007992 (−53.1%)
	gt_compressor_outlet_air_temperature	**0.01571**	0.01587 (+1%)	0.0157 (−0.1%)	0.01388 (−11.6%)
	gas_turbine_shaft_torque	**0.005027**	0.004993 (−0.7%)	0.005038 (+0.2%)	0.003613 (−28.1%)
	gas_turbine_exhaust_gas_pressure	**0.004985**	0.005062 (+1.5%)	0.005043 (+1.2%)	0.003119 (−37.4%)
	gt_compressor_outlet_air_temperature:hp_turbine_exit_pressure	**0.003217**	0.003212 (−0.2%)	0.003262 (+1.4%)	0.004906 (+52.5%)
physiochemical_protein (n: 45730, p: 10, D: 5)	Intercept	**7.749**	6.979 (−9.9%)	7.72 (−0.4%)	7.749 (0%)
	F6	**5.889**	5.986 (+1.6%)	5.691 (−3.4%)	3.806 (−35.4%)

	F1:F6	**5.288**	5.187 (−1.9%)	5.25 (−0.7%)	2.762 (−47.8%)
	F1	**5.012**	4.725 (−5.7%)	5.066 (+1.1%)	1.384 (−72.4%)
	F6:F7	**3.202**	3.034 (−5.2%)	3.009 (−6%)	1.483 (−53.7%)
pumadyn32nh (n: 8192, p: 33, D: 5)	tau4:theta5	**0.0183**	0.01843 (+0.7%)	0.01848 (+1%)	0.0183 (0%)
	tau4	**0.01223**	0.01181 (−3.4%)	0.01267 (+3.6%)	0.01229 (+0.5%)
	theta5	**0.0009914**	0.002607 (+163%)	0.003528 (+255.9%)	0.001023 (+3.2%)
	theta3	**0.0006446**	0.0005413 (−16%)	0.0005821 (−9.7%)	0.0006446 (0%)
	da5	**0.0005979**	0.0006811 (+13.9%)	0.0005471 (−8.5%)	0.0005952 (−0.5%)
	tau1	**0.0003974**	0.0003712 (−6.6%)	0.0005881 (+48%)	0.0003911 (−1.6%)
	Intercept	**-0.0003658**	−0.002458 (+572%)	−0.0001511 (−58.7%)	−0.0003658 (0%)
red_wine (n: 1599, p: 12, D: 5)	Intercept	**5.636**	5.675 (+0.7%)	5.635 (0%)	5.636 (0%)
	alcohol	**0.2392**	0.2238 (−6.4%)	0.2418 (+1.1%)	0.2608 (+9%)
	sulphates	**0.1847**	0.2005 (+8.6%)	0.1901 (+2.9%)	0.1821 (−1.4%)
	volatile_acidity	**0.1125**	0.1326 (+17.9%)	0.1058 (−6%)	0.1349 (+19.9%)
	total_sulfur_dioxide	**0.1119**	0.1233 (+10.2%)	0.1221 (+9.1%)	0.09211 (−17.7%)
sarcos (n: 48933, p: 22, D: 5)	V15	**14.41**	14.24 (−1.2%)	14.46 (+0.3%)	10.4 (−27.8%)
	Intercept	**13.66**	14.2 (+4%)	14.06 (+2.9%)	13.66 (0%)
	V1	**4.656**	4.739 (+1.8%)	4.823 (+3.6%)	3.965 (−14.8%)
	V4	**4.246**	3.886 (−8.5%)	4.125 (−2.8%)	4.372 (+3%)
	V18	**3.735**	3.858 (+3.3%)	3.556 (−4.8%)	3.261 (−12.7%)
socmob (n: 1156, p: 6, D: 3)	counts_for_sons_first_occupation	**18.73**	15.01 (−19.9%)	19.92 (+6.4%)	18.32 (−2.2%)
	Intercept	**18.21**	10.91 (−40.1%)	20.6 (+13.1%)	18.21 (0%)
	counts_for_sons_first_occupation:sons_occupation.Manager	**3.427**	2.298 (−32.9%)	3.326 (−2.9%)	3.207 (−6.4%)
	sons_occupation.Manager	**2.875**	1.611 (−44%)	3.047 (+6%)	2.704 (−5.9%)
	race.white	**2.812**	2.402 (−14.6%)	3.054 (+8.6%)	2.812 (0%)
	family_structure.nonintact	**2.621**	2.557 (−2.4%)	2.712 (+3.5%)	2.544 (−2.9%)
	family_structure.nonintact:race.white	**2.254**	2.618 (+16.1%)	2.441 (+8.3%)	2.254 (0%)
solar_flare (n: 1066, p: 11, D: 1)	Intercept	**0.3031**	0.3303 (+9%)	0.2313 (−23.7%)	0.3031 (0%)
	class.E	**0.1138**	0.1201 (+5.5%)	0.09706 (−14.7%)	0.1138 (0%)
	spot_distribution.I	**0.07133**	0.0777 (+8.9%)	0.05889 (−17.4%)	0.07133 (0%)
	activity.2	**0.05658**	0.0545 (−3.7%)	0.05299 (−6.3%)	0.05658 (0%)
	largest_spot_size.K	**0.03674**	0.04359 (+18.6%)	0.02733 (−25.6%)	0.03674 (0%)
space_ga (n: 3107, p: 7, D: 4)	houses:pop	**0.3214**	0.318 (−1.1%)	0.3215 (0%)	0.1294 (−59.7%)
	pop	**0.3201**	0.3209 (+0.2%)	0.3205 (+0.1%)	0.1387 (−56.7%)

	houses	**0.2858**	0.3025 (+5.8%)	0.2819 (−1.4%)	0.1585 (−44.5%)
	income	**0.08183**	0.07806 (−4.6%)	0.07767 (−5.1%)	0.04229 (−48.3%)
	ycoord	**0.07958**	0.08196 (+3%)	0.07674 (−3.6%)	0.07708 (−3.1%)
	education	**0.07759**	0.07654 (−1.4%)	0.08197 (+5.6%)	0.05881 (−24.2%)
	Intercept	**-0.5762**	−0.599 (+4%)	−0.5409 (−6.1%)	−0.5762 (0%)
student_performance_por (n: 649, p: 31, D: 3)	Intercept	**11.9**	12.21 (+2.6%)	11.57 (−2.8%)	11.9 (0%)
	failures	**0.6817**	0.6963 (+2.1%)	0.7445 (+9.2%)	0.7618 (+11.8%)
	school.MS	**0.4385**	0.4249 (−3.1%)	0.4602 (+4.9%)	0.4582 (+4.5%)
	higher.yes	**0.2677**	0.2676 (0%)	0.2621 (−2.1%)	0.2744 (+2.5%)
	goout	**0.2162**	0.2076 (−4%)	0.2396 (+10.8%)	0.2112 (−2.3%)
	Fedu	**0.2105**	0.195 (−7.4%)	0.2354 (+11.8%)	0.2159 (+2.6%)
	studytime	**0.2069**	0.2611 (+26.2%)	0.2139 (+3.4%)	0.2157 (+4.3%)
superconductivity (n: 21263, p: 82, D: 5)	Intercept	**34.41**	37.6 (+9.3%)	37.28 (+8.3%)	34.41 (0%)
	range_ThermalConductivity	**8.706**	10.24 (+17.6%)	9.074 (+4.2%)	12.6 (+44.7%)
	range_atomic_radius	**5.834**	6.994 (+19.9%)	6.612 (+13.3%)	10.07 (+72.6%)
	wtd_gmean_Valence	**5.257**	5.338 (+1.5%)	5.441 (+3.5%)	4.771 (−9.2%)
	range_atomic_radius:wtd_gmean_Valence	**3.253**	3.765 (+15.7%)	3.044 (−6.4%)	2.433 (−25.2%)
	wtd_mean_ThermalConductivity	**2.174**	1.663 (−23.5%)	2.248 (+3.4%)	3.025 (+39.1%)
video_transcoding (n: 68784, p: 19, D: 5)	Intercept	**9.996**	13.3 (+33.1%)	8.713 (−12.8%)	9.996 (0%)
	o_height	**7.135**	8.856 (+24.1%)	7.088 (−0.7%)	7.133 (0%)
	o_codec.h264	**6.294**	9.064 (+44%)	5.311 (−15.6%)	5.611 (−10.9%)
	o_codec.h264:o_height	**5.193**	6.577 (+26.7%)	4.903 (−5.6%)	4.649 (−10.5%)
	o_bitrate:o_codec.h264	**2.71**	3.967 (+46.4%)	2.417 (−10.8%)	2.647 (−2.3%)
wave_energy (n: 72000, p: 49, D: 4)	Intercept	**3760000**	3754000 (−0.2%)	3740000 (−0.5%)	3760000 (0%)
	energy4	**23350**	25030 (+7.2%)	23670 (+1.4%)	23620 (+1.2%)
	energy3	**23100**	22790 (−1.3%)	22490 (−2.6%)	22170 (−4%)
	energy9	**23030**	23490 (+2%)	23520 (+2.1%)	22100 (−4%)
	energy7	**22980**	23420 (+1.9%)	23150 (+0.7%)	22150 (−3.6%)
	energy12	**22730**	22310 (−1.8%)	23030 (+1.3%)	22170 (−2.5%)
	energy15	**22650**	22340 (−1.4%)	22700 (+0.2%)	22640 (0%)
	energy10	**22640**	22440 (−0.9%)	21820 (−3.6%)	22570 (−0.3%)
	energy14	**22500**	22620 (+0.5%)	23320 (+3.6%)	21270 (−5.5%)
	energy5	**22370**	23210 (+3.8%)	23710 (+6%)	22060 (−1.4%)
white_wine (n: 4898, p: 12, D: 5)	Intercept	**5.878**	5.844 (−0.6%)	5.89 (+0.2%)	5.878 (0%)

alcohol	**0.2482**	0.2618 (+5.5%)	0.2396 (−3.5%)	0.2358 (−5%)
density	**0.1564**	0.1409 (−9.9%)	0.1628 (+4.1%)	0.1123 (−28.2%)
residual_sugar	**0.1461**	0.1617 (+10.7%)	0.1613 (+10.4%)	0.1179 (−19.3%)
volatile_acidity	**0.1275**	0.1271 (−0.3%)	0.1218 (−4.5%)	0.1296 (+1.6%)
free_sulfur_dioxide	**0.1137**	0.1274 (+12%)	0.1416 (+24.5%)	0.113 (−0.6%)

2.D.3 FastPD vs Path-dependent (Classification)

For the classification experiments, we utilized the OpenML-CC18 Task Collection. The experimental procedure, including 5-fold cross-validation and XGBoost hyperparameter tuning, mirrored that of the regression setting. However, datasets with over 500 features were excluded since they primarily were image-based datasets.

For multiclass problems (where the number of classes $K > 2$), the default approach of XGBoost involves training K separate binary classification models ($\hat{f}_k$), each outputting a raw score and distinguishing one class from the rest (one-vs-rest). The interpretability methods were then applied to the sum of these raw scores, $\hat{f} = \sum_{k=1}^{K} \hat{f}_k$.

Dataset	Variable/Interaction	FastPD (Reference)	FastPD (50)	FastPD (100)	Path-dependent
GesturePhaseSegmentationProcessed (n: 9873, p: 33, D: 5)	X26	**0.5959**	0.6367 (+6.8%)	0.616 (+3.4%)	0.5459 (−8.4%)
	X28	**0.5427**	0.6583 (+21.3%)	0.5467 (+0.7%)	0.5737 (+5.7%)
	X5	**0.3939**	0.5222 (+32.6%)	0.5049 (+28.2%)	0.3717 (−5.6%)
	X25	**0.371**	0.4044 (+9%)	0.324 (−12.7%)	0.324 (−12.7%)
	X11	**0.3238**	0.3388 (+4.6%)	0.3537 (+9.2%)	0.3915 (+20.9%)
	Intercept	**-3.879**	−3.606 (−7%)	−4.097 (+5.6%)	−3.879 (0%)
MiceProtein (n: 1080, p: 78, D: 2)	SOD1_N	**1.752**	1.784 (+1.8%)	1.792 (+2.3%)	1.875 (+7%)
	pCAMKII_N	**1.226**	1.307 (+6.6%)	1.303 (+6.3%)	1.196 (−2.4%)
	GluR3_N	**1.059**	1.062 (+0.3%)	1.05 (−0.8%)	0.9862 (−6.9%)
	BRAF_N	**1.057**	1.121 (+6.1%)	1.06 (+0.3%)	1.047 (−0.9%)
	CaNA_N	**0.9413**	1.015 (+7.8%)	0.9779 (+3.9%)	1.031 (+9.5%)
	Intercept	**-25.18**	−24.91 (−1.1%)	−24.92 (−1%)	−25.18 (0%)
PhishingWebsites (n: 11055, p: 31, D: 5)	URL_of_Anchor.0	**3.328**	3.581 (+7.6%)	3.313 (−0.5%)	3.187 (−4.2%)
	SSLfinal_State.1	**3.044**	3.45 (+13.3%)	3.026 (−0.6%)	3.991 (+31.1%)
	URL_of_Anchor.1	**2.665**	2.351 (−11.8%)	2.615 (−1.9%)	2.373 (−11%)
	URL_of_Anchor.0:URL_of_Anchor.1	**1.646**	1.353 (−17.8%)	1.705 (+3.6%)	1.642 (−0.2%)
	Intercept	**0.07477**	0.4047 (+441.3%)	−0.9864 (−1419.2%)	0.07477 (0%)
adult (n: 48842, p: 15, D: 5)	age	**0.7777**	0.8597 (+10.5%)	0.7766 (−0.1%)	0.8176 (+5.1%)
	capital.gain	**0.6253**	0.4404 (−29.6%)	0.5037 (−19.4%)	0.6606 (+5.6%)
	marital.status.Never.married	**0.6014**	0.6082 (+1.1%)	0.6356 (+5.7%)	0.6055 (+0.7%)
	education.num	**0.3952**	0.3975 (+0.6%)	0.3989 (+0.9%)	0.4742 (+20%)
	Intercept	**-2.074**	−2.7 (+30.2%)	−2.572 (+24%)	−2.074 (0%)
analcatdata_authorship (n: 841, p: 71, D: 1)	the	**1.519**	1.514 (−0.3%)	1.525 (+0.4%)	1.519 (0%)
	was	**1.326**	1.37 (+3.3%)	1.325 (−0.1%)	1.326 (0%)
	her	**1.1**	1.102 (+0.2%)	1.11 (+0.9%)	1.1 (0%)
	should	**0.8523**	0.8532 (+0.1%)	0.837 (−1.8%)	0.8523 (0%)
	Intercept	**-8.48**	−8.531 (+0.6%)	−8.648 (+2%)	−8.48 (0%)
analcatdata_dmft (n: 797, p: 5, D: 1)	Intercept	**0.4323**	0.5592 (+29.4%)	0.3711 (−14.2%)	0.4323 (0%)
	Ethnic.White	**0.2147**	0.2171 (+1.1%)	0.2132 (−0.7%)	0.2147 (0%)
	Ethnic.Dark	**0.119**	0.1117 (−6.1%)	0.1234 (+3.7%)	0.119 (0%)
	DMFT.Begin.4	**0.03935**	0.05373 (+36.5%)	0.03803 (−3.4%)	0.03935 (0%)
	DMFT.End.5	**0.03664**	0.03443 (−6%)	0.0418 (+14.1%)	0.03664 (0%)
balance-scale (n: 625, p: 5, D: 2)	left.distance	**0.9896**	1.081 (+9.2%)	1.005 (+1.6%)	0.9896 (0%)

	right.weight	**0.5251**	0.4571 (−12.9%)	0.55 (+4.7%)	0.5251 (0%)
	right.distance:right.weight	**0.4602**	0.4382 (−4.8%)	0.4897 (+6.4%)	0.4602 (0%)
	left.distance:right.distance	**0.441**	0.4379 (−0.7%)	0.4528 (+2.7%)	0.441 (0%)
	Intercept	**-2.739**	−2.546 (−7%)	−2.742 (+0.1%)	−2.739 (0%)
bank-marketing (n: 45211, p: 17, D: 5)	Intercept	**4.181**	4.418 (+5.7%)	4.298 (+2.8%)	4.181 (0%)
	V12	**1.257**	1.272 (+1.2%)	1.232 (−2%)	1.287 (+2.4%)
	V9.unknown	**0.5507**	0.3857 (−30%)	0.669 (+21.5%)	0.6991 (+26.9%)
	V11.may	**0.3244**	0.3225 (−0.6%)	0.3487 (+7.5%)	0.2573 (−20.7%)
	V12:V9.unknown	**0.2655**	0.2475 (−6.8%)	0.2858 (+7.6%)	0.308 (+16%)
	V10	**0.2515**	0.2542 (+1.1%)	0.169 (−32.8%)	0.1644 (−34.6%)
banknote-authentication (n: 1372, p: 5, D: 2)	V1	**5.331**	5.392 (+1.1%)	5.242 (−1.7%)	5.306 (−0.5%)
	V2	**4.571**	4.732 (+3.5%)	4.696 (+2.7%)	4.399 (−3.8%)
	V3	**4.31**	4.305 (−0.1%)	4.218 (−2.1%)	3.78 (−12.3%)
	Intercept	**2.143**	1.427 (−33.4%)	3.174 (+48.1%)	2.143 (0%)
	V2:V3	**0.8987**	0.8774 (−2.4%)	0.8531 (−5.1%)	0.8824 (−1.8%)
blood-transfusion-service-center (n: 748, p: 5, D: 1)	Intercept	**1.907**	2.078 (+9%)	1.9 (−0.4%)	1.907 (0%)
	V1	**0.6442**	0.6441 (0%)	0.6447 (+0.1%)	0.6442 (0%)
	V2	**0.4394**	0.4455 (+1.4%)	0.4422 (+0.6%)	0.4394 (0%)
	V4	**0.3667**	0.4028 (+9.8%)	0.3913 (+6.7%)	0.3667 (0%)
	V3	**0.156**	0.1566 (+0.4%)	0.1576 (+1%)	0.156 (0%)
breast-w (n: 699, p: 10, D: 1)	Intercept	**2.177**	1.714 (−21.3%)	1.977 (−9.2%)	2.177 (0%)
	Bare_Nuclei	**1.122**	1.147 (+2.2%)	1.139 (+1.5%)	1.122 (0%)
	Cell_Size_Uniformity	**0.879**	0.9123 (+3.8%)	0.8931 (+1.6%)	0.879 (0%)
	Clump_Thickness	**0.6338**	0.5865 (−7.5%)	0.6205 (−2.1%)	0.6338 (0%)
	Cell_Shape_Uniformity	**0.6229**	0.6393 (+2.6%)	0.627 (+0.7%)	0.6229 (0%)
	Bland_Chromatin	**0.5479**	0.5897 (+7.6%)	0.5402 (−1.4%)	0.5479 (0%)
car (n: 1728, p: 7, D: 5)	lug_boot.big	**1.925**	1.703 (−11.5%)	1.762 (−8.5%)	1.741 (−9.6%)
	buying.low	**1.324**	1.01 (−23.7%)	1.259 (−4.9%)	1.324 (0%)
	maint.low	**1.282**	1.389 (+8.3%)	1.156 (−9.8%)	1.253 (−2.3%)
	buying.med	**1.274**	1.551 (+21.7%)	1.415 (+11.1%)	1.364 (+7.1%)
	maint.med	**1.097**	1.179 (+7.5%)	1.15 (+4.8%)	1.041 (−5.1%)
	doors.5more	**1.029**	1.051 (+2.1%)	1.16 (+12.7%)	0.9159 (−11%)
	Intercept	**-10.95**	−10.85 (−0.9%)	−11.23 (+2.6%)	−10.95 (0%)
churn (n: 5000, p: 21, D: 5)	Intercept	**3.316**	3.108 (−6.3%)	3.528 (+6.4%)	3.316 (0%)

	total_day_charge	**0.4751**	0.5453 (+14.8%)	0.3872 (−18.5%)	0.4808 (+1.2%)
	international_plan.1	**0.3634**	0.3857 (+6.1%)	0.314 (−13.6%)	0.3752 (+3.2%)
	international_plan.1:total_intl_calls	**0.2522**	0.3301 (+30.9%)	0.2674 (+6%)	0.2373 (−5.9%)
	number_customer_service_calls.4	**0.226**	0.2262 (+0.1%)	0.1725 (−23.7%)	0.2172 (−3.9%)
	total_intl_calls	**0.1968**	0.2782 (+41.4%)	0.2155 (+9.5%)	0.1921 (−2.4%)
	number_customer_service_calls.5	**0.1169**	0.2977 (+154.7%)	0.1737 (+48.6%)	0.1203 (+2.9%)
climate-model-simulation-crashes (n: 540, p: 19, D: 2)	vconst_corr	**1.001**	0.8937 (−10.7%)	1.028 (+2.7%)	1.002 (+0.1%)
	vconst_2	**0.9663**	0.9107 (−5.8%)	1.023 (+5.9%)	0.9713 (+0.5%)
	convect_corr	**0.7112**	0.6609 (−7.1%)	0.7399 (+4%)	0.7023 (−1.3%)
	bckgrnd_vdc1	**0.4785**	0.4181 (−12.6%)	0.4985 (+4.2%)	0.4794 (+0.2%)
	vconst_2:vconst_corr	**0.4772**	0.465 (−2.6%)	0.4816 (+0.9%)	0.4773 (0%)
	Intercept	**-3.572**	−3.997 (+11.9%)	−3.505 (−1.9%)	−3.572 (0%)
cmc (n: 1473, p: 10, D: 3)	Intercept	**0.2812**	0.311 (+10.6%)	0.2435 (−13.4%)	0.2812 (0%)
	Number_of_children_ever_born	**0.1651**	0.1626 (−1.5%)	0.1678 (+1.6%)	0.1881 (+13.9%)
	Wifes_education.4	**0.1497**	0.1467 (−2%)	0.1324 (−11.6%)	0.135 (−9.8%)
	Wifes_education.3	**0.1123**	0.1067 (−5%)	0.1117 (−0.5%)	0.1051 (−6.4%)
	Wifes_age	**0.1016**	0.1022 (+0.6%)	0.1076 (+5.9%)	0.08719 (−14.2%)
connect-4 (n: 67557, p: 43, D: 5)	g1.1	**0.3457**	0.405 (+17.2%)	0.3575 (+3.4%)	0.3156 (−8.7%)
	a1.1	**0.3032**	0.2637 (−13%)	0.2406 (−20.6%)	0.2835 (−6.5%)
	c2.2	**0.1752**	0.1971 (+12.5%)	0.161 (−8.1%)	0.1605 (−8.4%)
	d1.2	**0.1502**	0.1682 (+12%)	0.1451 (−3.4%)	0.1834 (+22.1%)
	d2.1	**0.1377**	0.1346 (−2.3%)	0.1748 (+26.9%)	0.1006 (−26.9%)
	Intercept	**-0.6306**	−0.7717 (+22.4%)	−0.3145 (−50.1%)	−0.6306 (0%)
credit-approval (n: 690, p: 16, D: 2)	A9.f	**1.606**	1.64 (+2.1%)	1.62 (+0.9%)	1.71 (+6.5%)
	A15	**0.489**	0.4922 (+0.7%)	0.5052 (+3.3%)	0.5344 (+9.3%)
	A8	**0.338**	0.3257 (−3.6%)	0.3255 (−3.7%)	0.351 (+3.8%)
	A11	**0.2694**	0.2514 (−6.7%)	0.2939 (+9.1%)	0.3002 (+11.4%)
	Intercept	**0.1466**	0.1272 (−13.2%)	−0.02637 (−118%)	0.1466 (0%)
credit-g (n: 1000, p: 21, D: 4)	Intercept	**1.708**	1.481 (−13.3%)	1.744 (+2.1%)	1.708 (0%)
	checking_status.no.checking	**0.6726**	0.7092 (+5.4%)	0.6742 (+0.2%)	0.7091 (+5.4%)
	duration	**0.3661**	0.3567 (−2.6%)	0.3834 (+4.7%)	0.3751 (+2.5%)
	credit_amount	**0.3079**	0.3182 (+3.3%)	0.3174 (+3.1%)	0.2982 (−3.2%)
	credit_history.critical.other.existing.credit	**0.2568**	0.2282 (−11.1%)	0.2725 (+6.1%)	0.2447 (−4.7%)
	credit_amount:duration	**0.2043**	0.251 (+22.9%)	0.2353 (+15.2%)	0.1479 (−27.6%)

cylinder-bands (n: 540, p: 38, D: 5)	press_speed	**0.4623**	0.4539 (−1.8%)	0.4728 (+2.3%)	0.4688 (+1.4%)
	ink_pct	**0.3341**	0.3537 (+5.9%)	0.3564 (+6.7%)	0.3298 (−1.3%)
	ESA_Voltage	**0.3306**	0.3192 (−3.4%)	0.3274 (−1%)	0.3402 (+2.9%)
	press_type.woodhoe70	**0.2944**	0.2937 (−0.2%)	0.3007 (+2.1%)	0.2755 (−6.4%)
	solvent_pct	**0.2894**	0.2708 (−6.4%)	0.2735 (−5.5%)	0.285 (−1.5%)
	Intercept	**-0.4583**	−0.0698 (−84.8%)	−0.3707 (−19.1%)	−0.4583 (0%)
diabetes (n: 768, p: 9, D: 1)	Intercept	**1.378**	1.536 (+11.5%)	1.231 (−10.7%)	1.378 (0%)
	plas	**0.7336**	0.7202 (−1.8%)	0.7339 (0%)	0.7336 (0%)
	mass	**0.5288**	0.5546 (+4.9%)	0.5056 (−4.4%)	0.5288 (0%)
	age	**0.3541**	0.3525 (−0.5%)	0.352 (−0.6%)	0.3541 (0%)
	pedi	**0.1807**	0.1833 (+1.4%)	0.1787 (−1.1%)	0.1807 (0%)
dna (n: 3186, p: 181, D: 5)	A92	**0.9427**	0.9694 (+2.8%)	0.9219 (−2.2%)	1.137 (+20.6%)
	A84	**0.51**	0.4448 (−12.8%)	0.5111 (+0.2%)	0.6598 (+29.4%)
	A89	**0.488**	0.4886 (+0.1%)	0.5167 (+5.9%)	0.8322 (+70.5%)
	A93	**0.4782**	0.4644 (−2.9%)	0.3591 (−24.9%)	0.4856 (+1.5%)
	A95	**0.4725**	0.4878 (+3.2%)	0.4933 (+4.4%)	0.4948 (+4.7%)
	A94	**0.4435**	0.4833 (+9%)	0.447 (+0.8%)	0.453 (+2.1%)
	Intercept	**-3.09**	−2.91 (−5.8%)	−3.245 (+5%)	−3.09 (0%)
dresses-sales (n: 500, p: 13, D: 1)	Intercept	**0.7038**	0.8286 (+17.7%)	0.7303 (+3.8%)	0.7038 (0%)
	V6.Spring	**0.2741**	0.2489 (−9.2%)	0.2785 (+1.6%)	0.2741 (0%)
	V10.rayon	**0.1774**	0.1663 (−6.3%)	0.1842 (+3.8%)	0.1774 (0%)
	V10.cotton	**0.1542**	0.1557 (+1%)	0.1546 (+0.3%)	0.1542 (0%)
	V8.short	**0.126**	0.1367 (+8.5%)	0.1246 (−1.1%)	0.126 (0%)
electricity (n: 45312, p: 9, D: 5)	nswprice	**3.821**	3.881 (+1.6%)	3.923 (+2.7%)	3.188 (−16.6%)
	date	**2.297**	2.688 (+17%)	2.224 (−3.2%)	1.796 (−21.8%)
	date:nswprice	**1.55**	1.407 (−9.2%)	1.503 (−3%)	1.327 (−14.4%)
	nswprice:vicprice	**0.3858**	0.6725 (+74.3%)	0.4202 (+8.9%)	0.3269 (−15.3%)
	vicprice	**0.3618**	0.4327 (+19.6%)	0.4099 (+13.3%)	0.4026 (+11.3%)
	Intercept	**-0.5675**	−1.567 (+176.1%)	−0.5729 (+1%)	−0.5675 (0%)
eucalyptus (n: 736, p: 20, D: 2)	Vig	**0.6912**	0.6582 (−4.8%)	0.6811 (−1.5%)	0.7452 (+7.8%)
	Ht	**0.5555**	0.5429 (−2.3%)	0.5752 (+3.5%)	0.5935 (+6.8%)
	Crown_Fm	**0.423**	0.4139 (−2.2%)	0.44 (+4%)	0.4552 (+7.6%)
	Surv	**0.3366**	0.3796 (+12.8%)	0.3976 (+18.1%)	0.3338 (−0.8%)
	Intercept	**-4.564**	−4.236 (−7.2%)	−4.295 (−5.9%)	−4.564 (0%)

first-order-theorem-proving (n: 6118, p: 52, D: 5)	V19	**0.3181**	0.2757 (−13.3%)	0.3417 (+7.4%)	0.2213 (−30.4%)
	V23	**0.2871**	0.2907 (+1.3%)	0.2853 (−0.6%)	0.3112 (+8.4%)
	V38	**0.2435**	0.2327 (−4.4%)	0.221 (−9.2%)	0.1969 (−19.1%)
	V25	**0.2168**	0.2039 (−6%)	0.216 (−0.4%)	0.1875 (−13.5%)
	V39	**0.1941**	0.2417 (+24.5%)	0.1892 (−2.5%)	0.08547 (−56%)
	V4	**0.1925**	0.2139 (+11.1%)	0.2219 (+15.3%)	0.121 (−37.1%)
	V11	**0.1589**	0.1181 (−25.7%)	0.2226 (+40.1%)	0.1219 (−23.3%)
	Intercept	**-4.776**	−4.622 (−3.2%)	−4.835 (+1.2%)	−4.776 (0%)
ilpd (n: 583, p: 11, D: 1)	Intercept	**1.609**	1.48 (−8%)	1.514 (−5.9%)	1.609 (0%)
	V5	**0.2843**	0.2826 (−0.6%)	0.2842 (0%)	0.2843 (0%)
	V4	**0.2712**	0.2474 (−8.8%)	0.2567 (−5.3%)	0.2712 (0%)
	V6	**0.2215**	0.2427 (+9.6%)	0.2218 (+0.1%)	0.2215 (0%)
	V3	**0.1898**	0.1787 (−5.8%)	0.1847 (−2.7%)	0.1898 (0%)
jm1 (n: 10885, p: 22, D: 4)	Intercept	**2.17**	2.035 (−6.2%)	2.217 (+2.2%)	2.17 (0%)
	lOBlank	**0.3751**	0.3452 (−8%)	0.3802 (+1.4%)	0.3056 (−18.5%)
	loc	**0.3712**	0.3421 (−7.8%)	0.3587 (−3.4%)	0.4542 (+22.4%)
	lOBlank:total_Op	**0.1968**	0.1741 (−11.5%)	0.201 (+2.1%)	0.07209 (−63.4%)
	total_Op	**0.188**	0.2046 (+8.8%)	0.1904 (+1.3%)	0.1033 (−45.1%)
	total_Opnd	**0.1595**	0.1797 (+12.7%)	0.1506 (−5.6%)	0.1035 (−35.1%)
	lOBlank:loc	**0.1185**	0.1251 (+5.6%)	0.118 (−0.4%)	0.1152 (−2.8%)
jungle_chess_2pcs_raw_endgame_complete (n: 44819, p: 7, D: 5)	white_piece0_rank	**2.148**	2.407 (+12.1%)	2.14 (−0.4%)	2.18 (+1.5%)
	black_piece0_rank	**1.811**	2.105 (+16.2%)	1.64 (−9.4%)	1.835 (+1.3%)
	black_piece0_rank:black_piece0_strength	**0.9972**	1.041 (+4.4%)	0.9096 (−8.8%)	0.9841 (−1.3%)
	black_piece0_rank:white_piece0_strength	**0.7611**	0.7054 (−7.3%)	0.782 (+2.7%)	0.776 (+2%)
	black_piece0_strength:white_piece0_strength	**0.7242**	0.8054 (+11.2%)	0.789 (+8.9%)	0.7158 (−1.2%)
	Intercept	**-4.976**	−4.464 (−10.3%)	−4.663 (−6.3%)	−4.976 (0%)
kc1 (n: 2109, p: 22, D: 4)	Intercept	**2.664**	2.423 (−9%)	2.599 (−2.4%)	2.664 (0%)
	lOCode	**0.3439**	0.2703 (−21.4%)	0.3193 (−7.2%)	0.3206 (−6.8%)
	e	**0.2808**	0.2559 (−8.9%)	0.2767 (−1.5%)	0.2074 (−26.1%)
	loc	**0.2543**	0.2248 (−11.6%)	0.2558 (+0.6%)	0.2418 (−4.9%)
	i	**0.1886**	0.149 (−21%)	0.1968 (+4.3%)	0.2447 (+29.7%)
	d	**0.1677**	0.1581 (−5.7%)	0.1462 (−12.8%)	0.1092 (−34.9%)
kc2 (n: 522, p: 22, D: 2)	Intercept	**2.356**	2.195 (−6.8%)	2.279 (−3.3%)	2.356 (0%)
	i	**0.4001**	0.399 (−0.3%)	0.3871 (−3.2%)	0.2755 (−31.1%)

	uniq_Opnd	**0.3854**	0.3804 (−1.3%)	0.374 (−3%)	0.3627 (−5.9%)
	total_Opnd	**0.3419**	0.3641 (+6.5%)	0.3712 (+8.6%)	0.3247 (−5%)
	lOBlank	**0.1861**	0.1726 (−7.3%)	0.2096 (+12.6%)	0.1824 (−2%)
	uniq_Op	**0.1585**	0.1759 (+11%)	0.1678 (+5.9%)	0.1498 (−5.5%)
kr-vs-kp (n: 3196, p: 37, D: 5)	bxqsq.f	**3.697**	3.539 (−4.3%)	3.653 (−1.2%)	3.647 (−1.4%)
	wknck.f	**3.488**	3.915 (+12.2%)	3.624 (+3.9%)	3.346 (−4.1%)
	rimmx.f	**3.124**	2.326 (−25.5%)	2.886 (−7.6%)	3.055 (−2.2%)
	bkxbq.f	**1.343**	1.837 (+36.8%)	1.409 (+4.9%)	1.371 (+2.1%)
	Intercept	**0.7984**	0.8956 (+12.2%)	0.2632 (−67%)	0.7984 (0%)
letter (n: 20000, p: 17, D: 5)	x.ege	**4.61**	4.21 (−8.7%)	4.527 (−1.8%)	4.107 (−10.9%)
	y.ege	**3.441**	3.427 (−0.4%)	2.935 (−14.7%)	3.13 (−9%)
	xegvy	**2.67**	2.589 (−3%)	2.68 (+0.4%)	2.408 (−9.8%)
	xybar	**2.45**	2.085 (−14.9%)	2.143 (−12.5%)	2.3 (−6.1%)
	x2ybr	**2.286**	2.37 (+3.7%)	2.258 (−1.2%)	1.993 (−12.8%)
	Intercept	**-131.2**	−128.6 (−2%)	−128.2 (−2.3%)	−131.2 (0%)
mfeat-factors (n: 2000, p: 217, D: 2)	att1	**1.213**	1.191 (−1.8%)	1.187 (−2.1%)	1.222 (+0.7%)
	att181	**0.9014**	0.952 (+5.6%)	0.9538 (+5.8%)	0.9899 (+9.8%)
	att37	**0.7361**	0.8493 (+15.4%)	0.8563 (+16.3%)	0.9363 (+27.2%)
	att194	**0.6411**	0.6299 (−1.7%)	0.6215 (−3.1%)	0.6047 (−5.7%)
	att213	**0.6099**	0.6938 (+13.8%)	0.6682 (+9.6%)	0.6421 (+5.3%)
	Intercept	**-37.82**	−38.57 (+2%)	−37.7 (−0.3%)	−37.82 (0%)
mfeat-fourier (n: 2000, p: 77, D: 5)	att7	**0.9039**	0.8597 (−4.9%)	0.805 (−10.9%)	1.171 (+29.5%)
	att73	**0.8344**	0.8666 (+3.9%)	0.891 (+6.8%)	1.116 (+33.7%)
	att2	**0.8336**	0.8625 (+3.5%)	0.7984 (−4.2%)	0.7765 (−6.8%)
	att76	**0.5851**	0.4715 (−19.4%)	0.5686 (−2.8%)	0.6125 (+4.7%)
	att6	**0.54**	0.601 (+11.3%)	0.6122 (+13.4%)	0.6534 (+21%)
	Intercept	**-26.19**	−26.55 (+1.4%)	−26.44 (+1%)	−26.19 (0%)
mfeat-karhunen (n: 2000, p: 65, D: 5)	att1	**1.469**	1.548 (+5.4%)	1.439 (−2%)	1.522 (+3.6%)
	att3	**1.318**	1.246 (−5.5%)	1.301 (−1.3%)	1.487 (+12.8%)
	att4	**1.126**	1.338 (+18.8%)	1.064 (−5.5%)	1.133 (+0.6%)
	att7	**0.9811**	1.109 (+13%)	0.9565 (−2.5%)	1.126 (+14.8%)
	att2	**0.8767**	0.7276 (−17%)	1.049 (+19.7%)	0.9546 (+8.9%)
	Intercept	**-33.11**	−33.67 (+1.7%)	−33.2 (+0.3%)	−33.11 (0%)
mfeat-morphological (n: 2000, p: 7, D: 1)	att4	**3.748**	3.794 (+1.2%)	3.589 (−4.2%)	3.748 (0%)

	att2	**3.318**	3.212 (−3.2%)	3.308 (−0.3%)	3.318 (0%)
	att1	**2.913**	2.857 (−1.9%)	2.921 (+0.3%)	2.913 (0%)
	att6	**1.96**	1.964 (+0.2%)	1.957 (−0.2%)	1.96 (0%)
	Intercept	**-21.72**	−23.36 (+7.6%)	−22.37 (+3%)	−21.72 (0%)
mfeat-pixel (n: 2000, p: 241, D: 5)	att162	**0.4647**	0.4508 (−3%)	0.4105 (−11.7%)	0.6221 (+33.9%)
	att19	**0.4553**	0.3895 (−14.5%)	0.4393 (−3.5%)	0.507 (+11.4%)
	att57	**0.4189**	0.4877 (+16.4%)	0.4943 (+18%)	0.4012 (−4.2%)
	att214	**0.412**	0.4123 (+0.1%)	0.4044 (−1.8%)	0.4574 (+11%)
	att153	**0.3907**	0.4174 (+6.8%)	0.442 (+13.1%)	0.585 (+49.7%)
	att113	**0.3706**	0.4066 (+9.7%)	0.359 (−3.1%)	0.5165 (+39.4%)
	Intercept	**-31.93**	−31.29 (−2%)	−31.89 (−0.1%)	−31.93 (0%)
mfeat-zernike (n: 2000, p: 48, D: 2)	att29	**1.137**	1.151 (+1.2%)	1.141 (+0.4%)	1.186 (+4.3%)
	att36	**1.029**	1.11 (+7.9%)	1.068 (+3.8%)	1.028 (−0.1%)
	att45	**0.9863**	0.9713 (−1.5%)	0.9926 (+0.6%)	0.7626 (−22.7%)
	att43	**0.958**	1.018 (+6.3%)	0.9376 (−2.1%)	0.9981 (+4.2%)
	att33	**0.9426**	0.9923 (+5.3%)	0.9413 (−0.1%)	0.8801 (−6.6%)
	att19	**0.8309**	0.8354 (+0.5%)	0.7189 (−13.5%)	1.071 (+28.9%)
	Intercept	**-26.83**	−26.25 (−2.2%)	−27.78 (+3.5%)	−26.83 (0%)
nomao (n: 34465, p: 119, D: 5)	V6	**1.002**	0.9554 (−4.7%)	1.065 (+6.3%)	1.403 (+40%)
	V90	**0.8547**	0.7562 (−11.5%)	0.8676 (+1.5%)	0.9458 (+10.7%)
	V4	**0.6198**	0.6085 (−1.8%)	0.6263 (+1%)	0.676 (+9.1%)
	V100.3	**0.5639**	0.5918 (+4.9%)	0.6077 (+7.8%)	0.8928 (+58.3%)
	V61	**0.5574**	0.593 (+6.4%)	0.6077 (+9%)	0.4243 (−23.9%)
	V1	**0.5561**	0.5659 (+1.8%)	0.6561 (+18%)	0.8947 (+60.9%)
	Intercept	**-3.063**	−2.635 (−14%)	−1.931 (−37%)	−3.063 (0%)
numerai28.6 (n: 96320, p: 22, D: 2)	Intercept	**0.4794**	0.4686 (−2.3%)	0.4916 (+2.5%)	0.4794 (0%)
	attribute_13	**0.03296**	0.03171 (−3.8%)	0.03317 (+0.6%)	0.03048 (−7.5%)
	attribute_1	**0.02981**	0.02981 (0%)	0.02936 (−1.5%)	0.02959 (−0.7%)
	attribute_14	**0.02838**	0.02969 (+4.6%)	0.02848 (+0.4%)	0.02714 (−4.4%)
	attribute_16	**0.0274**	0.02492 (−9.1%)	0.02681 (−2.2%)	0.0282 (+2.9%)
optdigits (n: 5620, p: 65, D: 4)	input27	**1.119**	1.15 (+2.8%)	1.181 (+5.5%)	1.396 (+24.8%)
	input63	**0.8117**	0.8321 (+2.5%)	0.7816 (−3.7%)	0.9487 (+16.9%)
	input37	**0.7809**	0.8437 (+8%)	0.8531 (+9.2%)	1.058 (+35.5%)
	input54	**0.7408**	0.8106 (+9.4%)	0.7068 (−4.6%)	0.8397 (+13.4%)

	Intercept	**-35.07**	−34.46 (−1.7%)	−34.8 (−0.8%)	−35.07 (0%)
ozone-level-8hr (n: 2534, p: 73, D: 4)	Intercept	**4.69**	4.52 (−3.6%)	4.385 (−6.5%)	4.69 (0%)
	V41	**0.4146**	0.4809 (+16%)	0.4343 (+4.8%)	0.3055 (−26.3%)
	V56	**0.3559**	0.3246 (−8.8%)	0.3718 (+4.5%)	0.351 (−1.4%)
	V42	**0.2362**	0.2628 (+11.3%)	0.2491 (+5.5%)	0.1838 (−22.2%)
	V13	**0.149**	0.1463 (−1.8%)	0.1651 (+10.8%)	0.1626 (+9.1%)
	V55	**0.1438**	0.1736 (+20.7%)	0.1455 (+1.2%)	0.1416 (−1.5%)
pc1 (n: 1109, p: 22, D: 3)	Intercept	**3.846**	3.877 (+0.8%)	3.84 (−0.2%)	3.846 (0%)
	lOBlank	**0.544**	0.5678 (+4.4%)	0.5165 (−5.1%)	0.5753 (+5.8%)
	lOBlank:lOComment	**0.3371**	0.3327 (−1.3%)	0.339 (+0.6%)	0.2128 (−36.9%)
	I	**0.3226**	0.3427 (+6.2%)	0.3042 (−5.7%)	0.3796 (+17.7%)
	uniq_Opnd	**0.1911**	0.1854 (−3%)	0.19 (−0.6%)	0.1207 (−36.8%)
	locCodeAndComment	**0.1888**	0.2501 (+32.5%)	0.1761 (−6.7%)	0.1853 (−1.9%)
pc3 (n: 1563, p: 38, D: 4)	Intercept	**3.426**	3.742 (+9.2%)	3.513 (+2.5%)	3.426 (0%)
	LOC_BLANK	**0.6684**	0.6975 (+4.4%)	0.6552 (−2%)	0.6323 (−5.4%)
	HALSTEAD_CONTENT	**0.4008**	0.4438 (+10.7%)	0.3974 (−0.8%)	0.3645 (−9.1%)
	NUM_UNIQUE_OPERANDS	**0.1638**	0.1682 (+2.7%)	0.1575 (−3.8%)	0.2197 (+34.1%)
	HALSTEAD_LEVEL	**0.1331**	0.1263 (−5.1%)	0.1239 (−6.9%)	0.09466 (−28.9%)
	LOC_CODE_AND_COMMENT	**0.1328**	0.1559 (+17.4%)	0.1392 (+4.8%)	0.1227 (−7.6%)
	NUMBER_OF_LINES	**0.1213**	0.09314 (−23.2%)	0.1474 (+21.5%)	0.1031 (−15%)
pc4 (n: 1458, p: 38, D: 3)	Intercept	**4.275**	3.654 (−14.5%)	4.38 (+2.5%)	4.275 (0%)
	LOC_CODE_AND_COMMENT	**1.507**	1.59 (+5.5%)	1.471 (−2.4%)	1.64 (+8.8%)
	CYCLOMATIC_DENSITY	**0.3314**	0.3703 (+11.7%)	0.3038 (−8.3%)	0.3219 (−2.9%)
	LOC_BLANK	**0.3061**	0.3236 (+5.7%)	0.3011 (−1.6%)	0.2824 (−7.7%)
	CONDITION_COUNT	**0.2766**	0.2901 (+4.9%)	0.2584 (−6.6%)	0.2782 (+0.6%)
pendigits (n: 10992, p: 17, D: 5)	input16	**1.334**	1.591 (+19.3%)	1.309 (−1.9%)	1.176 (−11.8%)
	input14	**1.067**	1.08 (+1.2%)	1.077 (+0.9%)	1.1 (+3.1%)
	input2	**1.06**	1.166 (+10%)	1.032 (−2.6%)	1.375 (+29.7%)
	input10	**1.009**	1.026 (+1.7%)	1.004 (−0.5%)	0.9265 (−8.2%)
	input7	**0.7723**	0.8284 (+7.3%)	0.9029 (+16.9%)	1.026 (+32.8%)
	Intercept	**-36.75**	−37.21 (+1.3%)	−36.65 (−0.3%)	−36.75 (0%)
phoneme (n: 5404, p: 6, D: 5)	Intercept	**2.866**	2.469 (−13.9%)	2.834 (−1.1%)	2.866 (0%)
	V4	**1.115**	0.9594 (−14%)	1.062 (−4.8%)	1.271 (+14%)
	V1	**0.9529**	1.166 (+22.4%)	0.8947 (−6.1%)	0.994 (+4.3%)

	V3	**0.789**	0.4782 (−39.4%)	0.689 (−12.7%)	0.8698 (+10.2%)
	V2	**0.7639**	0.662 (−13.3%)	0.8119 (+6.3%)	0.6761 (−11.5%)
	V1:V3	**0.7302**	0.7381 (+1.1%)	0.6831 (−6.5%)	0.6196 (−15.1%)
qsar-biodeg (n: 1055, p: 42, D: 4)	Intercept	**1.872**	2.18 (+16.5%)	1.829 (−2.3%)	1.872 (0%)
	V36	**0.5753**	0.5523 (−4%)	0.5769 (+0.3%)	0.7878 (+36.9%)
	V38	**0.5006**	0.4735 (−5.4%)	0.4765 (−4.8%)	0.493 (−1.5%)
	V1	**0.2923**	0.304 (+4%)	0.3126 (+6.9%)	0.3063 (+4.8%)
	V22	**0.247**	0.2745 (+11.1%)	0.2252 (−8.8%)	0.2285 (−7.5%)
	V27	**0.2454**	0.253 (+3.1%)	0.2646 (+7.8%)	0.2759 (+12.4%)
satimage (n: 6430, p: 37, D: 5)	F12attr	**1.031**	1.161 (+12.6%)	1.022 (−0.9%)	0.3288 (−68.1%)
	D16attr	**0.6645**	0.7351 (+10.6%)	0.6434 (−3.2%)	0.5567 (−16.2%)
	C15attr	**0.6632**	0.5949 (−10.3%)	0.6421 (−3.2%)	0.3713 (−44%)
	F24attr	**0.6431**	0.6393 (−0.6%)	0.6657 (+3.5%)	0.4283 (−33.4%)
	E11attr	**0.604**	0.5513 (−8.7%)	0.4937 (−18.3%)	0.9466 (+56.7%)
	B8attr	**0.5692**	0.5816 (+2.2%)	0.5755 (+1.1%)	0.5131 (−9.9%)
	A7attr	**0.5248**	0.5368 (+2.3%)	0.4933 (−6%)	0.5098 (−2.9%)
	Intercept	**-13.61**	−13.86 (+1.8%)	−13.51 (−0.7%)	−13.61 (0%)
segment (n: 2310, p: 17, D: 3)	hue.mean	**1.577**	1.472 (−6.7%)	1.573 (−0.3%)	1.686 (+6.9%)
	intensity.mean	**1.038**	1.246 (+20%)	1.081 (+4.1%)	1.174 (+13.1%)
	rawgreen.mean	**0.9083**	1.009 (+11.1%)	0.9338 (+2.8%)	0.689 (−24.1%)
	rawblue.mean	**0.7006**	0.5916 (−15.6%)	0.6722 (−4.1%)	0.6068 (−13.4%)
	rawred.mean	**0.6916**	0.7473 (+8.1%)	0.7169 (+3.7%)	0.3869 (−44.1%)
	exgreen.mean	**0.6305**	0.7868 (+24.8%)	0.5637 (−10.6%)	0.5051 (−19.9%)
	Intercept	**-16.76**	−16.65 (−0.7%)	−16.73 (−0.2%)	−16.76 (0%)
semeion (n: 1593, p: 257, D: 3)	V16	**0.5414**	0.5967 (+10.2%)	0.5155 (−4.8%)	0.4258 (−21.4%)
	V229	**0.4821**	0.5496 (+14%)	0.4654 (−3.5%)	0.5539 (+14.9%)
	V77	**0.4651**	0.5043 (+8.4%)	0.4314 (−7.2%)	0.5042 (+8.4%)
	V96	**0.4304**	0.4823 (+12.1%)	0.3951 (−8.2%)	0.6107 (+41.9%)
	V15	**0.4267**	0.4812 (+12.8%)	0.4353 (+2%)	0.3625 (−15%)
	V177	**0.417**	0.4086 (−2%)	0.425 (+1.9%)	0.4731 (+13.5%)
	Intercept	**-30.14**	−29.92 (−0.7%)	−30.24 (+0.3%)	−30.14 (0%)
spambase (n: 4601, p: 58, D: 4)	Intercept	**1.827**	1.727 (−5.5%)	1.397 (−23.5%)	1.827 (0%)
	word_freq_george	**0.9277**	0.8711 (−6.1%)	0.9642 (+3.9%)	1.023 (+10.3%)
	word_freq_hp	**0.8724**	0.8814 (+1%)	0.8138 (−6.7%)	0.9547 (+9.4%)

	char_freq_.21	**0.4842**	0.4836 (−0.1%)	0.5165 (+6.7%)	0.858 (+77.2%)
	capital_run_length_longest	**0.4376**	0.4276 (−2.3%)	0.426 (−2.7%)	0.4318 (−1.3%)
	capital_run_length_total	**0.4237**	0.4077 (−3.8%)	0.4321 (+2%)	0.4258 (+0.5%)
	char_freq_.24	**0.372**	0.3631 (−2.4%)	0.3414 (−8.2%)	0.6447 (+73.3%)
splice (n: 3190, p: 61, D: 5)	attribute_32.G	**0.7944**	0.771 (−2.9%)	0.8258 (+4%)	0.7655 (−3.6%)
	attribute_31.A	**0.7598**	0.7031 (−7.5%)	0.7579 (−0.3%)	0.7452 (−1.9%)
	attribute_30.C	**0.7277**	0.6369 (−12.5%)	0.798 (+9.7%)	0.722 (−0.8%)
	attribute_30.T	**0.6651**	0.811 (+21.9%)	0.6216 (−6.5%)	0.6703 (+0.8%)
	attribute_32.C	**0.6098**	0.6685 (+9.6%)	0.4642 (−23.9%)	0.6048 (−0.8%)
	attribute_32.A	**0.6011**	0.4873 (−18.9%)	0.6426 (+6.9%)	0.6169 (+2.6%)
	Intercept	**-3.597**	−4.027 (+12%)	−3.588 (−0.3%)	−3.597 (0%)
steel-plates-fault (n: 1941, p: 28, D: 5)	V14	**0.8236**	0.816 (−0.9%)	0.7835 (−4.9%)	1.181 (+43.4%)
	V25	**0.8116**	0.6918 (−14.8%)	0.9687 (+19.4%)	0.8836 (+8.9%)
	V11	**0.7111**	0.8872 (+24.8%)	0.7183 (+1%)	0.7845 (+10.3%)
	V12	**0.6505**	0.6886 (+5.9%)	0.7197 (+10.6%)	0.7353 (+13%)
	Intercept	**-13.57**	−13.88 (+2.3%)	−13.3 (−2%)	−13.57 (0%)
texture (n: 5500, p: 41, D: 3)	V23	**2.455**	2.444 (−0.4%)	2.389 (−2.7%)	2.505 (+2%)
	V10	**2.006**	2.037 (+1.5%)	1.87 (−6.8%)	1.206 (−39.9%)
	V3	**1.91**	2.004 (+4.9%)	1.893 (−0.9%)	1.209 (−36.7%)
	V30	**1.63**	1.515 (−7.1%)	1.558 (−4.4%)	1.083 (−33.6%)
	V6	**0.8912**	0.8164 (−8.4%)	0.8066 (−9.5%)	1.208 (+35.5%)
	Intercept	**-39.86**	−41.69 (+4.6%)	−40.01 (+0.4%)	−39.86 (0%)
tic-tac-toe (n: 958, p: 10, D: 3)	middle.middle.square.o	**2.332**	2.278 (−2.3%)	2.198 (−5.7%)	2.012 (−13.7%)
	middle.middle.square.x	**1.882**	1.981 (+5.3%)	1.764 (−6.3%)	1.554 (−17.4%)
	top.left.square.o	**1.71**	1.696 (−0.8%)	1.535 (−10.2%)	1.501 (−12.2%)
	bottom.right.square.o	**1.703**	1.827 (+7.3%)	1.872 (+9.9%)	1.438 (−15.6%)
	top.right.square.x	**1.506**	1.6 (+6.2%)	1.643 (+9.1%)	1.349 (−10.4%)
	Intercept	**-1.818**	−2.839 (+56.2%)	−1.341 (−26.2%)	−1.818 (0%)
vehicle (n: 846, p: 19, D: 2)	ELONGATEDNESS	**1.544**	1.468 (−4.9%)	1.596 (+3.4%)	1.326 (−14.1%)
	DISTANCE_CIRCULARITY	**1.168**	1.185 (+1.5%)	1.209 (+3.5%)	0.9147 (−21.7%)
	MAX.LENGTH_RECTANGULARITY	**1.02**	1.063 (+4.2%)	0.9321 (−8.6%)	1.032 (+1.2%)
	KURTOSIS_ABOUT_MINOR	**0.6494**	0.7477 (+15.1%)	0.7149 (+10.1%)	0.3895 (−40%)
	SKEWNESS_ABOUT_MAJOR	**0.6262**	0.5876 (−6.2%)	0.6585 (+5.2%)	0.6867 (+9.7%)
	Intercept	**-5.312**	−5.066 (−4.6%)	−5.296 (−0.3%)	−5.312 (0%)

vowel (n: 990, p: 13, D: 3)	Feature_0	**2.975**	2.916 (−2%)	3.096 (+4.1%)	2.701 (−9.2%)
	Feature_4	**2.426**	2.455 (+1.2%)	2.449 (+0.9%)	2.429 (+0.1%)
	Feature_1	**2.368**	2.45 (+3.5%)	2.433 (+2.7%)	2.592 (+9.5%)
	Feature_0:Feature_1	**1.517**	1.476 (−2.7%)	1.392 (−8.2%)	1.273 (−16.1%)
	Intercept	**-39.77**	−38.88 (−2.2%)	−40.37 (+1.5%)	−39.77 (0%)
wall-robot-navigation (n: 5456, p: 25, D: 3)	V15	**2.281**	2.427 (+6.4%)	2.339 (+2.5%)	3.076 (+34.9%)
	V14:V15	**1.624**	1.621 (−0.2%)	1.777 (+9.4%)	1.087 (−33.1%)
	V14	**1.381**	1.704 (+23.4%)	1.53 (+10.8%)	1.387 (+0.4%)
	V20	**1.246**	1.238 (−0.6%)	1.375 (+10.4%)	1.367 (+9.7%)
	V19	**0.8858**	0.8706 (−1.7%)	0.9571 (+8%)	1.485 (+67.6%)
	Intercept	**-10.77**	−11.56 (+7.3%)	−10.79 (+0.2%)	−10.77 (0%)
wdbc (n: 569, p: 31, D: 5)	Intercept	**2.367**	3.815 (+61.2%)	3.279 (+38.5%)	2.367 (0%)
	V28	**1.124**	1.109 (−1.3%)	1.1 (−2.1%)	1.502 (+33.6%)
	V21	**0.9865**	0.9992 (+1.3%)	0.9409 (−4.6%)	1.469 (+48.9%)
	V8	**0.9086**	0.8257 (−9.1%)	0.8677 (−4.5%)	1.189 (+30.9%)
	V22	**0.7937**	0.8219 (+3.6%)	0.7633 (−3.8%)	0.8532 (+7.5%)
	V23	**0.7138**	0.6631 (−7.1%)	0.6834 (−4.3%)	0.9698 (+35.9%)
wilt (n: 4839, p: 6, D: 3)	Intercept	**6.194**	6.613 (+6.8%)	6.117 (−1.2%)	6.194 (0%)
	Mean_G:Mean_R	**3.12**	3.096 (−0.8%)	3.103 (−0.5%)	1.99 (−36.2%)
	Mean_G	**2.903**	3.066 (+5.6%)	2.981 (+2.7%)	2.175 (−25.1%)
	Mean_R	**2.829**	2.775 (−1.9%)	2.851 (+0.8%)	1.73 (−38.8%)
	SD_Plan	**0.3629**	0.3611 (−0.5%)	0.3538 (−2.5%)	0.3564 (−1.8%)
	Mean_NIR	**0.3565**	0.3651 (+2.4%)	0.3686 (+3.4%)	0.3257 (−8.6%)

Chapter 3

AX-Invariance

Abstract

Let Y denote a response, let A be an auxiliary attribute, and let X contain the features available for prediction. Many prediction targets depend not only on the conditional response distribution $P^{Y|A,X}$, but also on the population-specific distribution of (A, X). For example, the conditional mean can be written as $\mathbb{E}_P[Y \mid X = x] = \int \mathbb{E}_P[Y \mid A = a, X = x] \, \mathrm{d}P^{A|X=x}(a)$, and may therefore change when population composition changes even if $P^{Y|A,X}$ remains fixed. We introduce AX-invariance, which requires a prediction functional to depend on P only through $P^{Y|A,X}$. The principle is relevant whenever the distribution of (A, X) is unstable, unavailable at deployment, or should not determine the prediction target. This includes transfer learning with structurally missing covariates and, as our principal application, fairness with protected attributes. In the latter setting, we discuss that AX-invariance is necessary for causal fairness at the functional level whenever fairness requires the functional to be unaffected by the causal mechanism generating (A, X). We formulate functional-level path-specific and interventional fairness and show that under weak assumptions both imply AX-invariance; under stronger causal identification assumptions, they are equivalent to it. We construct AX-invariant estimands using prespecified reference weights and, under squared loss, characterise a continuous family from reference averaging to a pointwise worst-case target. Finally, we develop an asymptotic test with rejection falsifying AX-invariance. Simulations and a real data application provide descriptive illustrations.

3.1 Introduction

Prediction targets can depend on a population in two conceptually distinct ways: through the conditional behaviour of the response and through the composition of the covariates. Let Y be an outcome, let X contain the features used for prediction,

and let A be an auxiliary or stratifying attribute. The conditional mean regression target satisfies

$$\mathbb{E}_P[Y \mid X = x] = \int_{\mathcal{A}} \mathbb{E}_P[Y \mid A = a, X = x] \, \mathrm{d}P^{A|X=x}(a).$$

The conditional expectation, $\mathbb{E}_P[Y \mid A, X]$ describes response behaviour within the (A, X)-strata, whereas the weights describe how those strata are populated given $X = x$. Consequently, the target can change across places, periods, or deployment populations when $P^{A|X=x}$ changes, even if the full conditional response distribution $P^{Y|A,X}$ remains the same. This dependence is easy to overlook because the resulting prediction rule, $x \mapsto g(x) = \mathbb{E}_P[Y \mid X = x]$, is written only as a function of x. The arguments of a prediction rule do not reveal which population quantities were used to construct it. The conditional mean $\mathbb{E}_P[Y \mid X = x]$ is an example of the fact that excluding A from the final prediction rule does not prevent X from carrying information about A, or the target from using the population-specific distribution of A given X. The arguments of a prediction rule do not reveal how that rule was constructed.

This issue may arise in applications where the distribution of (A, X) is unstable, unknown, or unavailable at deployment, or when population composition is not intended to define the prediction target. For example, a model may be transferred between hospitals, countries, or measurement systems in which X contains variables recorded everywhere, whereas A contains variables observed only in the source population.

Our principal motivation is fairness. Suppose that A is a protected attribute and that the deployed prediction rule is required to be a function of X alone. Omitting A from the final rule does not prevent X from carrying information about A, nor does it prevent the population-specific distribution $P^{A|X}$ from determining the target. The conditional mean above is the leading example: it combines state-specific response surfaces using the protected-group composition observed at each x.

This functional-level perspective is also close to the level at which indirect discrimination is discussed in regulation. In the insurance context, European Commission guidance following the *Test-Achats* judgment distinguishes variables that merely correlate with gender from variables that are "true risk factors in their own right". EIOPA's Consultative Expert Group similarly emphasises that factors correlated with protected characteristics require objective justification and that their use should be appropriate and necessary (European Commission, 2012; EIOPA Consultative Expert Group on Digital Ethics in Insurance, 2021). The regulatory concern is therefore not exhausted by checking whether the protected attribute appears as an argument of the final formula. It also concerns whether an apparently neutral factor contributes genuine risk information or is valuable primarily because it acts as a substitute for a protected characteristic.

We formalise the underlying statistical principle as AX-invariance. A prediction functional maps a population distribution P to a measurable prediction function $x \mapsto \Phi_P(x)$. We call the functional AX-invariant if any two populations with the same conditional distribution $P^{Y|A,X}$ produce the same prediction function, regardless of how (A, X) is distributed. In plain terms, the target may use how Y behaves within each (A, X)-stratum, but not how frequently the strata occur in the population under study. Because the target assigns a value pointwise at each x, the definition excludes dependence on the marginal distribution P^X as well as on $P^{A|X}$.

A particularly close existing formulation of this proxy channel is given by Lindholm et al. (2024). In insurance pricing, they define a pricing functional as avoiding proxy discrimination when it is unchanged across portfolios that share $P^{Y|A,X}$, P^A, and P^X but differ in the dependence between A and X. Their formulation likewise places the restriction at the level of the functional rather than the realised price, and it shows that the unawareness price $\mathbb{E}_P[Y \mid X]$ generally fails the requirement. AX-invariance is closely related but uses a larger comparison class: it permits the entire joint distribution $P^{A,X}$, including both marginals, to vary while $P^{Y|A,X}$ is held fixed. Consequently, a reference distribution chosen as the current marginal P^A may satisfy the fixed-marginal criterion of Lindholm et al. (2024) but is not AX-invariant.

Statistical fairness is commonly formulated through constraints on predictions or errors within a population (Mitchell et al., 2021; Hardt et al., 2016; Agarwal et al., 2019). A complementary causal literature asks how predictions or outcomes behave under counterfactual, path-specific or interventional changes to a protected attribute (Kilbertus et al., 2017; Kusner et al., 2017; Nabi and Shpitser, 2018; Wu et al., 2019). Recent work also develops general tools for learning prediction targets under fairness constraints (Nabi et al., 2024). Predictor-level fairness criteria evaluate the resulting prediction rule while holding it fixed and therefore do not assess how the rule was constructed, including whether its generating functional used the population-specific distribution $P^{A,X}$; when such dependence is itself the fairness concern, predictor-level criteria are insufficient on their own. AX-invariance addresses the question whether the functional selecting the prediction target may change when upstream population composition changes but the conditional response distribution does not.

This question is related to work on fairness under covariate and subpopulation shift (Rezaei et al., 2021; Maity et al., 2021; Chen et al., 2022) and to work that seeks stable predictive relationships across environments (Peters et al., 2016; Rojas-Carulla et al., 2018; Henzi et al., 2025). Those literatures typically study the performance, calibration, or fairness of a learned predictor across specified environments or shift classes. We instead impose a property directly on the population target over the full class of distributions sharing $P^{Y|A,X}$.

The paper makes three main contributions. First, we formalise the distinction between fairness of a fixed predictor and fairness of the functional that produces it.

We show that a fixed predictor may be counterfactually fair even when its generating functional is not AX-invariant. We then show that, under the causal conditions stated below, functional-level path-specific and interventional fairness require AX-invariance; in specified cases, the notions coincide. Failure of AX-invariance can therefore rule out these causal requirements, although invariance alone does not establish causal fairness.

Second, we construct AX-invariant prediction targets by aggregating state-specific prediction risks relative to prespecified reference weights. Under squared loss, we characterise a continuous family that ranges from reference averaging to the worst-case target, which minimises the largest state-specific prediction risk. The reference-averaged endpoint is closely related to proposals for discrimination-free statistical profiling and insurance pricing (Pope and Sydnor, 2011; Lindholm et al., 2022, 2024); the full family gives reference standardisation a distributionally robust interpretation (Duchi and Namkoong, 2021; Sagawa et al., 2020; Kuhn et al., 2025). Both the reference weights and the degree of robustness remain explicit.

Third, we develop a data-based diagnostic that rebalances the states of A across values of X while preserving the conditional distribution of Y given (A, X), using distribution-shift resampling (Thams et al., 2023). Under the stated conditions, a systematic detected change in fitted predictions falsifies AX-invariance. Simulations and an adapted Census income example (Ding et al., 2021) provide descriptive illustrations.

Section 3.2 introduces AX-invariance, Section 3.3 develops its causal interpretation, Section 3.4 constructs invariant targets, and Section 3.5 develops the diagnostic. Empirical illustrations are given in Section 3.6.

3.2 Population-composition invariance

Let $Y \in \mathcal{Y} \subseteq \mathbb{R}$ be the response, let $A \in \mathcal{A}$ be an auxiliary or stratifying attribute, and let $X \in \mathcal{X}$ be some features. Our formal results allow any compact metric space $\mathcal{A}$, any standard Borel feature space $\mathcal{X}$, and any Borel subset $\mathcal{Y} \subseteq \mathbb{R}$.

For a distribution P, write $P^{Y|A,X}$ for the conditional distribution of Y given (A, X). In more precise terms, we choose $P^{Y|A,X}$ to be a Markov kernel, i.e., a family of conditional distributions of Y given $(A = a, X = x)$, that varies measurably with (a, x). The standard-Borel assumptions ensures existence.

Fix σ-finite dominating measures μ^A and μ^X, assume that $\operatorname{supp}(\mu^A) = \mathcal{A}$, and write $\mu^{A,X} = \mu^A \otimes \mu^X$. We define

$$\mathcal{P} := \left\{ P \text{ a probability distribution on } \mathcal{Y} \times \mathcal{A} \times \mathcal{X} : P^{A,X} \sim \mu^{A,X} \right\}.$$

Here, $\sim$ denotes mutual absolute continuity, i.e., having the same null sets. Equivalently for any $P \in \mathcal{P}$, $P^X \sim \mu^X$ and $P^{A|X=x} \sim \mu^A$ for μ^X-almost every x. For finite categorical A, this means that, under every $P \in \mathcal{P}$, the conditional distribution of

A given $X = x$ has the same support as μ^A for μ^X-almost every x. Because $P^{A,X}$ and $\mu^{A,X}$ have the same null sets, $P^{A,X}$-almost-everywhere uniqueness of $P^{Y|A,X}$ carries over to $\mu^{A,X}$-almost-everywhere uniqueness.

For any probability distribution $P \in \mathcal{P}$, define

$$\mathcal{Q}(P) \coloneqq \{Q \in \mathcal{P} \mid Q^{Y|A,X} = P^{Y|A,X}\}.$$

Thus $\mathcal{Q}(P)$ consists of all distributions in $\mathcal{P}$ that share the same conditional distribution of Y given (A, X) as P. Here, kernel equality is understood $\mu^{A,X}$-almost everywhere.

We report the following immediate property.

Proposition 3.2.1. *The set $\{\mathcal{Q}(P) : P \in \mathcal{P}\}$ is a partition of $\mathcal{P}$.*

A sketch of the partitioning is given in Figure 3.1.

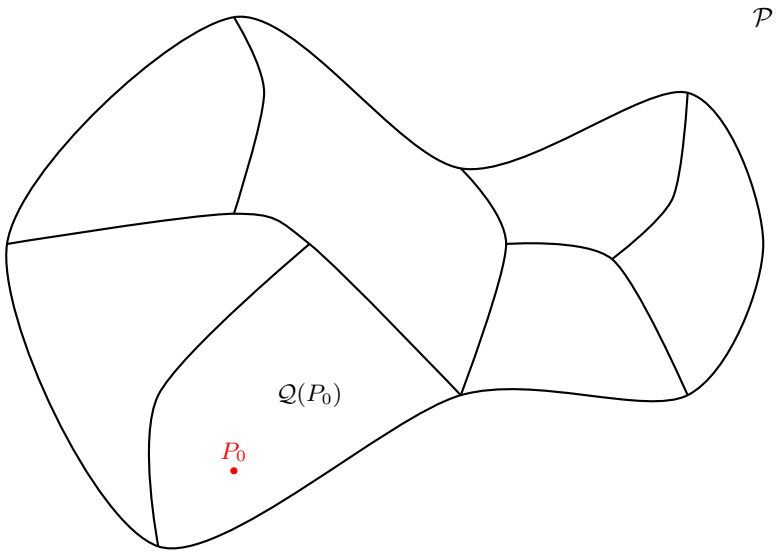


Figure 3.1: Partition of $\mathcal{P}$ into classes sharing the conditional distribution of Y given (A, X).

Let $\mathcal{G} = L^0(\mu^X)$ be the space of real-valued measurable functions of x, with functions identified when they agree μ^X-almost everywhere. A functional $\Phi : \mathcal{P} \to \mathcal{G}$ maps a distribution P to a prediction function in x.

Definition 3.2.2 (AX-invariance)**.** We say that a functional[1] $\Phi : \mathcal{P} \to \mathcal{G}$ is AX-invariant if, for every $P \in \mathcal{P}$ and every $Q', Q'' \in \mathcal{Q}(P)$,

$$\Phi_{Q'}(x) = \Phi_{Q''}(x) \qquad \text{for } \mu^X\text{-almost every } x \in \mathcal{X}.$$

[1]We use the convention that subscripts denote function evaluations, that is, $\Phi_P \coloneqq \Phi(P)$.

By construction, an AX-invariant functional is constant on every region $\mathcal{Q}(P)$ depicted in Figure 3.1. It may use the conditional distribution $P^{Y|A,X}$ but not the joint distribution of (A, X). It is natural not to use the marginal distribution of X: since the target assigns a value at each x, it should not change merely because some values of X become more or less common in the population. The definition additionally excludes both the marginal distribution of A and the conditional distribution of A given X. When the relevant densities exist, this distinction can be written as

$$p(y, a, x) = p^{Y|A,X}(y \mid a, x)p^{A|X}(a \mid x)p^{X}(x),$$

and AX-invariance allows the target to use the first factor but not the last two. The ordinary conditional mean makes the distinction concrete. Let

$$m_Q(a, x) = \mathbb{E}_Q[Y \mid A = a, X = x].$$

Whenever the expectation exists,

$$\mathbb{E}_Q[Y \mid X = x] = \int_{\mathcal{A}} m_Q(a, x)\, \mathrm{d}Q^{A|X=x}(a)$$

For $Q \in \mathcal{Q}(P)$, the function m_Q is fixed but the weights $\mathrm{d}Q^{A|X=x}(a)$ may change. The conditional-mean functional is therefore not AX-invariant in general. By contrast, replacing these weights with a prespecified reference distribution produces an AX-invariant functional; this is the starting point of Section 3.4. The next section specialises A to a protected attribute and relates this distributional requirement to causal fairness.

3.3 Causal fairness interpretation and scope of AX-invariance

Although AX-invariance is neither causal nor a fairness criterion by definition, it acquires a fairness interpretation when A is protected. In this section we make that specialisation and use structural causal models (SCMs) and do-notation (Pearl, 2009); the same ideas can be expressed with potential outcomes (Rubin, 2005).

3.3.1 Causal fairness for prediction functionals

For a fixed population distribution P, the learned prediction function and its associated predictor are

$$g_P : x \longmapsto \Phi_P(x), \qquad \widehat{Y} = g_P(X).$$

A predictor-level criterion asks whether $\hat{Y}$ is fair. It holds g_P fixed while considering interventions within a causal model.

At the functional level, the object is $\Phi : P \mapsto \Phi_P$. It describes how the prediction function depends on the supplied distribution P. Even if g_P does not take A as

an input, the functional Φ may use $P^{A|X}$ to produce it. Thus distributions P and Q with $P^{Y|A,X} = Q^{Y|A,X}$ but $P^{A|X} \neq Q^{A|X}$ may yield different functions g_P and g_Q. This is the indirect use of A considered here. A predictor-level criterion cannot detect this indirect use, since it evaluates only the resulting function g_P, not how that function depends on the population distribution used to construct it.

An SCM assigns each endogenous variable as a function of its direct causes and exogenous disturbances. For an SCM M, write P_M for the observational distribution of (Y, A, X) induced by M.

Counterfactual fairness and predictor-level path-specific criteria evaluate the fixed prediction function (Kusner et al., 2017; Wu et al., 2019). They can detect direct use of A, and may detect the use of features changed by an intervention on A. Since g_P is fixed, however, they cannot reveal whether $P^{A|X}$ affected the shape of g_P. We will study causal fairness at the level of the functional Φ.

Let $\mathcal{M}_{\text{cp}}$ contain the SCMs M over (Y, A, X) for which $P_M \in \mathcal{P}$ and (A, X) causally precede Y, meaning that the structural equations for A and X do not depend on Y. We impose no further restriction on the relation between A and X. They may be causally related, share exogenous disturbances, or form a feedback system as long as there is a unique solution. We also allow U_Y to be dependent on the disturbances generating (A, X). Figure 3.2 gives six examples.

The condition $M_0 \in \mathcal{M}_{\text{cp}}$ is natural when A and X are determined before the outcome occurs. Examples include applicant information before loan default, patient and health information before disease onset, and policy holder and car information before an accident.

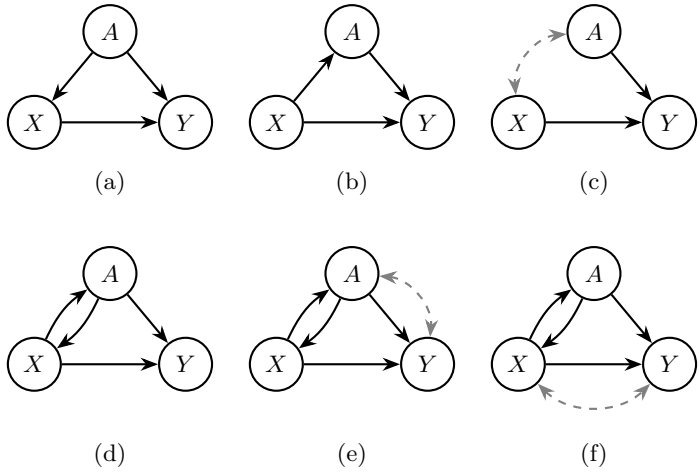


Figure 3.2: Six structures allowed by the broad causal setup. In every panel, A and X causally precede Y. Dashed grey bidirected edges represent unobserved confounding. Panels (d)–(f) allow feedback between A and X.

Write the response equation as

$$Y = f_Y(A, X, U_Y),$$

where U_Y is the response disturbance. For $a \in \mathcal{A}$, let M^a be the SCM obtained by replacing this equation by

$$Y = f_Y(a, X, U_Y)$$

and leaving the rest of the SCM unchanged. This removes the direct edge $A \to Y$ but preserves paths through X.

The next example showcases a situation where a prediction function is counterfactually fair, but the functional that produces it depends on $P^{A|X}$.

Example 3.3.1 (Predictor-level counterfactual fairness does not imply AX-invariance). Take $\mathcal{A} = \mathcal{X} = \{0, 1\}$ with counting dominating measures. Take $\mathcal{Y} = \{x + a : x, a \in \{0, 1\}\}$. For each fixed number $r \in (0, 1)$, let M_r be the acyclic SCM

$$X = \mathbb{1}\{U_X \le 1/2\}, \qquad A = \mathbb{1}\{U_A \le r\}, \qquad Y = X + A,$$

where U_X and U_A are independent $\mathrm{Unif}[0, 1]$ random variables, and write $U = (U_X, U_A)^\top$. Let $\Phi_P^{\mathrm{cm}}(x) = \mathbb{E}_P[Y \mid X = x]$. Then $P_{M_r}(A = 1 \mid X = x) = r$ and

$$P_{M_r}^{Y|A=a,X=x} = P_{M_r}^{Y|\mathrm{do}(A=a,X=x)} = \delta_{x+a},$$
$$\Phi_{P_{M_r}}^{\mathrm{cm}}(x) = x + r. \tag{3.1}$$

Here δ_z denotes the point mass at z. For each fixed r, augment M_r with the fixed prediction node

$$\widehat{Y}_r = g_r(X), \qquad g_r(x) = \Phi_{P_{M_r}}^{\mathrm{cm}}(x) = x + r,$$

and continue to use M_r for the augmented model. Write $\widehat{y}_{M_r}(u)$ for the deterministic value of the prediction node at exogenous state u. At $u = (u_X, u_A)^\top \in [0, 1]^2$,

$$\widehat{y}_{M_r}(u) = g_r\{X(u)\} = \mathbb{1}\{u_X \le 1/2\} + r.$$

For $a \in \mathcal{A}$, let $M_r^{\mathrm{do}(A=a)}$ denote the model obtained by replacing the assignment for A by the constant a. Since the assignment for X has no A input, for every u and every $a, a' \in \mathcal{A}$,

$$\widehat{y}_{M_r^{\mathrm{do}(A=a)}}(u) = \widehat{y}_{M_r}(u) = \widehat{y}_{M_r^{\mathrm{do}(A=a')}}(u). \tag{3.2}$$

Thus the fixed predictor $g_r(X)$ is counterfactually fair.[2] However, $P_{M_r}^{Y|A,X} = P_{M_{\widetilde{r}}}^{Y|A,X}$, so AX-invariance requires $g_r = g_{\widetilde{r}}$. Equation (3.1) shows that this fails whenever $r \neq \widetilde{r}$. Hence Φ^{cm} is not AX-invariant.

[2] Following Kusner et al. (2017), a fixed predictor in M is counterfactually fair if, for every $a' \in \mathcal{A}$ and $P_M^{A,X}$-almost every factual (a, x),

$$\mathcal{L}_M\left\{\widehat{y}_{M^{\mathrm{do}(A=a)}}(U) \mid A = a, X = x\right\} = \mathcal{L}_M\left\{\widehat{y}_{M^{\mathrm{do}(A=a')}}(U) \mid A = a, X = x\right\}.$$

Equation (3.2) holds pointwise in u and therefore implies this condition.

The example above is a symptom of the more general problem that a fixed prediction function does not reveal which functional produced it. Fix $r^\star \in (0,1)$ and define

$$\Psi_P(x) = (1 - r^\star)\mathbb{E}_P[Y \mid A = 0, X = x] + r^\star \mathbb{E}_P[Y \mid A = 1, X = x].$$

Because $r^\star$ does not depend on P, Ψ is AX-invariant. Yet $\Psi_{P_{M_r}}(x) = x + r^\star$ and $\Psi_{P_{M_{r^\star}}} = \Phi^{\mathrm{cm}}_{P_{M_{r^\star}}}$. Thus the same prediction function at $P_{M_{r^\star}}$ can arise from an AX-invariant functional, which does not use $P^{A,X}$, or from the non-invariant conditional-mean functional, which does use $P^{A,X}$. Predictor-level criteria cannot distinguish them.

An estimand whose value shifts when $P^{A|X}$ changes, i.e., when the protected attribute is reallocated within feature strata while $P^{Y|A,X}$ is held fixed, is hard to defend under common fairness paradigms. In the SCM setting where A and X causally precede Y, that is $M \in \mathcal{M}_{\mathrm{cp}}$, AX-invariance can be motivated as a necessary condition for an estimand Φ to be causally fair. To this end, the next lemma shows that any two probability distributions P_1, P_2 satisfying $P_1^{Y|A,X} = P_2^{Y|A,X}$ admit causal representations in $\mathcal{M}_{cp}$ with the same response assignment and response-noise distribution.

Lemma 3.3.2 (Common-response representation)**.** *For any $P_1, P_2 \in \mathcal{P}$ that satisfy $P_1^{Y|A,X} = P_2^{Y|A,X}$, there exist acyclic models $M_1, M_2 \in \mathcal{M}_{\mathrm{cp}}$, a common measurable response map*

$$f : \mathcal{A} \times \mathcal{X} \times [0,1] \longrightarrow \mathcal{Y},$$

and response disturbances $U_i^Y \sim \mathrm{Unif}[0,1]$, each independent of the exogenous variables generating (A_i, X_i), such that $P_{M_i} = P_i$ and $Y_i = f(A_i, X_i, U_i^Y)$ for $i = 1, 2$. Furthermore, for every (a, x),

$$P^{Y|X=x}_{M_1^a} = P^{Y|X=x}_{M_2^a} = P^{Y|\mathrm{do}(A=a,X=x)}_{M_1} = P^{Y|\mathrm{do}(A=a,X=x)}_{M_2}.$$

A proof is given in Appendix 3.A.

Consider the fairness notion that the functional Φ must not responds to upstream demographic structure but only on how Y arises from (A, X). Lemma 3.3.2 says that any functional-level causal fairness notion that dictates that the shape of the prediction function must be fully determined by the response assignment f and response-noise distribution $\mathcal{L}(U_Y)$, i.e. it is not affected by the mechanism that generates (A, X), must imply AX-invariance; provided its comparison class contains the models M_1 and M_2 constructed in the lemma for every response-equivalent pair of distributions.

We next make this general argument concrete through two causal fairness notions for prediction functionals: path-specific fairness and interventional fairness. Under the comparison classes stated below, each requires AX-invariance and, in specified cases under stronger causal assumptions, is equivalent to it.

3.3.2 Path-specific fairness for prediction functionals

Path-specific fairness begins by specifying which paths are unfair (Nabi and Shpitser, 2018). We designate the direct edge $A \to Y$ as unfair while paths through X are permissible. The model M^a defined earlier is the corresponding *a-fair world*: the response equation uses the fixed value a, while the rest of the SCM is unchanged.

A class $\mathcal{C} \subseteq \mathcal{M}_{\text{cp}}$ specifies the models being compared. A fairness claim about the actual population SCM M_0 requires $M_0 \in \mathcal{C}$.

Definition 3.3.3 (Weak and strong path-specific fairness)**.** Fix $\mathcal{C} \subseteq \mathcal{M}_{\text{cp}}$ with $M_0 \in \mathcal{C}$ and a functional $\Phi : \mathcal{P} \to \mathcal{G}$.

(i) The functional is *weakly path-specifically fair* for the edge $A \to Y$ with respect to $\mathcal{C}$ if, for any $M_1, M_2 \in \mathcal{C}$,

$$P_{M_1^a}^{Y|X=x} = P_{M_2^a}^{Y|X=x} \quad \text{for } \mu^{A,X}\text{-almost every } (a,x) \tag{3.3}$$

implies $\Phi_{P_{M_1}} = \Phi_{P_{M_2}}$.

(ii) Fix $a_0 \in \mathcal{A}$. The functional is *strongly path-specifically fair at* a_0 for the edge $A \to Y$ with respect to $\mathcal{C}$ if, for any $M_1, M_2 \in \mathcal{C}$,

$$P_{M_1^{a_0}}^{Y|X=x} = P_{M_2^{a_0}}^{Y|X=x} \quad \text{for } \mu^{X}\text{-almost every x}$$

implies $\Phi_{P_{M_1}} = \Phi_{P_{M_2}}$.

Path-specific fairness says that any model with the same conditional distributions of Y given X in the fair worlds must produce the same prediction function as the actual model. Within $\mathcal{C}$, weak fairness permits Φ to depend on the full family $\{P_{M^a}^{Y|X} : a \in \mathcal{A}\}$, whereas strong fairness restricts it to $P_{M^{a_0}}^{Y|X}$, the conditional distribution of Y given X in one prespecified a_0-fair world. When $\mu^A(\{a_0\}) > 0$, strong fairness is therefore the stricter requirement.

To connect these fair-world response distributions to the observational conditional distribution of Y given (A, X), introduce the response-no-confounding subclass

$$\mathcal{M}_{\text{nc}} \coloneqq \{M \in \mathcal{M}_{\text{cp}} : U_Y \perp\!\!\!\perp (A, X)\}.$$

This requires the response disturbance U_Y to be independent of (A, X), while leaving the relation between A and X unrestricted. The structures in Figure 3.2(e)–(f) therefore lie outside $\mathcal{M}_{\text{nc}}$. The common-response models constructed in Lemma 3.3.2 belong to this subclass.

Lemma 3.3.4 (No-confounding bridge)**.** *For every $M \in \mathcal{M}_{\text{nc}}$,*

$$P_{M^a}^{Y|X=x} = \mathcal{L}_M\{f_Y(a, x, U_Y)\} = P_M^{Y|\text{do}(A=a,X=x)} = P_M^{Y|A=a,X=x}, \tag{3.4}$$

for $\mu^{A,X}$-almost every (a, x).

A proof is given in Appendix 3.A. On $\mathcal{M}_{\text{nc}}$, the lemma identifies each fair-world response distribution with the corresponding $A = a$ slice of the observational conditional distribution of Y given (A, X).

Theorem 3.3.5 (Path-specific fairness and AX-invariance)**.** *Let $\Phi : \mathcal{P} \to \mathcal{G}$ be a functional.*

(i) For every $\mathcal{C} \subseteq \mathcal{M}_{\text{cp}}$ and every $a_0 \in \mathcal{A}$ satisfying $\mu^A(\{a_0\}) > 0$, strong path-specific fairness at a_0 with respect to $\mathcal{C}$ implies weak path-specific fairness with respect to $\mathcal{C}$.

(ii) For every $\mathcal{C}$ satisfying $\mathcal{M}_{\text{nc}} \subseteq \mathcal{C} \subseteq \mathcal{M}_{\text{cp}}$, weak path-specific fairness with respect to $\mathcal{C}$ implies AX-invariance. Moreover, for every $a_0 \in \mathcal{A}$, strong path-specific fairness at a_0 with respect to such a class also implies AX-invariance.

(iii) Under $M_0 \in \mathcal{M}_{\text{nc}}$, the functional is AX-invariant if and only if it is weakly path-specifically fair with respect to $\mathcal{M}_{\text{nc}}$.

Part (ii) is the practically useful one for our purposes. Its contrapositive shows that an estimand failing AX-invariance cannot be weakly or strongly path-specifically fair with respect to $\mathcal{M}_{\text{cp}}$. The inclusion $\mathcal{M}_{\text{nc}} \subseteq \mathcal{C}$ in (ii) ensures that the comparison class contains the common-response models from Lemma 3.3.2. The necessity of part (ii) does not require $M_0 \in \mathcal{M}_{\text{nc}}$; it also applies when $M_0 \in \mathcal{C} \setminus \mathcal{M}_{\text{nc}}$. Applying the equivalence in part (iii) to the actual population does require $M_0 \in \mathcal{M}_{\text{nc}}$. Without response no-confounding, the fair-world response distributions need not be identified from the observational distribution. Consequently, AX-invariance is necessary for path-specific fairness on the classes in part (ii), but does not by itself establish it.

3.3.3 Interventional fairness for prediction functionals

A second approach defines fairness through the response distributions under joint interventions on (A, X) (Kilbertus et al., 2017). Let $\mathcal{M}_{\text{va}} \subseteq \mathcal{M}_{\text{cp}}$ contain the SCMs for which

$$P_M^{Y|A=a,X=x} = P_M^{Y|\text{do}(A=a,X=x)} \quad \text{for } \mu^{A,X}\text{-almost every } (a, x). \tag{3.5}$$

Thus conditioning on (A, X) identifies the response under their joint intervention. Response-confounded models, such as those in Figure 3.2(e)–(f), need not satisfy this equality. Response no-confounding does imply it, so $\mathcal{M}_{\text{nc}} \subseteq \mathcal{M}_{\text{va}}$. Assuming the true causal model M_0 over (A, X, Y) satisfies $M_0 \in \mathcal{M}_{\text{va}}$, the equivalence class $\mathcal{Q}(P_{M_0})$ has a clean causal interpretation: it is the set of all interventional distributions obtained by intervening on (A, X) and setting its joint distribution to a law equivalent to $\mu^{A,X}$. In particular, AX-invariant functionals depend on M only through the structural mechanism by which Y is generated from (A, X), and ignore the mechanism by which (A, X) itself comes about.

Definition 3.3.6 (Interventional fairness for functionals)**.** A functional $\Phi : \mathcal{P} \to \mathcal{G}$ is *interventionally fair* with respect to a class $\mathcal{C}$ if $M_0 \in \mathcal{C}$, for any $M_1, M_2 \in \mathcal{C}$,

$$P_{M_1}^{Y|\mathrm{do}(A=a,X=x)} = P_{M_2}^{Y|\mathrm{do}(A=a,X=x)} \quad \text{for } \mu^{A,X}\text{-almost every } (a,x) \tag{3.6}$$

implies $\Phi_{P_{M_1}} = \Phi_{P_{M_2}}$.

Theorem 3.3.7 (Interventional fairness and AX-invariance)**.** *Let $\Phi : \mathcal{P} \to \mathcal{G}$ be a functional.*

(i) For every $\mathcal{C}$ satisfying $\mathcal{M}_{\mathrm{nc}} \subseteq \mathcal{C} \subseteq \mathcal{M}_{\mathrm{cp}}$, interventional fairness with respect to $\mathcal{C}$ implies AX-invariance.

(ii) Under $M_0 \in \mathcal{M}_{\mathrm{va}}$, the functional is AX-invariant if and only if it is interventionally fair with respect to $\mathcal{M}_{\mathrm{va}}$.

As in the path-specific fairness case, part (i) says that an estimand failing AX-invariance cannot be interventional fair with respect to $\mathcal{M}_{cp}$. The equivalence in part (ii) requires for the actual population $M_0 \in \mathcal{M}_{\mathrm{va}}$. Since $\mathcal{M}_{\mathrm{nc}} \subseteq \mathcal{M}_{\mathrm{va}}$, membership in $\mathcal{M}_{\mathrm{va}}$ is weaker than response no-confounding. When the comparison class is $\mathcal{M}_{\mathrm{nc}}$, interventional fairness and weak path-specific fairness both coincide with AX-invariance. The two notions remain causally distinct: one compares joint interventions, while the other compares fair worlds obtained by removing the direct edge. Proofs are in Appendix 3.A.

3.4 Building AX-invariant prediction targets

This section constructs prediction functionals from the conditional distribution $P^{Y|A,X}$ without using the population-specific distribution $P^{A,X}$. To aggregate prediction risks across values of A, fix a probability kernel $x \mapsto \rho_x$ from $\mathcal{X}$ to $\mathcal{A}$. The kernel is fixed in advance and does not change with P; at each x, ρ_x supplies the reference weights over $\mathcal{A}$. We assume that there is a Borel set $\mathcal{X}_\rho \subseteq \mathcal{X}$ with $\mu^X(\mathcal{X} \setminus \mathcal{X}_\rho) = 0$ such that, for every $x \in \mathcal{X}_\rho$,

$$\rho_x \ll \mu^A \qquad \text{and} \qquad \mathrm{supp}(\rho_x) = \mathcal{A}. \tag{3.7}$$

The two conditions in (3.7) serve different purposes. By Fubini's theorem, domination by μ^A ensures that replacing $P^{Y|A,X}$ by a $\mu^{A,X}$-almost-everywhere equal version leaves the aggregations below unchanged for μ^X-almost every x. The resulting targets are therefore well defined as elements of $L^0(\mu^X)$. Full support ensures that no non-empty open subset of $\mathcal{A}$ is ignored and, for continuous risks, identifies the ρ_x-essential supremum with the ordinary maximum.

When $\mathcal{A} = \{a_1, \ldots, a_K\}$ and μ^A is counting measure, domination is automatic and full support is equivalent to $\rho_x(a_j) > 0$ for every j; the uniform weights $\rho_x(a_j) = 1/K$

give a simple x-independent choice. If ρ_x varies with x, the reference weighting itself varies across feature values; this dependence is part of the target specification.

For a loss $\ell : \mathcal{Y} \times \mathbb{R} \to [0, \infty]$, define the risk

$$r_P^\ell(a, x, w) = \int_{\mathcal{Y}} \ell(y, w) \, \mathrm{d}P^{Y|A=a,X=x}(y).$$

For a candidate prediction w, $r_P^\ell(a, x, w)$ is its expected loss within state a at feature value x. For fixed (x, w), write $f(a) = r_P^\ell(a, x, w)$. We aggregate these state-specific risks using a parameter $\beta \in [0, \infty]$:

$$\mathcal{R}_{\beta,\rho,x}(f) = \begin{cases} \displaystyle\int_{\mathcal{A}} f(a) \, \rho_x(\mathrm{d}a), & \beta = 0, \\ \displaystyle\frac{1}{\beta} \log \int_{\mathcal{A}} \exp\{\beta f(a)\} \, \rho_x(\mathrm{d}a), & 0 < \beta < \infty, \\ \displaystyle\operatorname*{ess\,sup}_{\rho_x} f, & \beta = \infty. \end{cases}$$

For bounded Borel functions f and g, all three aggregators are monotone and translation equivariant:

$$f \le g \ \rho_x\text{-a.e.} \implies \mathcal{R}_{\beta,\rho,x}(f) \le \mathcal{R}_{\beta,\rho,x}(g), \quad \mathcal{R}_{\beta,\rho,x}(f + c) = \mathcal{R}_{\beta,\rho,x}(f) + c, \tag{3.8}$$

for every $c \in \mathbb{R}$ and $\beta \in [0, \infty]$. In particular, each aggregator maps a constant function to that constant. Setting $\beta = 0$ yields ordinary averaging under the reference distribution. As β increases, larger risks receive more influence. At $\beta = \infty$, the objective retains only the largest state-specific risk and produces the worst-case target defined below. The essential supremum is the largest value after sets receiving no reference mass are ignored. If f is continuous, compactness of $\mathcal{A}$ and the full support of ρ_x give

$$\operatorname*{ess\,sup}_{\rho_x} f = \max_{a \in \mathcal{A}} f(a).$$

We now focus on squared loss, for which the minimiser is unique.

3.4.1 A unified squared-error target

For the remainder of this paper, we consider the squared loss $\ell(y, w) = (y - w)^2$ with corresponding risk

$$r_P(a, x, w) = \mathbb{E}_P[(Y - w)^2 \mid A = a, X = x].$$

Let $m_P(a, x)$ and $v_P(a, x)$ denote the conditional mean and variance of the outcome:

$$\begin{aligned} m_P(a, x) &= \mathbb{E}_P[Y \mid A = a, X = x] = \int y \, \mathrm{d}P^{Y|A=a,X=x}(y), \\ v_P(a, x) &= \mathrm{Var}_P(Y \mid A = a, X = x) = \int \{y - m_P(a, x)\}^2 \, \mathrm{d}P^{Y|A=a,X=x}(y). \end{aligned} \tag{3.9}$$

The identities in (3.9) are understood up to $\mu^{A,X}$-almost-everywhere equality. We assume that the two moments admit jointly Borel-measurable, real-valued versions, again denoted by m_P and v_P, and that there is a Borel set $\widetilde{\mathcal{X}}_P \subseteq \mathcal{X}$ with $\mu^X(\mathcal{X}\backslash\widetilde{\mathcal{X}}_P) = 0$ such that, for every $x \in \widetilde{\mathcal{X}}_P$, the maps

$$a \longmapsto m_P(a,x) \quad \text{and} \quad a \longmapsto v_P(a,x)$$

are continuous and $v_P(a,x) \geq 0$ on $\mathcal{A}$. For finite categorical A, continuity in a is automatic, so this condition only requires measurable conditional means and variances that are finite for almost every x; no continuity or smoothness in x is required. We have,

$$r_P(a,x,w) = v_P(a,x) + \{m_P(a,x) - w\}^2, \tag{3.10}$$

for $\mu^{A,X}$-almost every (a,x). Furthermore let E_P be the measurable $\mu^{A,X}$-null set on which either identity fails. Then $\mathcal{G}_P = \{x \in \mathcal{X} : \mu^A((E_P)_x) = 0\}$ is Borel and μ^X-conull. Define $\mathcal{X}_P = \widetilde{\mathcal{X}}_P \cap \mathcal{G}_P$. Then or every $x \in \mathcal{X}_P$, equation (3.10) also holds for μ^A-almost every a, simultaneously for every $w \in \mathbb{R}$. Define the Borel, μ^X-conull set

$$\mathcal{X}_{P,\rho} = \mathcal{X}_P \cap \mathcal{X}_\rho.$$

For $x \in \mathcal{X}_{P,\rho}$, define

$$\Phi_P^{\beta,\rho}(x) = \operatorname*{arg\,min}_{w\in\mathbb{R}} \mathcal{R}_{\beta,\rho,x}\big(r_P(\,\cdot\,,x,w)\big), \qquad \beta \in [0,\infty]. \tag{3.11}$$

The theorem below verifies that the minimiser is well defined and measurable. Note that $\Phi_P^{\beta,\rho}$ is defined only on the μ^X-conull set $\mathcal{X}_{P,\rho}$, which nevertheless determines a unique element of $L^0(\mu^X)$.

Lemma 3.4.1 (KL representation)**.** *For every $P \in \mathcal{P}$, $x \in \mathcal{X}_{P,\rho}$, and $0 < \beta < \infty$, the target has the pointwise representation*

$$\Phi_P^{\beta,\rho}(x) = \operatorname*{arg\,min}_{w\in\mathbb{R}} \sup_{\substack{\pi\in\mathfrak{P}(\mathcal{A}) \\ \pi\ll\rho_x}} \left\{ \int_{\mathcal{A}} r_P(a,x,w)\,\pi(\mathrm{d}a) - \frac{1}{\beta}\,\mathrm{KL}(\pi\|\rho_x) \right\}. \tag{3.12}$$

Here $\mathfrak{P}(\mathcal{A})$ denotes the Borel probability measures on $\mathcal{A}$, and

$$\mathrm{KL}(\pi\|\rho_x) = \int_{\mathcal{A}} \log\left(\frac{\mathrm{d}\pi}{\mathrm{d}\rho_x}\right) \mathrm{d}\pi$$

when $\pi \ll \rho_x$, with value $+\infty$ otherwise.

The proof is given in Appendix 3.A. Here π provides alternative weights over the states of A, while the KL term penalises departure from the prespecified reference weights ρ_x. Increasing β weakens this penalty and allows more adverse reweighting, connecting reference averaging to the worst-case target (Kuhn et al., 2025).

Theorem 3.4.2. *Assume that the reference kernel ρ satisfies* (3.7)*. For every $P \in \mathcal{P}$, assume further that the conditional mean and variance surfaces m_P and v_P defined in* (3.9) *admit jointly Borel-measurable, real-valued versions such that, on a Borel μ^X-conull set of x, both are continuous in a, with $v_P(a,x) \geq 0$ for every $a \in \mathcal{A}$. Then the following statements hold.*

(i) For every $\beta \in [0,\infty]$, the minimiser in (3.11) *exists and is unique for every $x \in \mathcal{X}_{P,\rho}$. The target is measurable on $\mathcal{X}_{P,\rho}$, its element of $L^0(\mu^X)$ is independent of the admissible conditional kernel and moment-surface versions, and the functional $P \mapsto \Phi_P^{\beta,\rho}$ is AX-invariant.*

(ii) For every $\beta \in [0,\infty]$ and every $x \in \mathcal{X}_{P,\rho}$, the prediction lies between the smallest and largest state-specific conditional means:

$$\min_{a\in\mathcal{A}} m_P(a,x) \leq \Phi_P^{\beta,\rho}(x) \leq \max_{a\in\mathcal{A}} m_P(a,x). \tag{3.13}$$

(iii) For every $x \in \mathcal{X}_{P,\rho}$, the map $\beta \mapsto \Phi_P^{\beta,\rho}(x)$ is continuous on the extended interval $[0,\infty]$, including the limits as $\beta \downarrow 0$ and $\beta \to \infty$.

(iv) For every $x \in \mathcal{X}_{P,\rho}$, at $\beta = 0$ the solution is

$$\Phi_P^{0,\rho}(x) = \int_{\mathcal{A}} m_P(a,x)\, \rho_x(\mathrm{d}a). \tag{3.14}$$

For $0 < \beta < \infty$, the solution is the unique fixed point in the interval in (3.13) *satisfying*

$$\Phi_P^{\beta,\rho}(x) = \frac{\int_{\mathcal{A}} m_P(a,x) \exp\left\{\beta r_P\left(a,x,\Phi_P^{\beta,\rho}(x)\right)\right\} \rho_x(\mathrm{d}a)}{\int_{\mathcal{A}} \exp\left\{\beta r_P\left(a,x,\Phi_P^{\beta,\rho}(x)\right)\right\} \rho_x(\mathrm{d}a)}. \tag{3.15}$$

(v) For every $x \in \mathcal{X}_{P,\rho}$, the worst-case target ($\beta = \infty$) is

$$\Phi_P^{\infty,\rho}(x) = \arg\min_{w\in\mathbb{R}} \max_{a\in\mathcal{A}} \left[v_P(a,x) + \{m_P(a,x) - w\}^2\right]. \tag{3.16}$$

Because ρ_x has full support, the right-hand side does not depend on the particular reference measure; we therefore also write it as $\Phi_P^{\infty}(x)$.

We now look at some special cases.

Corollary 3.4.3 (Squared-error special cases)**.** *Fix $x \in \mathcal{X}_{P,\rho}$. For parts (ii)–(iv) below, assume that $\mathcal{A}$ has two elements. Label them a_- and a_+ so that $m_P(a_-,x) \leq m_P(a_+,x)$, and abbreviate*

$$m_- = m_P(a_-,x), \quad m_+ = m_P(a_+,x), \qquad v_- = v_P(a_-,x), \quad v_+ = v_P(a_+,x).$$

(i) If $v_P(a,x)$ does not depend on a, then

$$\Phi_P^{\infty}(x) = \frac{1}{2}\left\{\min_{a\in\mathcal{A}} m_P(a,x) + \max_{a\in\mathcal{A}} m_P(a,x)\right\}.$$

(ii) If $m_- < m_+$, define the equal-risk value

$$w_{\mathrm{eq}} = \frac{m_- + m_+}{2} + \frac{v_+ - v_-}{2(m_+ - m_-)}.$$

Then

$$\Phi_P^{\infty}(x) = \begin{cases} m_-, & w_{\mathrm{eq}} < m_-, \\ w_{\mathrm{eq}}, & m_- \le w_{\mathrm{eq}} \le m_+, \\ m_+, & w_{\mathrm{eq}} > m_+. \end{cases} \tag{3.17}$$

(iii) If $m_- = m_+ = m_0$, then $\Phi_P^{\beta,\rho}(x) = m_0$ for every $\beta \in [0,\infty]$, irrespective of v_- and v_+.

(iv) If $\rho_x(a_-) = \rho_x(a_+) = 1/2$ and $v_- = v_+$, then the complete robustness path is constant:

$$\Phi_P^{\beta,\rho}(x) = \frac{m_- + m_+}{2}, \qquad \beta \in [0,\infty].$$

The target also simplifies when the conditional mean is additive and the conditional variance does not vary across states of A. If

$$m_P(a,x) = g_A(a) + g_X(x), \qquad v_P(a,x) = v_X(x),$$

then setting $w = g_X(x) + u$ gives

$$r_P(a,x,g_X(x)+u) = v_X(x) + \{g_A(a) - u\}^2.$$

Using (3.8) therefore yields

$$\Phi_P^{\beta,\rho}(x) = g_X(x) + \operatorname*{arg\,min}_{u\in\mathbb{R}} \mathcal{R}_{\beta,\rho,x}\big(\{g_A(\,\cdot\,) - u\}^2\big).$$

In particular, if ρ_x does not depend on x, the second term is a fixed, reference-dependent offset.

3.4.2 Computation when the attribute is finite

We now provide an algorithm to calculate $\Phi_P^{\beta,\rho}(x)$ pointwise in x when $\mathcal{A} = \{a_1,\dots,a_K\}$. Fix $x \in \mathcal{X}_{P,\rho}$ and abbreviate

$$\rho_j = \rho_x(a_j), \qquad m_j = m_P(a_j,x), \qquad v_j = v_P(a_j,x), \qquad r_j(w) = v_j + (m_j - w)^2.$$

Here $\rho_j > 0$ and $\sum_j \rho_j = 1$. At the reference endpoint $\beta = 0$, the objective in (3.11) is $F_0(w) = \sum_{j=1}^K \rho_j r_j(w)$, with unique minimiser $\bar{m}_\rho = \sum_{j=1}^K \rho_j m_j$. For $0 < \beta < \infty$, the objective in (3.11) becomes

$$F_\beta(w) = \frac{1}{\beta} \log \sum_{j=1}^K \rho_j \exp\{\beta r_j(w)\}.$$

At a candidate prediction w, define the weights

$$q_j(w) = \frac{\rho_j \exp\{\beta r_j(w)\}}{\sum_{k=1}^K \rho_k \exp\{\beta r_k(w)\}}.$$

Let

$$\bar{m}_q(w) = \sum_{j=1}^K q_j(w) m_j, \qquad \mathrm{Var}_{q(w)}(m) = \sum_{j=1}^K q_j(w)\{m_j - \bar{m}_q(w)\}^2.$$

Thus $\mathrm{Var}_{q(w)}(m)$ is the weighted variance of the values $m_1, \ldots, m_K$ under the probabilities $q_1(w), \ldots, q_K(w)$. Define the first-order residual $H_\beta(w) = w - \bar{m}_q(w)$. Direct differentiation gives

$$F'_\beta(w) = 2H_\beta(w), \qquad F''_\beta(w) = 2 + 4\beta \,\mathrm{Var}_{q(w)}(m) > 0. \tag{3.18}$$

The positive curvature makes $H_\beta = F'_\beta/2$ strictly increasing. Since $\bar{m}_q(w)$ lies between $\min_j m_j$ and $\max_j m_j$, H_β has opposite weak signs at the two endpoints. Bisection therefore retains the unique zero and halves its bracket at each step; a width of at most $2\,\mathrm{tol}$ gives midpoint error at most tol.

Suppose that a twice differentiable f satisfies $0 < \mu \le f''(w) \le L$ for every w. Then f has a unique minimiser w^*. For the gradient-descent update $w_{t+1} = w_t - \alpha f'(w_t)$ with $\alpha = 2/(\mu + L)$, the mean value theorem gives, for some ξ_t between w_t and w^*,

$$w_{t+1} - w^* = \{1 - \alpha f''(\xi_t)\}(w_t - w^*), \qquad |w_{t+1} - w^*| \le \frac{L - \mu}{L + \mu} |w_t - w^*|.$$

The factor is smaller than one, so the iterates converge to w^* from any starting value.

For F_β, put $D = \max_j m_j - \min_j m_j$. Since $\mathrm{Var}_{q(w)}(m) = \min_c \sum_{j=1}^K q_j(w)\{m_j - c\}^2$, comparison with c being the midpoint gives $\mathrm{Var}_{q(w)}(m) \le D^2/4$. By (3.18), $2 \le F''_\beta(w) \le 2 + \beta D^2$. Thus the general result with $\mu = 2$ and $L = 2 + \beta D^2$ gives the algorithmic step $\alpha_\beta = 2/(4 + \beta D^2)$ and proves convergence.

Let w^*_β denote the minimiser. The gradient method stops at the first iterate w_t satisfying $|H_\beta(w_t)| \le \mathrm{tol}$. Since $H_\beta(w^*_\beta) = 0$ and $H'_\beta(w) \ge 1$, the mean value theorem gives $|w_t - w^*_\beta| \le |H_\beta(w_t)|$. The returned iterate is therefore within tol of the minimiser.

Algorithm 3.1 Finite-state computation of the finite-β target; each iteration costs $O(K)$.

input m_j, v_j, $\rho_j > 0$ for $j = 1, \dots, K$ with $\sum_j \rho_j = 1$; $\beta \in [0, \infty)$; tolerance tol > 0; method $\mathsf{M} \in \{\textsc{Bisection}, \textsc{Gradient}\}$

output Approximation $\widehat{w}$ to $\Phi_P^{\beta,\rho}(x)$

$l \leftarrow \min_j m_j$; $u \leftarrow \max_j m_j$; $w \leftarrow \sum_{j=1}^K \rho_j m_j$
if $l = u$ or $\beta = 0$ **then**
  return w
end if
For any z, compute $H_\beta(z)$ stably as follows:
  $r_j \leftarrow v_j + (m_j - z)^2$; $r_{\max} \leftarrow \max_j r_j$
  $s_j \leftarrow \log \rho_j + \beta(r_j - r_{\max})$; $c \leftarrow \max_j s_j$
  $q_j \leftarrow \exp(s_j - c) \big/ \sum_{k=1}^K \exp(s_k - c)$; $H_\beta(z) \leftarrow z - \sum_{j=1}^K q_j m_j$
if $\mathsf{M} = \textsc{Bisection}$ **then**
  while $u - l > 2\,\text{tol}$ **do**
    $w \leftarrow (l + u)/2$; compute $H_\beta(w)$
    if $H_\beta(w) \le 0$ **then**
      $l \leftarrow w$
    else
      $u \leftarrow w$
    end if
  end while
  return $(l + u)/2$
else
  $\alpha_\beta \leftarrow 2/\{4 + \beta(u - l)^2\}$
  Compute $H_\beta(w)$
  while $|H_\beta(w)| > \text{tol}$ **do**
    $w \leftarrow w - 2\alpha_\beta H_\beta(w)$; compute $H_\beta(w)$
  end while
  return w
end if

Writing $\varepsilon = 1/\beta$ shows exactly what a finite-temperature calculation returns:

$$\varepsilon \log \sum_{j=1}^K \rho_j \exp\{r_j(w)/\varepsilon\} = F_{1/\varepsilon}(w).$$

It therefore computes the finite-β target. If the worst-case target is desired, Theorem 3.4.2(iii) gives

$$\Phi_P^{1/\varepsilon,\rho}(x) \longrightarrow \Phi_P^{\infty}(x) \qquad \text{as } \varepsilon \downarrow 0.$$

Thus the exact finite-β target approaches the worst-case target as $\beta = 1/\varepsilon$ increases. Both branches of Algorithm 3.1 return a value within tol of the exact finite-β target, so approximating the worst-case target requires both a large β and a sufficiently small optimisation tolerance.

3.5 AX-invariance from data

By definition, a functional Φ is AX-invariant if

$$\Phi_{Q'} = \Phi_{Q''} \quad \text{for every } P \in \mathcal{P} \text{ and every } Q', Q'' \in \mathcal{Q}(P).$$

No empirical procedure can verify all of these equalities. It can, however, falsify them: one pair $Q', Q'' \in \mathcal{Q}(P_0)$ for which the two targets differ is enough to reject AX-invariance.

Consider a comparison that removes the association between A and X. Fix a reference distribution $u^A \sim \mu^A$; for finite categorical A, we take u^A to be uniform. Define the dependence removing shift

$$\tau(P)(\mathrm{d}y, \mathrm{d}a, \mathrm{d}x) = P^{Y|A=a,X=x}(\mathrm{d}y)\, u^A(\mathrm{d}a)\, P^X(\mathrm{d}x).$$

Thus $\tau(P)$ retains P^X and $P^{Y|A,X}$ but replaces $P^{A|X}$ by u^A. Since $P^X \sim \mu^X$ and $u^A \sim \mu^A$, we have $\tau(P) \in \mathcal{Q}(P)$. For the observed population P_0 we then have

$$\Phi_{P_0} - \Phi_{\tau(P_0)} \neq 0 \quad \text{in } L^0(\mu^X)$$

falsifies AX-invariance.

The exact resampling scheme below need not reproduce $\tau(P_0)$ exactly. Instead, it induces a distribution $Q_{n,m}$ that may depend on the original sample size n and the resample size m. Theorem 3.5.1 shows that $Q_{n,m} \in \mathcal{Q}(P_0)$ for every $m \leq n$, which is all the falsification argument requires. We use this induced distribution for two separate tasks: descriptively ranking prediction functionals and testing a prespecified necessary implication of AX-invariance.

3.5.1 Exact distinct replacement resampling

Let $P_0 \in \mathcal{P}$, and let $\mathcal{D}_n = (Z_i)_{i=1}^n$, with $Z_i = (Y_i, A_i, X_i)$, be an i.i.d. sample from P_0. Write $\mathcal{Z} = \mathcal{Y} \times \mathcal{A} \times \mathcal{X}$. For $z \in \mathcal{Z}$, let δ_z denote the Dirac probability measure at z. Write $P_n = n^{-1} \sum_{i=1}^n \delta_{Z_i}$ and $W_i = (A_i, X_i)$. Let $u = \mathrm{d}u^A / \mathrm{d}\mu^A$. Let $\mathcal{V}$ be a random element independent of $\mathcal{D}_n$, and let $\widehat{p}(a \mid x)$ be a $\sigma(\mathcal{V})$-measurable conditional density estimator of

$$p_0(a \mid x) = \frac{\mathrm{d}P_0^{A|X=x}}{\mathrm{d}\mu^A}(a).$$

Assign observation Z_i the weight

$$\omega_{n,i} = \frac{u(A_i)}{\widehat{p}(A_i \mid X_i)}.$$

For uniform categorical A, this is the inverse-propensity weight up to a common factor. We assume throughout that the weights are finite, strictly positive and measurable with respect to $\mathcal{H}_n = \sigma(W_{1:n}, \mathcal{V})$. Define $\mathcal{G}_n = \sigma(\mathcal{D}_n, \mathcal{V}) = \sigma(Y_{1:n}, \mathcal{H}_n)$.

For $1 \le m \le n$, exact distinct replacement sampling (DRPL) selects a distinct tuple $(i_1, \dots, i_m)$. The selection is outcome-blind in the precise sense that

$$\mathcal{L}(i_1, \dots, i_m \mid \mathcal{G}_n) = \mathcal{L}(i_1, \dots, i_m \mid \mathcal{H}_n) \quad \text{almost surely,} \tag{3.19}$$

and the latter conditional law assigns each distinct tuple the probability

$$w^{\mathtt{DRPL}}_{(i_1,\dots,i_m)} = \frac{\prod_{\ell=1}^{m} \omega_{n,i_\ell}}{\displaystyle\sum_{\substack{(j_1,\dots,j_m)\in\{1,\dots,n\}^m \\ j_1,\dots,j_m \text{ distinct}}} \prod_{\ell=1}^{m} \omega_{n,j_\ell}}. \tag{3.20}$$

One exact implementation described by Thams et al. (2023) is *REPL-rejection*: propose an entire m-tuple by weighted sampling with replacement and accept it only if all m indices are distinct. Conditional on $\mathcal{V}$, $(Z_{i_1}, \dots, Z_{i_m})$ is exchangeable but need not be independent.

Given the selection of the distinct tuple $(i_1, \dots, i_m)$, the empirical distribution supplied to the learner is

$$\widehat{Q}_{n,m} = \frac{1}{m} \sum_{\ell=1}^{m} \delta_{Z_{i_\ell}}.$$

For $j \in \{1, \dots, n\}$, let $S_{n,m,j} = \mathbb{1}\{j \in \{i_1, \dots, i_m\}\}$. Because of (3.19),

$$\pi_{n,m,j} = \Pr(S_{n,m,j} = 1 \mid \mathcal{H}_n) = \Pr(S_{n,m,j} = 1 \mid \mathcal{G}_n).$$

Hence

$$\widetilde{Q}_{n,m} = \mathbb{E}[\widehat{Q}_{n,m} \mid \mathcal{G}_n] = \frac{1}{m} \sum_{j=1}^{n} \pi_{n,m,j} \delta_{Z_j}.$$

For every $B \in \mathcal{B}(\mathcal{Z})$, where $\mathcal{B}(\mathcal{Z})$ is the Borel sigma-field on $\mathcal{Z}$, define

$$Q_{n,m}(B) := \mathbb{E}\{\widehat{Q}_{n,m}(B) \mid \mathcal{V}\} = \mathbb{E}\{\widetilde{Q}_{n,m}(B) \mid \mathcal{V}\} = \Pr(Z_{i_\ell} \in B \mid \mathcal{V}),$$

for $\ell = 1, \dots, m$. The second equality follows from iterated conditioning, and the third from exchangeability. Thus $Q_{n,m}$ is the conditional law of Z_{i_ℓ} given $\mathcal{V}$, for any $\ell = 1, \dots, m$.

Theorem 3.5.1. *For every $n \ge 1$ and $1 \le m \le n$, almost surely,*

$$Q^{Y|A,X}_{n,m} = P_0^{Y|A,X}, \qquad Q^{A,X}_{n,m} \sim P_0^{A,X}.$$

Hence $Q_{n,m} \in \mathcal{Q}(P_0)$. Moreover, $\widehat{Q}_{n,n} = P_n$.

Consequently, if Φ is AX-invariant, then, almost surely,

$$\Phi_{Q_{n,m}} = \Phi_{P_0} \quad \text{in } L^0(\mu^X) \qquad \text{for every } 1 \le m \le n.$$

Unlike the procedures studied by Thams et al. (2023), our diagnostic discussed below does not require the joint law of $(Z_{i_1}, \dots, Z_{i_m})$ to converge to $\tau(P_0)^{\otimes m}$. In particular, we do not impose the restriction $m = o(\sqrt{n})$.

Proposition 3.5.2. *For $m < n$ and $B \in \mathcal{B}(\mathcal{Z})$, define*

$$R_{n,m}(B) := \mathbb{E}\left[\frac{1}{n-m}\sum_{i=1}^{n}(1 - S_{n,m,i})\mathbb{1}\{Z_i \in B\} \;\middle|\; \mathcal{V}\right].$$

Then, almost surely, $R_{n,m} \in \mathcal{Q}(P_0)$ and

$$P_0 = \frac{m}{n}Q_{n,m} + \left(1 - \frac{m}{n}\right)R_{n,m}. \tag{3.21}$$

Consequently,

$$d_{\mathrm{TV}}(Q_{n,m}, P_0) \le 1 - \frac{m}{n}. \tag{3.22}$$

Proposition 3.5.2 is proved in Appendix 3.A. The proposition explains how the exact-DRPL target can vary with m. The restriction $m < n$ is needed because $R_{n,m}$ is not defined at $m = n$. At that endpoint, $\widehat{Q}_{n,n} = P_n$, whereas $Q_{n,n} = \mathbb{E}[P_n \mid \mathcal{V}] = P_0$. Increasing m increases the number of distinct observations used to fit $\widehat{\Phi}_{\widehat{Q}_{n,m}}$ while potentially changing $Q_{n,m}$. Its effect on power need not be monotone. Consistency of $\widehat{p}$ is not needed to ensure $Q_{n,m} \in \mathcal{Q}(P_0)$, but the weights determine which member $Q_{n,m}$ is selected.

3.5.2 Ranking prediction functionals

For a given m, fit $\widehat{\Phi}_{P_n}$ using $\mathcal{D}_n$ and fit $\widehat{\Phi}_{\widehat{Q}_{n,m}}$ using the selected observations $(Z_{i\ell})_{\ell=1}^{m}$. The corresponding population and fitted contrasts are

$$D_{n,m} = \Phi_{P_0} - \Phi_{Q_{n,m}}, \qquad \widehat{D}_{n,m} = \widehat{\Phi}_{P_n} - \widehat{\Phi}_{\widehat{Q}_{n,m}}.$$

On held-out covariates $X_1^{\mathrm{e}}, \ldots, X_r^{\mathrm{e}} \overset{\text{i.i.d.}}{\sim} P_0^X$ that are independent of both fits, the empirical ranking criterion is

$$\frac{1}{r}\sum_{j=1}^{r}\widehat{D}_{n,m}(X_j^{\mathrm{e}})^2. \tag{3.23}$$

Conditional on the two fitted functions, this criterion estimates $\mathbb{E}_{P_0}\{\widehat{D}_{n,m}(X)^2\}$. The corresponding target based on the population functionals is $\mathbb{E}_{P_0}[\{0 - D_{n,m}(X)\}^2] = \mathbb{E}_{P_0}\{D_{n,m}(X)^2\}$, which measures the $L^2(P_0^X)$ distance from zero.

The ranking criterion also reflects how the functionals are fitted. To describe the fitted procedure over repeated training samples, let $\mathcal{T}$ collect all randomness in the auxiliary construction, training data, selected resample, and fitting algorithm, and write $\widehat{D}_{n,m,\mathcal{T}}$ for the resulting fitted contrast. Let $X \sim P_0^X$ be independent of $\mathcal{T}$, assume that $\widehat{D}_{n,m,\mathcal{T}}(X)$ is square integrable, and define the systematic fitted contrast

$$\mu_{n,m}(x) = \mathbb{E}_{\mathcal{T}}\{\widehat{D}_{n,m,\mathcal{T}}(x)\}, \qquad P_0^X\text{-almost every } x.$$

The expected fitted ranking score has the three-way decomposition

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\mathbb{E}_{P_0}\{\widehat{D}_{n,m,\mathcal{T}}(X)^2\} = \{\mathbb{E}_{P_0}\mu_{n,m}(X)\}^2 \\ + \operatorname{var}_{P_0}\{\mu_{n,m}(X)\} \\ + \mathbb{E}_{P_0}\left[\operatorname{var}_{\mathcal{T}}\{\widehat{D}_{n,m,\mathcal{T}}(X)\}\right].\end{aligned} \tag{3.24}$$

The first term is the squared average systematic shift from zero, and the second is the variance across covariate values of that systematic shift. Both terms therefore describe the systematic shift from zero. Their sum,

$$B_{n,m}^2 := \mathbb{E}_{P_0}\{\mu_{n,m}(X)^2\} = \{\mathbb{E}_{P_0}\mu_{n,m}(X)\}^2 + \operatorname{var}_{P_0}\{\mu_{n,m}(X)\},$$

is the integrated squared bias relative to the zero function. By contrast, the third term,

$$V_{n,m} := \mathbb{E}_{P_0}\left[\operatorname{var}_{\mathcal{T}}\{\widehat{D}_{n,m,\mathcal{T}}(X)\}\right],$$

is the fitting variability at a given covariate value, averaged over P_0^X, or the integrated fitting variance. Hence the expected fitted score is $B_{n,m}^2 + V_{n,m}$. This separation shows how much of the score is due to a persistent fitted contrast and how much due to instability of the fitting procedure. The integrated squared bias reflects both population violation of AX-invariance and systematic fitting error.

The criterion ranks $\widehat{D}_{n,m}$ and not $D_{n,m}$. Its expected score can differ from the population score through both systematic fitting error, which changes $\mu_{n,m}$, and fitting variability, measured by $V_{n,m}$. Their difference is

$$\widehat{D}_{n,m} - D_{n,m} = \{\widehat{\Phi}_{P_n} - \Phi_{P_0}\} - \{\widehat{\Phi}_{\widehat{Q}_{n,m}} - \Phi_{Q_{n,m}}\}.$$

Consequently, the criterion evaluates the fitted procedure rather than the population functional alone.

3.5.3 Testing a covariance implication of AX-invariance

The ranking described in the previous section is descriptive. We now propose a test for AX-invariance.

We need three independent data sources. The auxiliary random element $\mathcal{V}$ is used to construct $\widehat{p}$ and is independent of $\mathcal{D}_n$. The sample $\mathcal{D}_n$ is used to fit $\widehat{\Phi}_{P_n}$ and, after outcome-blind DRPL selection, to fit $\widehat{\Phi}_{\widehat{Q}_{n,m}}$. Let $\mathcal{T}_{n,m}$ be the sigma-field generated by $\mathcal{V}$, $\mathcal{D}_n$, the selected tuple, and any additional fitting or tuning randomness, and assume that $(\omega, x) \mapsto \widehat{D}_{n,m}(\omega, x)$ is $\mathcal{T}_{n,m} \otimes \mathcal{B}(\mathcal{X})$-measurable. Finally,

$$\mathcal{E}_r = (A_j^{\mathrm{e}}, X_j^{\mathrm{e}})_{j=1}^r$$

is i.i.d. from $P_0^{A,X}$ and independent of $\mathcal{T}_{n,m}$.

For numerically valued A, full AX-invariance implies the necessary condition

$$\operatorname{cov}_{P_0}\{D_{n,m}(X), A\} = 0 \qquad \text{for every } 1 \le m \le n. \tag{3.25}$$

Thus a nonzero covariance falsifies invariance.

On $\mathcal{E}_r$, write

$$\overline{\widehat{D}}_{n,m} = \frac{1}{r}\sum_{j=1}^{r} \widehat{D}_{n,m}(X_j^{\mathrm{e}}), \qquad \overline{A}_r = \frac{1}{r}\sum_{j=1}^{r} A_j^{\mathrm{e}},$$

and use the usual sample covariance

$$\widehat{\mathrm{cov}}_r\{\widehat{D}_{n,m}(X), A\} = \frac{1}{r}\sum_{j=1}^{r}\{\widehat{D}_{n,m}(X_j^{\mathrm{e}}) - \overline{\widehat{D}}_{n,m}\}(A_j^{\mathrm{e}} - \overline{A}_r).$$

Define

$$\widehat{\sigma}^2_{n,m} = \frac{1}{r}\sum_{j=1}^{r}\left[\{\widehat{D}_{n,m}(X_j^{\mathrm{e}}) - \overline{\widehat{D}}_{n,m}\}(A_j^{\mathrm{e}} - \overline{A}_r) - \widehat{\mathrm{cov}}_r\{\widehat{D}_{n,m}(X), A\}\right]^2.$$

Conditional on $\mathcal{T}_{n,m}$, and with $(A, X) \sim P_0^{A,X}$ a fresh independent draw, define

$$W_{n,m} = \{\widehat{D}_{n,m}(X) - \mathbb{E}_{P_0}\widehat{D}_{n,m}(X)\}\{A - \mathbb{E}_{P_0}A\},$$

and set

$$\sigma^2_{n,m} = \mathrm{var}_{P_0}(W_{n,m}).$$

For the triangular sequence below, all ratios involving $\sigma_{n,m}$ are set to zero outside $\{0 < \sigma_{n,m} < \infty\}$. For every $\varepsilon > 0$, the evaluation sample is required to satisfy

$$\frac{1}{\sigma^2_{n,m}}\mathbb{E}_{P_0}\Big[\{W_{n,m} - \mathbb{E}_{P_0}W_{n,m}\}^2 \times \mathbb{1}\big\{|W_{n,m} - \mathbb{E}_{P_0}W_{n,m}| > \varepsilon\sqrt{r}\,\sigma_{n,m}\big\}\Big] \xrightarrow{p} 0, \tag{3.26}$$

$$\frac{\sqrt{r}}{\sigma_{n,m}}\{\overline{\widehat{D}}_{n,m} - \mathbb{E}_{P_0}\widehat{D}_{n,m}(X)\}\{\overline{A}_r - \mathbb{E}_{P_0}A\} \xrightarrow{p} 0, \tag{3.27}$$

$$\frac{\widehat{\sigma}^2_{n,m}}{\sigma^2_{n,m}} \xrightarrow{p} 1. \tag{3.28}$$

These conditions control, respectively, the conditional tails, the sample-mean centring remainder, and consistency of the empirical variance on its possibly changing scale.

Theorem 3.5.3 (Asymptotic normality of the held-out covariance)**.** *For $r \to \infty$ and $1 \le m = m_r \le n = n_r$, suppose* $\Pr(0 < \sigma_{n,m} < \infty) \to 1$. *Suppose that* (3.26)–(3.28) *hold. Suppose further that the fitting contribution is negligible:*

$$\frac{\sqrt{r}}{\sigma_{n,m}}\left|\mathrm{cov}_{P_0}\{\widehat{D}_{n,m}(X) - D_{n,m}(X), A\}\right| \xrightarrow{p} 0, \tag{3.29}$$

Then

$$\frac{\sqrt{r}\left[\widehat{\mathrm{cov}}_r\{\widehat{D}_{n,m}(X), A\} - \mathrm{cov}_{P_0}\{D_{n,m}(X), A\}\right]}{\widehat{\sigma}_{n,m}} \xrightarrow{d} \mathcal{N}(0,1), \tag{3.30}$$

with the ratio set to zero when $\widehat{\sigma}_{n,m} = 0$. *Fix* $\alpha \in (0,1)$. *The test that rejects when*

$$\left|\widehat{\mathrm{cov}}_r\{\widehat{D}_{n,m}(X), A\}\right| > z_{1-\alpha/2}\frac{\widehat{\sigma}_{n,m}}{\sqrt{r}} \tag{3.31}$$

has pointwise asymptotic level α *under the covariance null* (3.25)*, for each fixed* P_0 *and prespecified sequence* $(m_r)_r$ *satisfying these conditions.*

The proof in Appendix 3.A first establishes the studentised evaluation-sample limit centred at the covariance of the fitted contrast. The substantive extra requirement is (3.29): it links that covariance to the population contrast being tested. If $\mathrm{var}_{P_0}(A) < \infty$, Cauchy–Schwarz gives the sufficient condition

$$\frac{\sqrt{r\ \mathrm{var}_{P_0}(A)}}{\sigma_{n,m}}\left\{\|\widehat{\Phi}_{P_n} - \Phi_{P_0}\|_{L^2(P_0^X)} + \|\widehat{\Phi}_{\widehat{Q}_{n,m}} - \Phi_{Q_{n,m}}\|_{L^2(P_0^X)}\right\} \xrightarrow{p} 0.$$

If $\sigma_{n,m}$ is bounded away from zero and $r = \lfloor\sqrt{m}\rfloor$, it is enough for the expression in braces to be $o_p(m^{-1/4})$. Note however that $\sigma_{n,m}$ may shrink under the null.

3.6 Experiments

We study AX-invariance in both simulated scenarios and a real-world setting. In the simulation study, we consider a simple example in which we vary the dependence between A and X as well as the resampling size m, and then extend the analysis to multidimensional X. In the empirical study, focusing on fairness, we test for AX-invariance with respect to sex when estimating income in the `ACSIncome` data set (Ding et al., 2021).

Table 3.1 lists the estimators considered in our study. Appendix 3.B provides detailed descriptions of the methods used to derive several of these estimators.

3.6.1 Simulation study

We define three data-generating processes (DGPs), Scenarios 1–3. Scenarios 1 and 2 use binary A, whereas Scenario 3 uses non-binary A. We report covariance-test results across repeated simulations and varying resample sizes.

Data-generating processes

Scenario 1 (partially linear model). Let $Z_A \sim \mathrm{Bernoulli}(1/2)$, $Z_{X_1} \sim \mathcal{N}(0, 1/4)$, and $Z_{X_2}, Z_Y \sim \mathcal{N}(0,1)$. We generate

$$\begin{aligned} A &= Z_A, \\ X_1 &= \alpha(A - 1/2) + (1-\alpha)Z_{X_1}, \\ X_2 &= Z_{X_2}, \\ Y &= \sin(X_2) + \theta A + Z_Y, \end{aligned}$$

Method	Target	Estimation method
`WC-Plugin-IV`	Φ_P^{β,u^A} in (3.11), $\beta \in [0,\infty]$	Direct plug-in midpoint for binary A under state-common conditional variance.
`WC-EFF1-IV`	Φ_P^{β,u^A} in (3.11), $\beta \in [0,\infty]$	Cross-fitted augmented pseudo-outcome estimator for binary A under state-common conditional variance; Appendix 3.B.2.
`WC-EFF2-IV`	Φ_P^{β,u^A} in (3.11), $\beta \in [0,\infty]$	Partially linear estimator for binary A under state-common conditional variance; Appendix 3.B.1.
`WC-Plugin`	Φ_P^{∞,u^A} in (3.11)	Direct plug-in using the gradient-descent branch of Algorithm 3.1, with the same prespecified $\varepsilon > 0$ for both fits and $\beta = 1/\varepsilon$.
`Unif-Plugin`	Φ_P^{0,u^A} in (3.14)	Direct uniform-reference plug-in for finite A.
`COND-MEAN`	$\mathbb{E}_P[Y \mid X]$	Direct regression of Y on X.
`Infeasible fair`	Simulation-only oracle benchmark	Regression on components of X known from the DGP to enter the response directly, excluding variables used only as proxies for A.

Table 3.1: Methods used in the empirical results. The prefix `WC` refers to the worst-case member Φ_P^{∞}, and the suffix `IV` records the state-common conditional-variance assumption. For binary A with uniform reference weights, this assumption makes Φ_P^{β,u^A} equal to the midpoint of the two state-specific conditional means for every $\beta \in [0,\infty]$; hence the first three targets do not depend on β and include the worst-case target at $\beta = \infty$. `WC-EFF2-IV` additionally requires the partially linear model. `WC-Plugin` computes a finite-β approximation to the worst-case target.

where $X = (X_1, X_2)^\top$ and $\alpha, \theta \in \mathbb{R}$, and simulate n instances.

A natural oracle benchmark uses only X_2 to predict Y, thereby excluding X_1, which is included solely as a proxy for A. Without knowledge of the DGP, unrestricted access to X would generally lead one to exploit information about A indirectly through X_1. As α increases, the component of X_1 attributable to A grows, while $X_1 \perp\!\!\!\perp A$ when $\alpha = 0$. We therefore expect rejection frequencies to be near the nominal level at $\alpha = 0$ and to increase for non-AX-invariant estimators as α increases. For each value of α, Table 3.2 reports the simulated relationship between A and X.

In the simulations, we set $\theta = 1$, which is also reflected in the following multidimensional DGP.

Scenario 2 (multidimensional X). In this scenario, we extend the DGP from Scenario 1 to include a multidimensional set of features X in order to resemble a real-world data setting. Consequently, the resulting models exhibit similar behaviour with respect to α.

Introduce $Z'_A \sim \text{Bernoulli}(1/2)$, $(Z'_{X_1}, \ldots, Z'_{X_9})^\top \sim \mathcal{N}(0, \Sigma)$, where Σ is a 9×9 matrix with unit diagonal and all off-diagonal entries equal to 0.1, and let $Z'_{X_{10}}, Z'_Y \sim$

$\mathcal{N}(0,1)$. We generate

$$\begin{aligned} A' &= Z'_A, \\ X'_j &= \alpha(A' - 1/2) + (1-\alpha)Z'_{X_j}, \qquad j = 1, \ldots, 8, \\ X'_9 &= Z'_{X_9}, \\ X'_{10} &= Z'_{X_{10}}, \\ Y' &= \sin(X'_{10}) + X'_9 + A' + Z'_Y. \end{aligned}$$

Scenario 2 closely resembles Scenario 1, differing only in the larger number of features in X' that are not direct predictors but may be used indirectly to infer A'. An additional feature of X' is included linearly, although this feature is unrelated to A'.

Scenario 3 (non-binary A). We further extend our simulation study to the case of non-binary A. Scenario 3 uses the same DGP as Scenario 1, except that A is chosen uniformly from $\{0, 0.25, 0.5, 0.75, 1\}$.

Table 3.2 reports the relationship between A and X in each scenario using the empirical R^2 from a linear regression of A on X. When $\alpha = 0$, the reported R^2 is zero, as expected because $A \perp\!\!\!\perp X$. As α increases, the proportion of variation in A explained by X also increases. Across scenarios, the relationship is similar for each fixed value of α.

	R^2 by scenario				
α	1	2A	2B	2C	3
0.0	0.000	0.000	0.000	0.000	0.000
0.1	0.012	0.015	0.016	0.016	0.006
0.2	0.059	0.072	0.071	0.071	0.030
0.3	0.155	0.186	0.185	0.184	0.084
0.4	0.308	0.353	0.355	0.355	0.181

Table 3.2: Empirical R^2 values from fitting A given X with a linear model, by scenario and α, estimated from 10^6 data points.

Simulation results

First, we aim to choose the resample size m to maximize the power of the test. In other words, we want m to be as large as possible while ensuring that $Q_{n,m}$ remains sufficiently far from P_0. In particular, m should be large enough to allow a large evaluation sample r, thereby increasing power, while r must still be small enough relative to m to satisfy the fitting condition in (3.29). Figure 3.3 plots the empirical R^2 over m when fitting a linear model of A on X from the Scenario 1 DGP. The R^2 remains stable until around $m = 10^5$, after which it starts moving towards the R^2

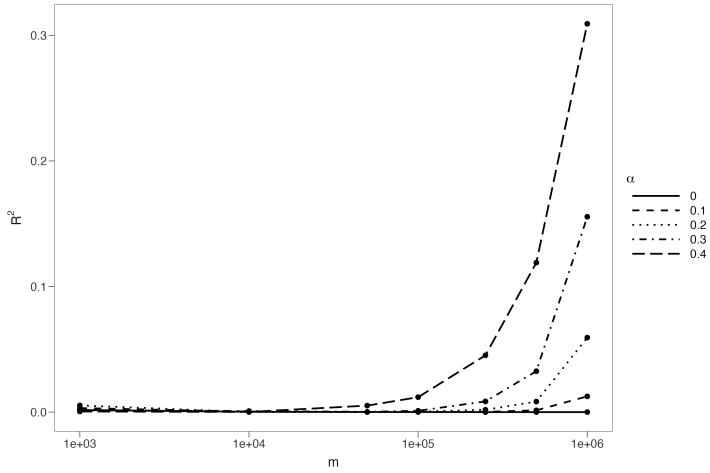


Figure 3.3: Empirical R^2 from regressing A on X in the resample, by resample size m, for Scenario 1 and several values of α. The horizontal axis is logarithmic.

value under P_0 shown in Table 3.2. When $\alpha = 0$, we have $A \perp\!\!\!\perp X$, so the test has no power, which is reflected by $R^2 = 0$ in both the figure and the table.

Given these observations, we conduct 1000 iterations for each scenario. In each iteration, a data set of size $n = 10^6$ is drawn from the DGP and split as described in Section 3.5.3, with $m = 10^5$, $r = \sqrt{m}$, and $q = 0.1$. For each iteration and each α, we calculate the ranking score in (3.23) and compare the standing across methods. We then estimate

$$\widehat{c} = \widehat{\mathrm{cov}}_r\Big\{\widehat{\Phi}_{P_n}(X) - \widehat{\Phi}_{\widehat{Q}_{n,m}}(X), A\Big\}$$

and the plug-in estimate $\widehat{\sigma}$ of the asymptotic variance given in Theorem 3.5.3. This gives the two-sided p-value

$$p = 2\left[1 - \Phi_{\mathcal{N}}\left(\left|\frac{\sqrt{r}\,\widehat{c}}{\widehat{\sigma}_{n,m}}\right|\right)\right].$$

where $\Phi_{\mathcal{N}}$ is the standard normal distribution function. The results are shown in Figures 3.4, 3.6, and 3.7, using the estimators in Table 3.1. The `Unif-Plugin` method appears only in non-binary Scenario 3 because it otherwise coincides with `WC-Plugin-IV`.

We also include a simulation-only estimator that predicts using only features known from the DGP to enter the response directly, thereby avoiding variables used only as proxies for A. In Scenario 1, this is the plug-in estimator of $\mathbb{E}_{P_0}[Y \mid X_2]$, which we call `Infeasible fair`. This estimator and all plug-in estimators used by the methods in Table 3.1 are fitted using XGBoost with a learning rate of 0.5, maximum depth 2, 80% subsampling, and 50 boosting rounds.

To investigate the ranking score more closely, we generate a separate fixed data set of size r from the DGP and use it to estimate the two-component bias–variance decomposition implied by (3.24). Over all iterations, we save the values of $\widehat{\Phi}_{P_n}$ and $\widehat{\Phi}_{\widehat{Q}_{n,m}}$ on the fixed data set, repeating this for each method and value of α. At each evaluation point, the mean fitted contrast over iterations estimates $\mu_{n,m}(x)$, while its variance over iterations estimates the fitting variance. The figures combine the first two terms in (3.24) into the integrated squared bias relative to the zero function, $B^2_{n,m}$, and report the integrated fitting variance $V_{n,m}$ separately. Repeated estimation is available here because the DGP is known and independent training data can be generated for every iteration. Results for Scenario 1 are shown in Figure 3.5.

Figure 3.4 shows that, as α increases and the relationship between A and X strengthens, the conditional mean receives the worst rank in nearly all repetitions. Among the remaining estimators, the ranks are fairly evenly distributed, except that `WC-EFF1-IV` receives first place in most repetitions. At $\alpha = 0.4$, this method also often receives the second-to-last rank.

When $\alpha = 0$, the conditional mean does not have the same approximately uniform rank distribution as the other estimators. A priori, we expect the `Infeasible fair` estimator to receive higher rankings than the competing estimators; however, this is not always the case.

We explain these observations using the decomposition in Figure 3.5. Keeping the scales of the vertical axes in mind, the conditional mean's low rank is driven mainly by its integrated squared bias relative to the zero function. Its integrated fitting variance is fairly stable over α and is also relatively high compared with the other estimators, which explains its ranking in the independent case $\alpha = 0$. In that case, the integrated squared bias of the conditional mean is of a similar order to those of the remaining estimators. The strong ranking of `WC-EFF1-IV` is due to its low fitting variance; excluding the conditional mean, it is consistently the estimator with the highest integrated squared bias. The decomposition also confirms that `Infeasible fair` has the smallest integrated squared bias, while the ranking also reflects its fitting variance, which is generally higher than, but on the same scale as, that of the other estimators.

Overall, the conditional-mean procedure receives the worst fitted ranking in these experiments. The rankings compare fitted procedures but do not determine which population functionals are globally AX-invariant. We therefore also apply the statistical test described above. Figure 3.6 shows that the conditional mean is rejected in nearly all cases when the relationship between A and X is medium to strong, while fewer instances are rejected when the relationship is weak. When $A \perp\!\!\!\perp X$, the rejection rate is 5%, matching the significance level. For the remaining estimators, the rejection frequencies stay near the 5% level for all values of α, so this diagnostic provides no evidence against the tested covariance implication in these settings. It does not establish global AX-invariance. The corresponding covariance

estimates are shown as boxplots in Figure 3.7. As expected from the rejection frequencies, the estimated covariance for the conditional mean moves quickly away from zero as α grows; already at $\alpha = 0.2$, the boxplot no longer overlaps zero. The remaining estimators are distributed evenly around zero.

3.6.2 Real-world data

We now turn to examining AX-invariance in real-world data. We begin by identifying some of the necessary criteria under which our testing framework can be expected to detect AX-invariance:

- **Sample size**. The data should be large enough to accommodate a properly sized resample such that the evaluation set can be sufficiently small relative to m.
- **Size of A.** The conditional density $p(A \mid X)$ must be estimated well for the resampling procedure.
- **Signal of A on Y.** For a reasonably performing estimator not to be AX-invariant, the protected attribute A should have enough signal, given the sample size, on Y that A can be used as a predictor.
- **Signal of X on A.** If there is no signal of X on A, then AX-invariance is achieved through unawareness of A.

In the last two cases, the conditional-mean comparison may become uninformative. In particular, if $Y \perp\!\!\!\perp A \mid X$, or if $X \perp\!\!\!\perp A$ and $P^A = u^A$, then $\mathbb{E}_P[Y \mid X] = \mathbb{E}_{\tau(P)}[Y \mid X]$.

One area in which AX-invariance can be of particular interest is non-life insurance pricing. However, popular open-source CAS data sets do not appear to satisfy all these criteria. In some cases, this means that AX-invariance can be achieved through unawareness.

U.S. Census salary prediction

The `ACSIncome` data set, introduced by Ding et al. (2021), serves as a modern replacement for the widely used `UCIadult` data set. It contains 1,664,500 observations from the U.S. Census with the 12 features listed in Table 3.3.

We formulate the regression problem as estimating log income from the remaining features, excluding sex, which we treat as the protected attribute. This formulation is relevant for applications such as bank credit assessment, where an applicant's salary may be unknown, for example for a student nearing graduation.

Specifically, we define $Y = \log(\texttt{PINCP})$ and use features X consisting of the ordered numeric variables `AGEP`, `SCHL`, and `WKHP`. We treat `SCHL` as ordered because educational attainment increases with the recorded level. The high-cardinality

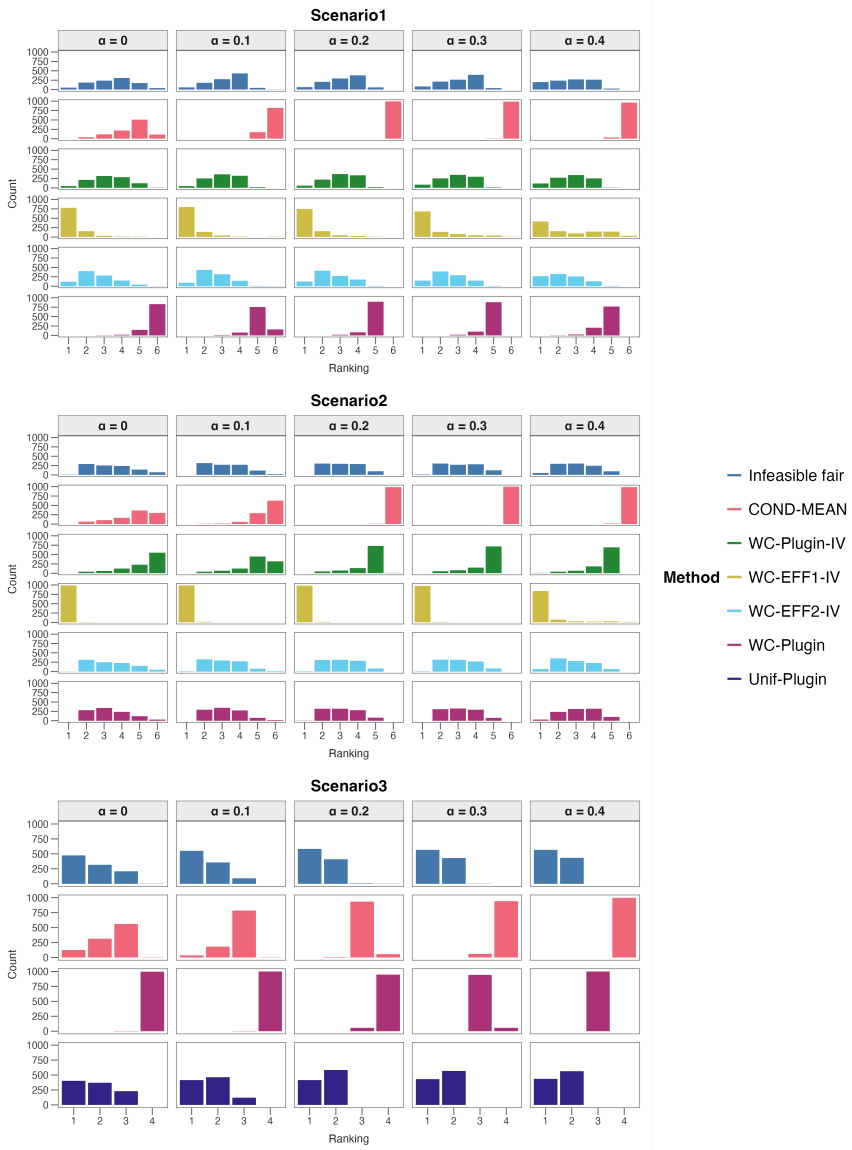


Figure 3.4: Ranks of the fitted procedures over 1000 repetitions. For each method, the figure reports how often it received each rank. The repetitions use $n = 10^6$, $m = 10^5$, $r = \sqrt{m}$, and $q = 0.1$ in all three scenarios.

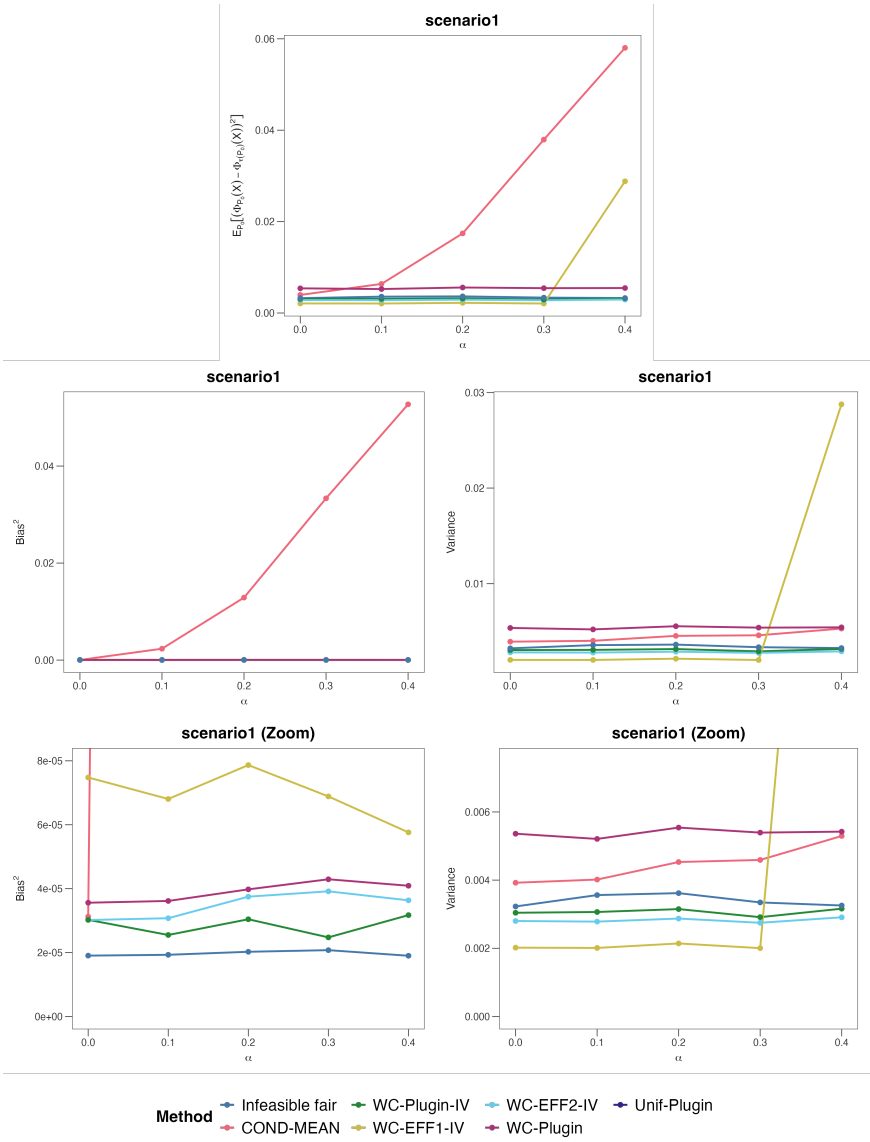


Figure 3.5: Monte Carlo decomposition of the expected fitted ranking score in (3.24) over 1000 iterations of Scenario 1. The panel labelled Bias^2 reports the integrated squared bias relative to the zero function and therefore combines the overall systematic shift with systematic heterogeneity across X. The panel labelled Variance reports integrated fitting variance, and the top panel reports their sum. Zoomed vertical axes are also shown for both components. The iterations use $n = 10^6$, $m = 10^5$, $r = \sqrt{m}$, and $q = 0.1$.

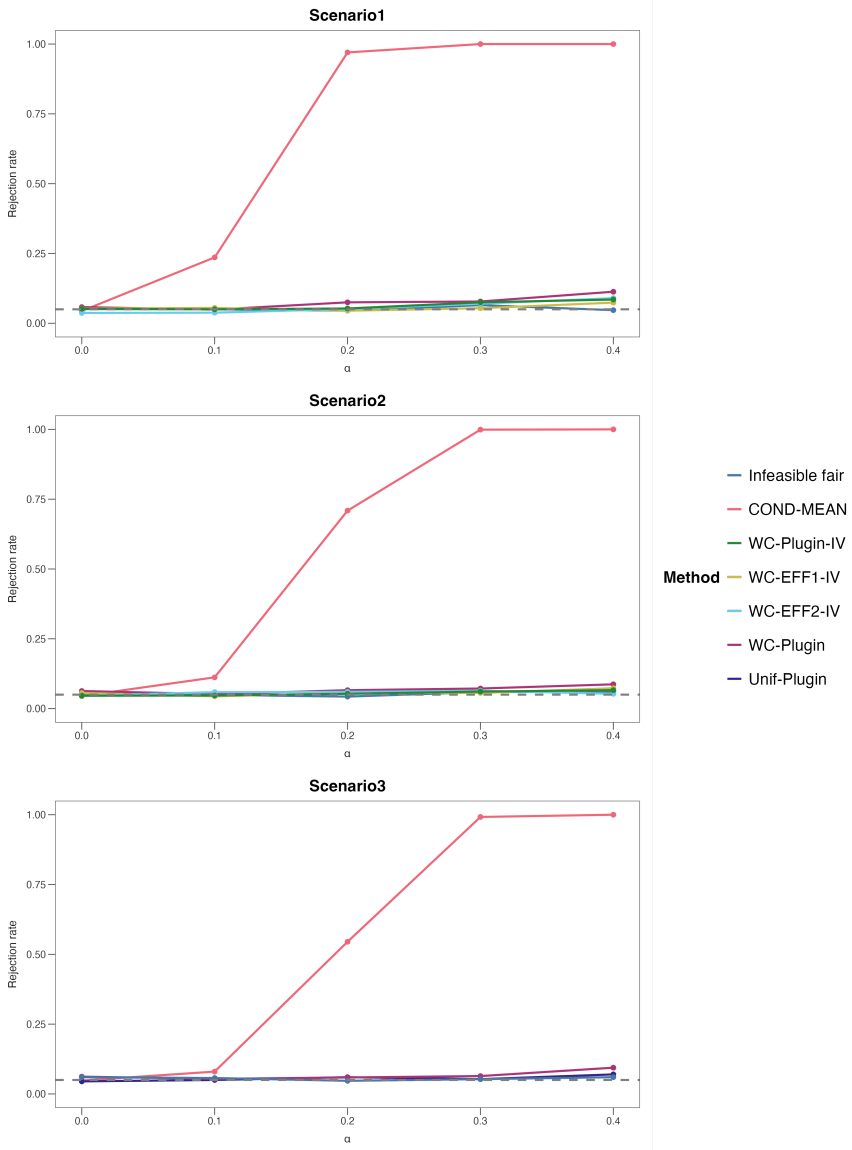


Figure 3.6: Rejection frequencies for the covariance diagnostic over 1000 repetitions. The dashed line marks 5%. The repetitions use $n = 10^6$, $m = 10^5$, $r = \sqrt{m}$, and $q = 0.1$ in all three scenarios.

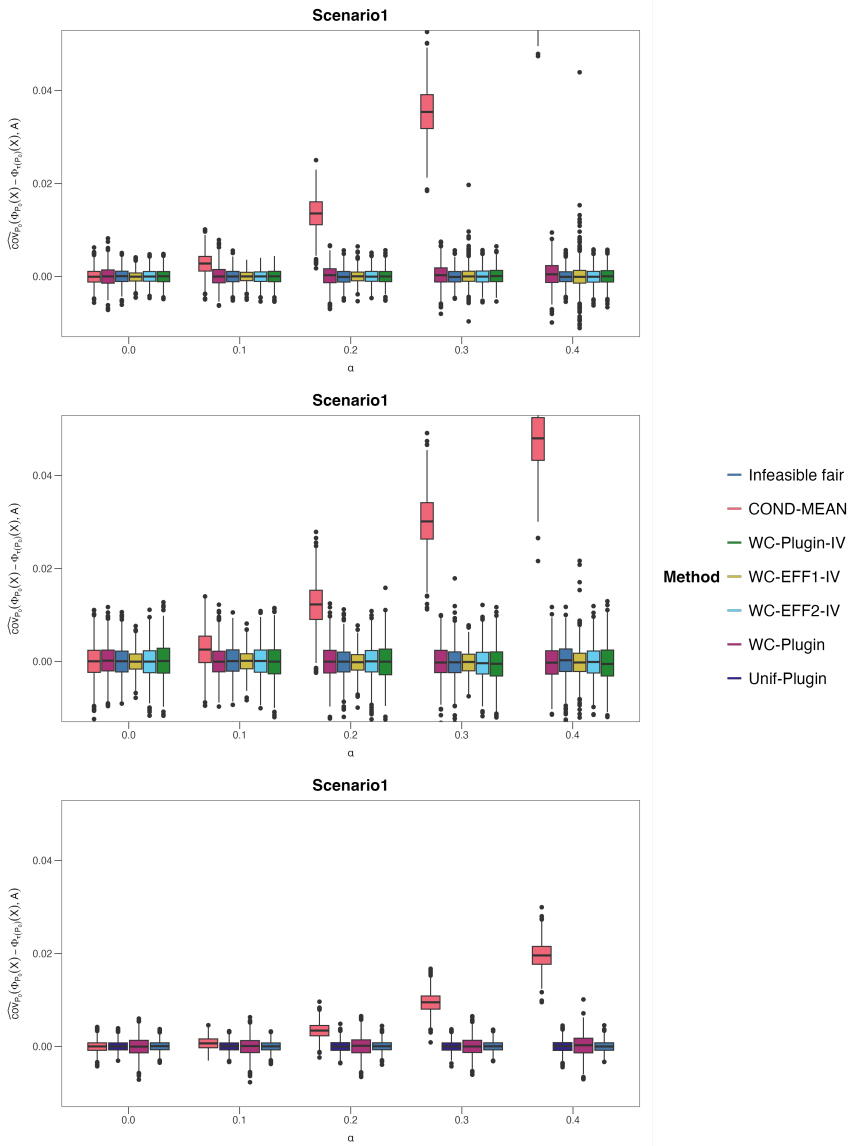


Figure 3.7: Histograms of the mean estimated covariances $\widehat{\mathrm{cov}}_r\{\widehat{D}_{n,m}(X), A\}$ over 1000 iterations. The iterations use $n = 10^6$, $m = 10^5$, $r = \sqrt{m}$, and $q = 0.1$ for all three scenarios. The plots are cut off on the vertical scale at $(-0.01, 0.5)$, leaving some points outside the visible area. In particular, the conditional mean deviates far from zero at large α.

Feature	Support	Ordered?	Description
AGEP	[17,96]	Yes	Age
COW	[1,8]	No	Class of worker
SCHL	[1,24]	-	Educational attainment
MAR	[1,5]	No	Marital status
OCCP	[10,9830]	No	Occupation
POBP	[1,554]	No	Place of birth
RELP	[0,17]	No	Relationship to reference person
WKHP	[1,99]	Yes	Weekly work hours
SEX	[1,2]	No	Sex
RAC1P	[1,9]	No	Race
ST	[1,72]	No	State code
PINCP	[104,1423000]	Yes	Total personal income

Table 3.3: Features in the new adult data set. All features are integers; mappings from integers to categories are given by Ding et al. (2021).

variables COW, MAR, OCCP, POBP, RELP, RAC1P, and ST are encoded using target encoding on the sample set. If a feature level does not appear in the training set, its encoding is the mean target value over the entire training set. The protected attribute A is the binary variable SEX.

When fitting SEX from the other features with a linear model, the reported value is $R^2 = 0.0810$. Table 3.2 suggests that this corresponds to α close to 0.2 in the simulation study, although the data-generating processes are likely not similar.

The testing follows Section 3.5.3 with some modifications. An evaluation set is taken from the input data with size equal to the square root of the training-set size. A fraction $q = 0.1$ of the training set is used to calculate the resampling weights. Testing is then performed sequentially: $m = 10^5$ data points are sampled using these weights and tested on the first $\sqrt{m}$ entries of the evaluation set. The sampled points are removed from the training data, m new points are sampled from the remaining observations, and these are appended to the first resample, which now has size $2m$ and is tested on the first $\sqrt{2m}$ evaluation observations. The procedure is repeated 14 times, at which point the total resample size is $14m$. Using this procedure, we also calculate the estimated covariance and rankings in accordance with Sections 3.5.3 and 3.5.2, respectively. All plug-in estimators in Table 3.1 use XGBoost with maximum depth 2 and 50 boosting rounds.

The results are shown in the tables below. Table 3.4 reports the rankings, from which we observe that the conditional mean and WC-Plugin-IV are ranked worst most frequently, although no certain conclusion emerges. Since independent repeated fits are unavailable, we cannot estimate the integrated squared bias and fitting-variance components. We therefore turn to the p-values of the covariance test in Table 3.5. The test appears volatile for both low and high values of m; however, for mid-sized m, we convincingly reject AX-invariance of the conditional mean.

The `WC-Plugin-IV` estimator is also rejected in some cases, but not in the same convincing manner. The estimated covariances in Table 3.6 tell a similar story by showing a negative covariance statistically significantly away from zero.

m	COND-MEAN	WC-Plugin-IV	WC-EFF1-IV	WC-EFF2-IV
100,000	**0.0195**	0.0190	0.00759	0.0169
200,000	0.0190	0.0180	0.00405	**0.0219**
300,000	**0.0204**	0.0191	0.00291	0.0110
400,000	**0.0161**	0.00725	0.00296	0.00872
500,000	0.00870	**0.0197**	0.00318	0.00584
600,000	0.00889	0.00733	0.00394	**0.00965**
700,000	**0.0105**	0.00549	0.00301	0.00824
800,000	0.00910	**0.0183**	0.00239	0.00650
900,000	0.00822	**0.00986**	0.00246	0.00445
1,000,000	**0.00817**	0.00772	0.00398	0.00499
1,100,000	**0.0163**	0.00809	0.00630	0.0108
1,200,000	0.00800	**0.00845**	0.00210	0.00539
1,300,000	0.00920	**0.0101**	0.00284	0.00403
1,400,000	**0.00916**	0.00775	0.00255	0.00324

Table 3.4: Adult data. Ranking score by m and method. The method with the highest score (worst ranking) is shown in bold for each m.

m	COND-MEAN	WC-Plugin-IV	WC-EFF1-IV	WC-EFF2-IV
100,000	**0.00382**	0.556	0.248	0.678
200,000	**0.0483**	**0.0276**	0.309	0.152
300,000	0.341	0.262	0.359	0.219
400,000	0.627	0.0684	0.713	0.0782
500,000	**0.0000216**	0.229	0.914	0.809
600,000	**0.00000000742**	0.656	0.821	0.564
700,000	**0.0000471**	**0.00365**	0.912	0.680
800,000	**0.0231**	**0.00860**	0.211	0.637
900,000	**0.00867**	0.315	0.0871	0.704
1,000,000	**0.000286**	**0.00182**	0.246	0.703
1,100,000	**0.000706**	0.194	0.685	0.592
1,200,000	0.202	0.632	**0.00745**	0.791
1,300,000	0.515	0.833	0.377	0.619
1,400,000	**0.0281**	0.522	0.326	0.115

Table 3.5: Adult data. Two-sided p-values by m and method. Values below 0.05 are shown in bold.

m	COND-MEAN	WC-Plugin-IV	WC-EFF1-IV	WC-EFF2-IV
100,000	**-0.00558**	0.00113	-0.00140	0.000754
200,000	-0.00379	**0.00411**	0.000902	0.00295
300,000	-0.00188	**0.00216**	0.000690	0.00180
400,000	0.000857	-0.00216	-0.000279	**0.00228**
500,000	**-0.00545**	0.00235	-0.0000844	0.000256
600,000	**-0.00746**	-0.000529	-0.000197	0.000788
700,000	**-0.00576**	-0.00299	0.0000840	0.000519
800,000	-0.00298	**0.00496**	-0.000850	-0.000527
900,000	**-0.00328**	-0.00139	0.00118	-0.000351
1,000,000	**-0.00453**	-0.00377	-0.00102	0.000371
1,100,000	**-0.00602**	-0.00162	-0.000447	0.000776
1,200,000	-0.00159	0.000609	**0.00169**	0.000268
1,300,000	**-0.000868**	0.000293	0.000651	0.000439
1,400,000	**-0.00291**	0.000781	-0.000688	0.00125

Table 3.6: Adult data. Estimated covariance by m and method. The estimate furthest from zero is shown in bold in each row.

3.7 Discussion

This paper separates the arguments of a prediction function from the population quantities used to construct it. A target may be written as a function of x while still depending on the population-specific distribution of (A, X). The ordinary conditional mean is the leading example: it averages the state-specific response surfaces using $P^{A|X=x}$. AX-invariance rules out this dependence. Once the conditional distribution of Y given (A, X) is fixed, changing the composition of (A, X) cannot change the population target.

The causal results explain why this population-level condition matters. A fixed prediction function does not reveal whether protected composition was used to construct it, and the same numerical predictor may arise from an AX-invariant functional or from a population-dependent one. On the rich comparison classes considered here, the path-specific and interventional fairness requirements therefore imply AX-invariance. The implication is only necessary: AX-invariance by itself does not establish a causal fairness notion. Though under stronger causal assumption it is equivalent to some causal fairness notions.

Ax-invariance does not determine a unique predictor. The reference kernel ρ_x specifies how states of A are compared, while β determines how strongly high-risk states influence the robust target. And a choice may depend on the given data and problem.

In the simulations, the diagnostic detects the designed population dependence of the conditional mean as X becomes more informative about A, whereas the invariant constructions remain comparatively stable between P_0 and the corresponding $Q_{n,m}$.

The Census application provides descriptive evidence against AX-invariance for the conditional mean and, at some resample sizes, for WC-Plugin-IV.

3.A Proofs

3.A.1 Proof of Proposition 3.2.1

Proof of Proposition 3.2.1. Define $P \equiv_{Y|AX} Q$ when their conditional response kernels agree $\mu^{A,X}$-almost everywhere. For every $P \in \mathcal{P}$, its response kernel equals itself on all of $\mathcal{A} \times \mathcal{X}$, so $P \equiv_{Y|AX} P$. If $P \equiv_{Y|AX} Q$, then $Q \equiv_{Y|AX} P$. If $P \equiv_{Y|AX} Q$ and $Q \equiv_{Y|AX} S$, choose conull sets S_{PQ} and S_{QS} on which the corresponding kernels agree. The response kernels of P and S then agree on the conull intersection $S_{PQ} \cap S_{QS}$ because, for every (a, x) in that intersection,

$$P^{Y|A=a,X=x} = Q^{Y|A=a,X=x} = S^{Y|A=a,X=x}.$$

The relation is therefore reflexive, symmetric, and transitive. Thus $\equiv_{Y|AX}$ is an equivalence relation, and $\mathcal{Q}(P)$ is the equivalence class of P. In particular $\mathcal{Q}(P)$ partitions $\mathcal{P}$. □

3.A.2 Proofs for causal fairness results

Proof of Lemma 3.3.2. Let S_0 be a measurable $\mu^{A,X}$-conull set on which the chosen response-kernel versions $P_1^{Y|A,X}$ and $P_2^{Y|A,X}$ agree. Because $P_2^{A,X} \sim \mu^{A,X}$, the set S_0 is also $P_2^{A,X}$-conull. Therefore, for measurable $C \subseteq \mathcal{A} \times \mathcal{X}$ and $B \subseteq \mathcal{Y}$,

$$\begin{aligned}\int_C P_1^{Y|A,X}(B \mid a, x)\, P_2^{A,X}(\mathrm{d}a, \mathrm{d}x) &= \int_C P_2^{Y|A,X}(B \mid a, x)\, P_2^{A,X}(\mathrm{d}a, \mathrm{d}x) \\ &= P_2\{(A, X) \in C, Y \in B\}.\end{aligned}$$

Thus $P_1^{Y|A,X}$ is a response-kernel version under both P_1 and P_2. Since $\mathcal{Y}$ is standard Borel, the kernel-randomisation lemma (Kallenberg, 2021, Lemma 4.22) gives a measurable map

$$f : \mathcal{A} \times \mathcal{X} \times [0,1] \longrightarrow \mathcal{Y}$$

such that $f(a, x, U^Y)$ has distribution $P_1^{Y|A,X}(\,\cdot \mid a, x)$ when $U^Y \sim \mathrm{Unif}[0,1]$. A second application gives measurable maps $T_i : [0,1] \to \mathcal{A} \times \mathcal{X}$ such that $T_i(U^{A,X})$ has distribution $P_i^{A,X}$ when $U^{A,X} \sim \mathrm{Unif}[0,1]$.

Write $T_i = (T_i^A, T_i^X)$. For each i, take independent uniform random variables $U_i^{A,X}$ and U_i^Y and define the acyclic assignments

$$A_i = T_i^A(U_i^{A,X}), \qquad X_i = T_i^X(U_i^{A,X}), \qquad Y_i = f(A_i, X_i, U_i^Y).$$

For measurable $C \subseteq \mathcal{A} \times \mathcal{X}$ and $B \subseteq \mathcal{Y}$,

$$\begin{aligned}\Pr\{(A_i, X_i) \in C, Y_i \in B\} &= \int_C P_1^{Y|A,X}(B \mid a, x)\, P_i^{A,X}(\mathrm{d}a, \mathrm{d}x) \\ &= P_i\{(A, X) \in C, Y \in B\}.\end{aligned}$$

Thus $P_{M_i} = P_i$. The assignments are acyclic with (A_i, X_i) preceding Y_i, so $M_i \in \mathcal{M}_{\mathrm{cp}}$. Moreover, $U_i^Y \perp\!\!\!\perp (A_i, X_i)$, so in fact $M_i \in \mathcal{M}_{\mathrm{nc}}$. For every $a \in \mathcal{A}$, measurable $D \subseteq \mathcal{X}$, and measurable $B \subseteq \mathcal{Y}$,

$$P_{M_i^a}(Y \in B, X \in D) = \int_D P_1^{Y|A,X}(B \mid a, x)\, P_i^X(\mathrm{d}x).$$

Thus, for every a, the kernel $x \mapsto P_1^{Y|A,X}(\,\cdot \mid a, x)$ is a version of $P_{M_i^a}^{Y|X=x}$. Choose these versions simultaneously for $i = 1, 2$ and all a; they are jointly measurable because $P_1^{Y|A,X}$ is jointly measurable. Furthermore, for every (a, x),

$$P_{M_i^a}^{Y|X=x} = \mathcal{L}\{f(a, x, U_i^Y)\} = P_{M_i}^{Y|\mathrm{do}(A=a,X=x)} = P_1^{Y|A,X}(\,\cdot \mid a, x).$$

The map f and the uniform response-noise distribution are common to the two models, which proves the result. □

Proof of Lemma 3.3.4. Fix $M \in \mathcal{M}_{\mathrm{nc}}$ and define the probability kernel

$$R_M(B \mid a, x) \coloneqq \Pr_M\{f_Y(a, x, U_Y) \in B\}, \qquad B \subseteq \mathcal{Y} \text{ measurable.}$$

Measurability of f_Y makes R_M a jointly measurable kernel. For measurable $C \subseteq \mathcal{A} \times \mathcal{X}$, independence of U_Y and (A, X) gives

$$\Pr_M\{Y \in B, (A, X) \in C\} = \int_C R_M(B \mid a, x)\, P_M^{A,X}(\mathrm{d}a, \mathrm{d}x).$$

Hence R_M is a version of $P_M^{Y|A,X}$ and $P_M^{Y|\mathrm{do}(A=a,X=x)} = R_M(\,\cdot \mid a, x)$.

The modification from M to M^a changes only the response assignment. Consequently, the joint distribution of (X, U_Y) is unchanged. Since $U_Y \perp\!\!\!\perp (A, X)$ under M, also $U_Y \perp\!\!\!\perp X$, and for every measurable $D \subseteq \mathcal{X}$ and $B \subseteq \mathcal{Y}$,

$$\begin{aligned} P_{M^a}(Y \in B, X \in D) &= \Pr_M\{f_Y(a, X, U_Y) \in B,\ X \in D\} \\ &= \int_D R_M(B \mid a, x)\, P_M^X(\mathrm{d}x). \end{aligned}$$

Hence, for every a, $x \mapsto R_M(\,\cdot \mid a, x)$ is a version of $P_{M^a}^{Y|X=x}$. Choose these versions simultaneously for all a; their joint measurability follows from that of R_M. Combining this fact with the observational-version identity and the joint-intervention identity above proves every equality in (3.4) for $\mu^{A,X}$-almost every (a, x). □

Proof of Theorem 3.3.5. We first prove part (iii). Suppose that Φ is weakly path-specifically fair with respect to $\mathcal{M}_{\mathrm{nc}}$. Let $P_1, P_2 \in \mathcal{P}$ satisfy $P_1^{Y|A,X} = P_2^{Y|A,X}$. By Lemma 3.3.2, there are $M_1, M_2 \in \mathcal{M}_{\mathrm{nc}}$ inducing P_1, P_2 whose chosen fair-world conditional distributions satisfy

$$P_{M_1^a}^{Y|X=x} = P_{M_2^a}^{Y|X=x} \qquad \text{for every } (a, x).$$

Weak path-specific fairness yields $\Phi_{P_1} = \Phi_{P_2}$. This proves AX-invariance.

Conversely, assume $M_0 \in \mathcal{M}_{\mathrm{nc}}$. Let Φ be AX-invariant and let $M_1, M_2 \in \mathcal{M}_{\mathrm{nc}}$ satisfy (3.3). By (3.3) and the two instances of (3.4), there are three $\mu^{A,X}$-conull sets on which the respective equalities hold. Their finite intersection is conull, and on that intersection

$$P_{M_1}^{Y|A=a,X=x} = P_{M_1^a}^{Y|X=x} = P_{M_2^a}^{Y|X=x} = P_{M_2}^{Y|A=a,X=x}.$$

Hence $P_{M_1}^{Y|A,X} = P_{M_2}^{Y|A,X}$, and AX-invariance gives $\Phi_{P_{M_1}} = \Phi_{P_{M_2}}$. This proves weak path-specific fairness with respect to $\mathcal{M}_{\mathrm{nc}}$ and completes part (iii).

For part (i), suppose that Φ is strongly path-specifically fair at a_0 with respect to $\mathcal{C}$, and let $M_1, M_2 \in \mathcal{C}$ satisfy (3.3). Choose a measurable $\mu^{A,X}$-conull set S on which the equality in (3.3) holds, and put

$$S_{a_0} = \{x \in \mathcal{X} : (a_0, x) \in S\}.$$

Since μ^A is σ-finite and $\mu^A(\{a_0\}) > 0$, one has $0 < \mu^A(\{a_0\}) < \infty$, and

$$\begin{aligned} 0 &= (\mu^A \otimes \mu^X)(S^c) \\ &\geq (\mu^A \otimes \mu^X)\{S^c \cap (\{a_0\} \times \mathcal{X})\} \\ &= \mu^A(\{a_0\})\, \mu^X(\mathcal{X} \setminus S_{a_0}). \end{aligned}$$

Thus $\mu^X(\mathcal{X} \setminus S_{a_0}) = 0$, and

$$P_{M_1^{a_0}}^{Y|X=x} = P_{M_2^{a_0}}^{Y|X=x} \qquad \text{for } \mu^X\text{-almost every } x.$$

Strong path-specific fairness therefore gives $\Phi_{P_{M_1}} = \Phi_{P_{M_2}}$, which is weak path-specific fairness.

For part (ii), suppose that $\mathcal{M}_{\mathrm{nc}} \subseteq \mathcal{C}$ and that Φ is weakly path-specifically fair with respect to $\mathcal{C}$. Let $P_1, P_2 \in \mathcal{P}$ satisfy $P_1^{Y|A,X} = P_2^{Y|A,X}$. By Lemma 3.3.2, there are $M_1, M_2 \in \mathcal{M}_{\mathrm{nc}} \subseteq \mathcal{C}$ inducing P_1, P_2 and satisfying

$$P_{M_1^a}^{Y|X=x} = P_{M_2^a}^{Y|X=x} \qquad \text{for every } (a, x).$$

Weak path-specific fairness with respect to $\mathcal{C}$ gives $\Phi_{P_1} = \Phi_{P_2}$. Since P_1, P_2 were arbitrary response-equivalent distributions, Φ is AX-invariant.

For the strong-fairness conclusion, fix $a_0 \in \mathcal{A}$ and suppose that Φ is strongly path-specifically fair at a_0 with respect to $\mathcal{C}$. Let $P_1, P_2 \in \mathcal{P}$ have the same response kernel. Lemma 3.3.2 supplies $M_1, M_2 \in \mathcal{M}_{\mathrm{nc}} \subseteq \mathcal{C}$ with observational distributions P_1, P_2 and

$$P_{M_1^{a_0}}^{Y|X=x} = P_{M_2^{a_0}}^{Y|X=x} \qquad \text{for every } x.$$

Strong path-specific fairness gives $\Phi_{P_1} = \Phi_{P_2}$, proving AX-invariance. □

Proof of Theorem 3.3.7. For part (i), suppose that Φ is interventionally fair with respect to $\mathcal{C}$. Let $P_1, P_2 \in \mathcal{P}$ have the same response kernel. Lemma 3.3.2 supplies $M_1, M_2 \in \mathcal{M}_{\mathrm{nc}} \subseteq \mathcal{C}$ inducing these distributions and having the same joint-intervention response distributions $\mu^{A,X}$-almost everywhere. Interventional fairness yields $\Phi_{P_1} = \Phi_{P_2}$, proving AX-invariance.

For part (ii), interventional fairness with respect to $\mathcal{M}_{\mathrm{va}}$ implies AX-invariance by part (i), since $\mathcal{M}_{\mathrm{nc}} \subseteq \mathcal{M}_{\mathrm{va}} \subseteq \mathcal{M}_{\mathrm{cp}}$. Conversely, suppose that Φ is AX-invariant and $M_0 \in \mathcal{M}_{\mathrm{va}}$. Let $M_1, M_2 \in \mathcal{M}_{\mathrm{va}}$ satisfy (3.6). Intersect the conull set from (3.6) with the two conull sets supplied by (3.5). This finite intersection is conull, and on it

$$P_{M_1}^{Y|A=a,X=x} = P_{M_1}^{Y|\mathrm{do}(A=a,X=x)} = P_{M_2}^{Y|\mathrm{do}(A=a,X=x)} = P_{M_2}^{Y|A=a,X=x}.$$

Hence $P_{M_1}^{Y|A,X} = P_{M_2}^{Y|A,X}$, and AX-invariance gives $\Phi_{P_{M_1}} = \Phi_{P_{M_2}}$. □

3.A.3 Proofs for invariant prediction targets

For each $P \in \mathcal{P}$, fix a conditional-kernel version and jointly Borel-measurable, real-valued moment versions m_P, v_P satisfying the assumptions of Theorem 3.4.2, and put

$$\bar{r}_P(a, x, w) \coloneqq v_P(a, x) + \{m_P(a, x) - w\}^2.$$

Let E_P be the measurable $\mu^{A,X}$-null set on which either identity in (3.9) fails for these choices. Fubini's theorem gives a Borel μ^X-conull set of x for which $\mu^A((E_P)_x) = 0$. The set $\mathcal{X}_P$ in the main text is an existentially chosen witness, so shrink it to its intersection with this conull set without relabelling it. Then, for every $x \in \mathcal{X}_{P,\rho}$ and $w \in \mathbb{R}$,

$$r_P(a, x, w) = \bar{r}_P(a, x, w) \qquad \text{for } \rho_x\text{-almost every } a, \tag{3.32}$$

because $\rho_x \ll \mu^A$. Thus the average, LogSumExp, and essential-supremum aggregators in (3.11) have the same values whether calculated from r_P or from $\bar{r}_P$. In the pointwise continuity and maximum arguments below, we use the continuous representative $\bar{r}_P$.

Proof of Lemma 3.4.1. Fix $P \in \mathcal{P}$, $x \in \mathcal{X}_{P,\rho}$, $0 < \beta < \infty$, and $w \in \mathbb{R}$, and set $f(a) = \bar{r}_P(a, x, w)$. Continuity of the moment surfaces makes f continuous and hence bounded on the compact space $\mathcal{A}$. Define

$$Z_f = \int_{\mathcal{A}} \exp\{\beta f(a')\}\rho_x(\mathrm{d}a'), \qquad \frac{\mathrm{d}\pi_f}{\mathrm{d}\rho_x}(a) = \frac{\exp\{\beta f(a)\}}{Z_f}.$$

Boundedness gives $0 < Z_f < \infty$ and $\pi_f \sim \rho_x$. For every $\pi \ll \rho_x$, define

$$b_f(a) = -\beta f(a) + \log Z_f, \qquad B_f = \sup_{a \in \mathcal{A}} |b_f(a)| < \infty.$$

The Radon–Nikodym chain rule gives, π-almost surely,

$$\log \frac{\mathrm{d}\pi}{\mathrm{d}\pi_f} = \log \frac{\mathrm{d}\pi}{\mathrm{d}\rho_x} + b_f.$$

Moreover,

$$\left| \int_{\mathcal{A}} b_f \, \mathrm{d}\pi \right| \leq B_f < \infty.$$

Consequently, integration gives the following identity in the extended real line:

$$\mathrm{KL}(\pi \| \pi_f) = \mathrm{KL}(\pi \| \rho_x) + \int_{\mathcal{A}} b_f \, \mathrm{d}\pi = \mathrm{KL}(\pi \| \rho_x) - \beta \int_{\mathcal{A}} f \, \mathrm{d}\pi + \log Z_f.$$

The displayed bound shows directly that one divergence is infinite if and only if the other is infinite, so no indeterminate difference of infinities is involved. Rearranging, with the convention that a finite number minus $+\infty$ is $-\infty$, gives

$$\int_{\mathcal{A}} f \, \mathrm{d}\pi - \frac{1}{\beta} \mathrm{KL}(\pi \| \rho_x) = \frac{1}{\beta} \log Z_f - \frac{1}{\beta} \mathrm{KL}(\pi \| \pi_f) \leq \frac{1}{\beta} \log Z_f.$$

Gibbs' inequality gives $\mathrm{KL}(\pi \| \pi_f) \geq 0$, with equality if and only if $\pi = \pi_f$. Hence equality in the preceding display holds if and only if $\pi = \pi_f$. The variational problem is explicitly restricted to $\pi \ll \rho_x$, and the optimizer $\pi_f \sim \rho_x$ belongs to this class. Since $\beta^{-1} \log Z_f = \mathcal{R}_{\beta,\rho,x}(f)$, taking the supremum over this effective domain gives

$$\mathcal{R}_{\beta,\rho,x}(f) = \sup_{\substack{\pi \in \mathfrak{P}(\mathcal{A}) \\ \pi \ll \rho_x}} \left\{ \int_{\mathcal{A}} f \, \mathrm{d}\pi - \frac{1}{\beta} \mathrm{KL}(\pi \| \rho_x) \right\}.$$

This holds for every w. By (3.32), both the aggregator and every integral in this restricted variational problem are unchanged when f is replaced by $r_P(\,\cdot\,, x, w)$. Substitution into (3.11) proves the proposed version of (3.12). □

Proof of Theorem 3.4.2. Fix $P \in \mathcal{P}$, a conditional-kernel version, and jointly Borel-measurable, real-valued moment-surface versions satisfying the assumptions of the theorem. Fix $x \in \mathcal{X}_{P,\rho}$ and abbreviate

$$m(a) = m_P(a, x), \qquad v(a) = v_P(a, x), \qquad \nu = \rho_x.$$

The maps m and v are continuous on the compact space $\mathcal{A}$, hence bounded. Put

$$r(a, w) = v(a) + \{m(a) - w\}^2, \qquad F_\beta(w) = \mathcal{R}_{\beta,\rho,x}\big(r(\,\cdot\,, w)\big),$$

and let $L = \min_a m(a)$, $U = \max_a m(a)$, and $M = \max_a |m(a)|$.

Part (i): well-definedness and invariance. For every $\beta \in [0, \infty]$, the map $f \mapsto \mathcal{R}_{\beta,\rho,x}(f)$ is convex on bounded Borel functions. This is immediate for the average and essential supremum, while the LogSumExp case follows from Hölder's inequality. For $t \in [0, 1]$, let $w_t = tw_1 + (1 - t)w_2$. Then

$$r(a, w_t) = tr(a, w_1) + (1 - t)r(a, w_2) - t(1 - t)(w_1 - w_2)^2.$$

Convexity and translation equivariance therefore give

$$F_\beta(w_t) \le tF_\beta(w_1) + (1-t)F_\beta(w_2) - t(1-t)(w_1-w_2)^2,$$

so F_β is strongly convex. Moreover, $r(a,w) \ge \{(|w|-M)_+\}^2$ for every a, and monotonicity gives

$$F_\beta(w) \ge \{(|w|-M)_+\}^2.$$

Thus F_β is coercive. It is finite everywhere and hence continuous as a finite convex function on $\mathbb{R}$. It therefore attains its minimum, which is unique by strong convexity. Full support of ν and continuity in a also give

$$F_\infty(w) = \max_{a\in\mathcal{A}} r(a,w).$$

We next prove measurability. Write

$$F_\beta(x,w) = \mathcal{R}_{\beta,\rho,x}\big(\bar r_P(\,\cdot\,,x,w)\big), \qquad (x,w) \in \mathcal{X}_{P,\rho}\times\mathbb{R}.$$

For $\beta<\infty$, it is Borel in (x,w) by measurability of integration against a probability kernel, applied to $\bar r_P$ or $\exp\{\beta\bar r_P\}$. The relevant integrals are finite, and the latter is positive, because their a-sections are continuous on compact $\mathcal{A}$. For $\beta=\infty$, if $D\subseteq\mathcal{A}$ is countable and dense, then

$$F_\infty(x,w) = \max_{a\in\mathcal{A}} \bar r_P(a,x,w) = \sup_{a\in D} \bar r_P(a,x,w),$$

so F_∞ is Borel as a countable supremum.

Enumerate $\mathbb{Q}=\{q_1,q_2,\dots\}$ and define

$$\gamma_\beta(x) = \inf_{j\ge1} F_\beta(x,q_j),\ \ j_n(x) = \min\{j : F_\beta(x,q_j) < \gamma_\beta(x) + n^{-1}\},\ \ s_n(x) = q_{j_n(x)}.$$

Continuity in w shows that $\gamma_\beta(x)$ is the minimum value. The countable infimum γ_β is Borel, and the first-index construction makes j_n and s_n Borel. If $w_\beta(x)$ is the unique minimiser, apply the strong-convexity inequality above with $w_1=s_n(x)$ and $w_2=w_\beta(x)$. Using $F_\beta(x,w_\beta(x)) \le F_\beta(x,w_t)$, rearranging, and letting $t\downarrow 0$ gives

$$|s_n(x)-w_\beta(x)|^2 \le F_\beta(x,s_n(x)) - \gamma_\beta(x) < n^{-1}.$$

Hence $s_n(x)\to w_\beta(x)$, so the minimiser is measurable. Its zero extension from the Borel conull set $\mathcal{X}_{P,\rho}$ is a measurable representative in $L^0(\mu^X)$.

Finally, consider two admissible constructions, indexed by $i=1,2$, either for the same population or for response-equivalent $P,Q\in\mathcal{P}$. Set $P_1=P_2=P$ in the first case and $P_1=P$, $P_2=Q$ in the second. Write $\mathcal{X}_{i,\rho}$ for the Borel μ^X-conull domain $\mathcal{X}_{P_i,\rho}$ associated with construction i. Their conditional kernels agree as probability measures outside a measurable $\mu^{A,X}$-null set: in the first case by almost-everywhere uniqueness and $P^{A,X}\sim\mu^{A,X}$, and in the second by response

equivalence. Fubini's theorem gives a Borel μ^X-conull set $\mathcal{X}_{12}$ on which the two kernels, and hence their squared-error risks for every w, agree for μ^A-almost every a. For every $x \in \mathcal{X}_{12} \cap \mathcal{X}_{1,\rho} \cap \mathcal{X}_{2,\rho}$, domination $\rho_x \ll \mu^A$ makes the risks equal ρ_x-almost everywhere. All three objectives therefore agree for every w, so uniqueness makes their minimisers agree on a μ^X-conull set. The same-population case proves version-independence in $L^0(\mu^X)$, and the response-equivalent case proves AX-invariance. This completes part (i).

Part (ii): internality. If $w < L$, then, for every $a \in \mathcal{A}$,

$$r(a, w) - r(a, L) = (L - w)\{2m(a) - L - w\} \geq (L - w)^2.$$

The aggregator properties in (3.8) then give $F_\beta(w) \geq F_\beta(L) + (L - w)^2 > F_\beta(L)$. Similarly, if $w > U$,

$$r(a, w) - r(a, U) = (w - U)\{w + U - 2m(a)\} \geq (w - U)^2,$$

so $F_\beta(w) > F_\beta(U)$. The minimiser must therefore lie in $[L, U]$, which proves (3.13) for every $\beta \in [0, \infty]$.

Part (iii): continuity in β. By part (ii), all minimisers lie in the compact interval $I = [L, U]$. The risks $r(a, w)$ are bounded on $\mathcal{A} \times I$. Let

$$C = \max_{(a,w) \in \mathcal{A} \times I} r(a, w) - \min_{(a,w) \in \mathcal{A} \times I} r(a, w).$$

For each w, apply Jensen's inequality and Hoeffding's lemma to the bounded random variable $r(A, w)$ under $A \sim \nu$. Its range length is at most C, so, uniformly in $w \in I$,

$$0 \leq F_\beta(w) - F_0(w) \leq \frac{\beta C^2}{8}.$$

Thus $F_\beta \to F_0$ uniformly on I as $\beta \downarrow 0$. For $\beta_0 \in (0, \infty)$, choose $0 < b_- < \beta_0 < b_+ < \infty$. The integrand

$$(a, \beta, w) \longmapsto \exp\{\beta r(a, w)\}$$

is continuous on the compact set $\mathcal{A} \times [b_-, b_+] \times I$, so it has a finite constant envelope. Dominated convergence therefore shows that its integral against ν is continuous in (β, w). The integral is strictly positive, so composing with $(\beta, z) \mapsto \beta^{-1} \log z$ proves joint continuity of $F_\beta(w)$. Uniform continuity on the compact set $[b_-, b_+] \times I$ then gives

$$\sup_{w \in I} |F_\beta(w) - F_{\beta_0}(w)| \longrightarrow 0 \qquad (\beta \to \beta_0).$$

For the worst-case target, fix $\varepsilon > 0$. Uniform continuity on $\mathcal{A} \times I$ provides $\delta > 0$ such that $|r(a, w) - r(a', w)| < \varepsilon$ whenever $d(a, a') < \delta$, uniformly in $w \in I$. Choose a finite cover $\mathcal{A} \subseteq \bigcup_{j=1}^N B(a_j, \delta/2)$. Full support of ν gives

$$c_\varepsilon = \min_{1 \leq j \leq N} \nu\big(B(a_j, \delta/2)\big) > 0.$$

The number c_ε may depend on ε, but it is independent of w and β. For each $w \in I$, choose a maximiser a_w. Some cover ball contains a_w, and every point of that ball is within δ of a_w. Integrating the exponential over that ball shows that, for every $\beta > 0$ and every $w \in I$,

$$F_\infty(w) - \varepsilon + \frac{\log c_\varepsilon}{\beta} \le F_\beta(w) \le F_\infty(w),$$

where the upper bound follows from $r(a, w) \le F_\infty(w)$. Equivalently,

$$\sup_{w \in I} \{F_\infty(w) - F_\beta(w)\} \le \varepsilon - \frac{\log c_\varepsilon}{\beta}.$$

Given $\eta > 0$, take $\varepsilon = \eta/2$ and then choose β large enough that $-\log c_{\eta/2}/\beta < \eta/2$. The last display is then smaller than η, proving $F_\beta \to F_\infty$ uniformly on I as $\beta \to \infty$.

The required implication from objective convergence to minimiser convergence is as follows. Suppose continuous G_n, G on I satisfy $\|G_n - G\|_\infty \to 0$, and let their unique minimisers be u_n, u. For $\varepsilon > 0$, if $K_\varepsilon = \{z \in I : |z - u| \ge \varepsilon\}$ is non-empty, compactness and uniqueness give

$$\eta_\varepsilon = \min_{z \in K_\varepsilon} \{G(z) - G(u)\} > 0.$$

Once $\|G_n - G\|_\infty < \eta_\varepsilon/3$, no point of K_ε can minimise G_n, and hence $|u_n - u| < \varepsilon$. Applying this argument to every sequence $\beta_n \in [0, \infty)$ with $\beta_n \to 0$, every sequence $\beta_n \to \beta_0 \in (0, \infty)$, and every sequence $\beta_n \to \infty$ proves continuity of $\beta \mapsto \Phi_P^{\beta,\rho}(x)$ on the extended interval. This proves part (iii).

Part (iv): the finite-β solutions. Let $\bar{m} = \int_{\mathcal{A}} m(a)\nu(\mathrm{d}a)$. At $\beta = 0$, expanding the square gives

$$F_0(w) = \int_{\mathcal{A}} v(a)\nu(\mathrm{d}a) + \mathrm{Var}_\nu\{m(A)\} + (w - \bar{m})^2.$$

Its unique minimiser is $\bar{m}$, which proves (3.14).

Now let $0 < \beta < \infty$ and, for a candidate w, define the tilted probability measure

$$\omega_{\beta,x,w}(\mathrm{d}a) = \frac{\exp\{\beta r(a, w)\}\nu(\mathrm{d}a)}{\int_{\mathcal{A}} \exp\{\beta r(a', w)\}\nu(\mathrm{d}a')}.$$

To justify differentiation, set

$$Z(w) = \int_{\mathcal{A}} e^{\beta r(a,w)}\nu(\mathrm{d}a), \qquad N(w) = \int_{\mathcal{A}} m(a)e^{\beta r(a,w)}\nu(\mathrm{d}a).$$

On any compact interval $J \subset \mathbb{R}$, the functions $m(a)$, $r(a, w)$, and

$$\partial_w r(a, w) = 2\{w - m(a)\}$$

are bounded uniformly over $(a, w) \in \mathcal{A} \times J$. It follows that $e^{\beta r(a,w)}$, its derivative $2\beta\{w - m(a)\}e^{\beta r(a,w)}$, and the same two functions multiplied by $m(a)$ all have finite constant envelopes on $\mathcal{A} \times J$. The dominated differentiation theorem therefore applies separately to Z and N. Since $F_\beta(w) = \beta^{-1} \log Z(w)$ and $Z(w) > 0$, it gives

$$F'_\beta(w) = 2\left\{w - \int_{\mathcal{A}} m(a)\omega_{\beta,x,w}(\mathrm{d}a)\right\}.$$

The quotient rule applied to $N(w)/Z(w)$, using the same dominated derivatives, gives

$$\frac{\mathrm{d}}{\mathrm{d}w}\int_{\mathcal{A}} m(a)\omega_{\beta,x,w}(\mathrm{d}a) = -2\beta \operatorname{Var}_{\omega_{\beta,x,w}}\{m(A)\},$$

so

$$F''_\beta(w) = 2 + 4\beta \operatorname{Var}_{\omega_{\beta,x,w}}\{m(A)\} > 0.$$

The first-order condition at the unique minimiser, with the definition of $\omega_{\beta,x,w}$ substituted directly, is precisely (3.15), because (3.32) permits $\bar{r}_P$ to be replaced by r_P in both ρ_x-integrals. It is the unique fixed point because its left-minus-right side is $F'_\beta/2$ and is strictly increasing. At L this difference is non-positive, and at U it is non-negative, which also proves the bisection justification used in Section 3.4.2. This proves part (iv).

Part (v): the worst-case target. The representation obtained in part (i) can also be written as

$$F_\infty(w) = \max_{a \in \mathcal{A}}\left[v(a) + \{m(a) - w\}^2\right].$$

Taking its unique minimiser proves (3.16). This objective contains no occurrence of ν, so its minimiser is independent of the particular full-support reference measure. This proves part (v). □

Proof of Corollary 3.4.3. For the fixed x, abbreviate $m(a) = m_P(a, x)$ and $v(a) = v_P(a, x)$. Define

$$r^\circ_x(a, w) = v(a) + \{m(a) - w\}^2, \qquad F_\beta(w) = \mathcal{R}_{\beta,\rho,x}\big(r^\circ_x(\,\cdot\,, w)\big).$$

By (3.32), F_β is the objective defining $\Phi_P^{\beta,\rho}(x)$. If $v(a)$ is constant in a, the worst-case objective differs by a constant from

$$\max_a\{m(a) - w\}^2 = \max\big\{(\min_a m(a) - w)^2, (\max_a m(a) - w)^2\big\}.$$

The equality holds because every $m(a)$ lies between its minimum and maximum, and the largest distance from w is attained at one of those endpoints. The unique minimiser of the displayed upper envelope is the midpoint of the two extreme means, proving part (i).

For part (ii), suppose $\mathcal{A} = \{a_-, a_+\}$ and put $\Delta = m_+ - m_- > 0$. Equating the two quadratic risks gives

$$w_{\mathrm{eq}} = \frac{m_- + m_+}{2} + \frac{v_+ - v_-}{2(m_+ - m_-)}.$$

Indeed, writing $r_\pm(w) = v_\pm + (m_\pm - w)^2$ gives

$$r_-(w) - r_+(w) = 2\Delta(w - w_{\mathrm{eq}}).$$

Thus the upper envelope equals r_+ to the left of w_{eq} and r_- to its right. By Theorem 3.4.2(ii), its minimiser lies in $[m_-, m_+]$. If $w_{\mathrm{eq}} < m_-$, the envelope equals r_- throughout this interval, and $r'_-(w) = 2(w - m_-) \geq 0$ there, so it is minimised at m_-. If $w_{\mathrm{eq}} > m_+$, it equals r_+ throughout the interval and is minimised at m_+ because $r'_+(w) = 2(w - m_+) \leq 0$ there. Finally, if $w_{\mathrm{eq}} \in [m_-, m_+]$, then $r'_+(w) \leq 0$ on $[m_-, w_{\mathrm{eq}}]$ and $r'_-(w) \geq 0$ on $[w_{\mathrm{eq}}, m_+]$. The upper envelope therefore decreases up to w_{eq} and increases after it. These three cases give exactly (3.17).

If the means coincide at m_0, (3.8) gives

$$\mathcal{R}_{\beta,\rho,x}\big(v_P(\,\cdot\,, x) + (m_0 - w)^2\big) = (m_0 - w)^2 + \mathcal{R}_{\beta,\rho,x}\big(v_P(\,\cdot\,, x)\big).$$

The second term is independent of w, proving part (iii). Under equal reference weights and equal variances, reflection about the midpoint swaps the two state risks and leaves their aggregate unchanged:

$$F_\beta(m_- + m_+ - w) = F_\beta(w), \qquad \beta \in [0, \infty].$$

Uniqueness therefore forces the minimiser to be the midpoint, proving part (iv).

□

3.A.4 Proofs for the diagnostic

Proof of Theorem 3.5.1. Put $K(w, \mathrm{d}y) = P_0^{Y|W=w}(\mathrm{d}y)$ and $a_{n,m,i} = \pi_{n,m,i}/m$. Each $a_{n,m,i}$ is $\mathcal{H}_n$-measurable. Because $(Y_j, W_j)_{j=1}^n$ are i.i.d. and $\mathcal{V}$ is independent of $\mathcal{D}_n$, the conditional law of Y_i given $\mathcal{H}_n$ is $K(W_i, \cdot)$. Hence the definitions of $\widetilde{Q}_{n,m}$ and $Q_{n,m}$, iterated conditioning, and independence of W_i and $\mathcal{V}$ give, for every non-negative measurable h,

$$Q_{n,m}h = \sum_{i=1}^{n} \mathbb{E}\{a_{n,m,i} h(Z_i) \mid \mathcal{V}\} = \int h(y, w) K(w, \mathrm{d}y)\, g_{n,m}(w) P_0^W(\mathrm{d}w),$$

where $g_{n,m}(w) = \sum_{i=1}^n \mathbb{E}[a_{n,m,i} \mid W_i = w, \mathcal{V}]$. On the standard Borel observation space, a countable determining class and the monotone-class theorem make this factorisation simultaneous almost surely for all non-negative measurable h.

Every distinct tuple has positive conditional probability under (3.20), so each $a_{n,m,i} > 0$ almost surely. Thus $\mathbb{E}[a_{n,m,i} \mid W_i, \mathcal{V}] > 0$ almost surely. Since W_i is

independent of $\mathcal{V}$, Fubini's theorem implies that, for almost every realization of $\mathcal{V}$, $g_{n,m}(w) > 0$ for P_0^W-almost every w. Taking $h \equiv 1$ shows that $g_{n,m}$ integrates to one. The displayed identity therefore gives $Q_{n,m}^{Y|W} = K$ and $Q_{n,m}^W = g_{n,m} P_0^W \sim P_0^W \sim \mu^{A,X}$, so $Q_{n,m} \in \mathcal{Q}(P_0)$.

When $m = n$, $(i_1, \dots, i_n)$ is a permutation of $(1, \dots, n)$. Consequently, $\widehat{Q}_{n,n} = P_n$ almost surely. □

Proof of Proposition 3.5.2. Put $K(w, \mathrm{d}y) = P_0^{Y|W=w}(\mathrm{d}y)$. For $m < n$, define the empirical distribution of the observations Z_i with $S_{n,m,i} = 0$ by

$$\widehat{R}_{n,m} = \frac{1}{n-m} \sum_{i=1}^{n} (1 - S_{n,m,i}) \delta_{Z_i}, \qquad R_{n,m} = \mathbb{E}[\widehat{R}_{n,m} \mid \mathcal{V}].$$

Fix i. Since $m < n$, there is a distinct m-tuple containing only indices in $\{1, \dots, n\} \setminus \{i\}$. Its numerator and the normalising denominator in (3.20) are strictly positive. Hence Z_i has strictly positive conditional probability of being excluded, so $1 - \pi_{n,m,i} > 0$. Put

$$b_{n,m,i} = \frac{1 - \pi_{n,m,i}}{n-m}, \qquad \bar{g}_{n,m}(w) = \sum_{i=1}^{n} \mathbb{E}[b_{n,m,i} \mid W_i = w, \mathcal{V}].$$

Equation (3.19) gives

$$\mathbb{E}[\widehat{R}_{n,m} \mid \mathcal{G}_n] = \sum_{i=1}^{n} b_{n,m,i} \delta_{Z_i}.$$

Because each $b_{n,m,i}$ is $\mathcal{H}_n$-measurable, iterated conditioning gives, for every nonnegative measurable h,

$$\begin{aligned} R_{n,m} h &= \sum_{i=1}^{n} \mathbb{E}\{b_{n,m,i} h(Z_i) \mid \mathcal{V}\} \\ &= \sum_{i=1}^{n} \mathbb{E}\left[b_{n,m,i} \int h(y, W_i) K(W_i, \mathrm{d}y) \mid \mathcal{V} \right] \\ &= \int h(y, w) \bar{g}_{n,m}(w) P_0^W(\mathrm{d}w) K(w, \mathrm{d}y). \end{aligned}$$

Moreover, $\bar{g}_{n,m} > 0$ P_0^W-almost surely. Since exactly m indices are selected,

$$\sum_{i=1}^{n} b_{n,m,i} = \frac{n - \sum_{i=1}^{n} \pi_{n,m,i}}{n-m} = \frac{n-m}{n-m} = 1.$$

Independence of $\mathcal{V}$ and each W_i then gives

$$\int \bar{g}_{n,m}(w) P_0^W(\mathrm{d}w) = \sum_{i=1}^{n} \mathbb{E}[b_{n,m,i} \mid \mathcal{V}] = 1.$$

Hence $R_{n,m} \in \mathcal{Q}(P_0)$. The pathwise identity

$$P_n = \frac{m}{n} \widehat{Q}_{n,m} + \left(1 - \frac{m}{n}\right) \widehat{R}_{n,m}$$

holds almost surely. Because $\mathcal{D}_n$ is independent of $\mathcal{V}$, $\mathbb{E}[P_n \mid \mathcal{V}] = P_0$. Conditioning the pathwise identity on $\mathcal{V}$ therefore proves (3.21). Equation (3.21) also gives

$$d_{\mathrm{TV}}(Q_{n,m}, P_0) = \left(1 - \frac{m}{n}\right) d_{\mathrm{TV}}(Q_{n,m}, R_{n,m}) \leq 1 - \frac{m}{n},$$

which proves (3.22). □

Proof of Theorem 3.5.3. For each r, write $n = n_r$ and $m = m_r$, and let $W_{n,m,j}$ denote $W_{n,m}$ evaluated at the jth evaluation observation. Conditional on $\mathcal{T}_{n,m}$, these variables are i.i.d. within row r. On $\{0 < \sigma_{n,m} < \infty\}$,

$$\begin{aligned}
&\frac{\sqrt{r}}{\sigma_{n,m}} \left[\widehat{\mathrm{cov}}_r\{\widehat{D}_{n,m}(X), A\} - \mathrm{cov}_{P_0}\{D_{n,m}(X), A\}\right] \\
&\quad = \frac{1}{\sqrt{r}\,\sigma_{n,m}} \sum_{j=1}^{r} \{W_{n,m,j} - \mathbb{E}_{P_0} W_{n,m}\} \\
&\qquad - \frac{\sqrt{r}}{\sigma_{n,m}} \{\overline{\widehat{D}}_{n,m} - \mathbb{E}_{P_0} \widehat{D}_{n,m}(X)\}\{\overline{A}_r - \mathbb{E}_{P_0} A\} \\
&\qquad + \frac{\sqrt{r}}{\sigma_{n,m}} \mathrm{cov}_{P_0}\{\widehat{D}_{n,m}(X) - D_{n,m}(X), A\}. \qquad (3.33)
\end{aligned}$$

Let S_r denote the first term on the right-hand side of (3.33), put $G_r = \{0 < \sigma_{n,m} < \infty\}$, and let $L_r(\varepsilon)$ denote the left-hand side of (3.26). Fix any subsequence $r_\ell \to \infty$. Since $\Pr(G_r^c) \to 0$ and $L_r(1/k) \xrightarrow{p} 0$ for every $k \geq 1$, there is a further subsequence r_{ℓ_q} such that, almost surely, $G_{r_{\ell_q}}$ holds for all sufficiently large q and $L_{r_{\ell_q}}(1/k) \to 0$ for every k. By monotonicity, the latter implies $L_{r_{\ell_q}}(\varepsilon) \to 0$ for every $\varepsilon > 0$.

On G_r, define

$$\xi_{rj} = \frac{W_{n,m,j} - \mathbb{E}_{P_0} W_{n,m}}{\sqrt{r}\,\sigma_{n,m}}, \qquad 1 \leq j \leq r.$$

Conditional on $\mathcal{T}_{n,m}$, the variables $\xi_{r1}, \ldots, \xi_{rr}$ are i.i.d., each with mean zero and variance $1/r$; moreover, $S_r = \sum_{j=1}^{r} \xi_{rj}$ and their conditional Lindeberg sum at threshold ε is $L_r(\varepsilon)$. Consequently, for almost every outcome in the event just constructed, the ordinary Lindeberg–Feller theorem applied to the conditional distributions of $\xi_{r1}, \ldots, \xi_{rr}$ gives, along r_{ℓ_q} and for every bounded continuous $\varphi : \mathbb{R} \to \mathbb{R}$,

$$\mathbb{E}\{\varphi(S_r) \mid \mathcal{T}_{n,m}\} \longrightarrow \int \varphi(z)\,\mathcal{N}(0,1)(\mathrm{d}z) \qquad \text{almost surely.}$$

Because every subsequence $r_\ell \to \infty$ has such a further subsequence, the subsequence criterion yields

$$\mathbb{E}\{\varphi(S_r) \mid \mathcal{T}_{n,m}\} \xrightarrow{p} \int \varphi(z)\,\mathcal{N}(0,1)(\mathrm{d}z)$$

as $r \to \infty$ for the prescribed sequence $(n_r, m_r)_{r \geq 1}$. The conditional expectations are uniformly bounded, so the convergence also holds in L^1; taking expectations gives

$S_r \xrightarrow{d} \mathcal{N}(0,1)$. The second term in (3.33) is $o_p(1)$ by (3.27), and the third is $o_p(1)$ by (3.29). Finally, (3.28) gives $\widehat{\sigma}_{n,m}/\sigma_{n,m} \xrightarrow{p} 1$. Together with $\Pr(0 < \sigma_{n,m} < \infty) \to 1$, Slutsky's theorem proves (3.30). It also implies $\Pr(\widehat{\sigma}_{n,m} = 0) \to 0$, so the zero-denominator convention does not affect the limit.

Under the covariance null (3.25), the centring term in (3.30) is zero. Outside the event $\{\widehat{\sigma}_{n,m} = 0\}$, whose probability tends to zero, the rejection event is exactly that the absolute studentised statistic exceeds $z_{1-\alpha/2}$. The continuity of the standard normal distribution therefore gives $\Pr(\text{reject}) \to \alpha$ for the rule in (3.31). □

3.B Estimating optimal AX-invariant estimators

3.B.1 The partially linear model

Assume that A is a one-dimensional binary random variable, i.e. $\mathcal{A} = \{0,1\}$. We will consider a partially linear model of the form

$$\begin{aligned} Y &= \theta A + h(X) + \varepsilon, \\ A &= w(X) + \varepsilon_A, \end{aligned} \quad \text{with} \quad \begin{aligned} \mathbb{E}_{P_0}[\varepsilon \mid A, X] &= 0, \\ \mathbb{E}_{P_0}[\varepsilon_A \mid X] &= 0, \end{aligned}$$

for $\theta \in \mathbb{R}$ and measurable functions $h : \mathcal{X} \to \mathbb{R}$ and $w : \mathcal{X} \to \mathbb{R}$.

In this model, it holds that

$$\frac{\mathrm{cov}_{P_0}(Y - h(X), A - w(X))}{\mathrm{cov}_{P_0}(A, A - w(X))} = \frac{\mathrm{cov}_{P_0}(\theta A + \varepsilon, \varepsilon_A)}{\mathrm{cov}_{P_0}(A, \varepsilon_A)} = \theta + \frac{\mathrm{cov}_{P_0}(\varepsilon, \varepsilon_A)}{\mathrm{cov}_{P_0}(A, \varepsilon_A)} = \theta, \quad (3.34)$$

where in the last step we used $\mathrm{cov}_{P_0}(\varepsilon, \varepsilon_A) = \mathbb{E}_{P_0}[\varepsilon\varepsilon_A] = \mathbb{E}_{P_0}[\mathbb{E}_{P_0}[\varepsilon\varepsilon_A \mid A, X]] = \mathbb{E}_{P_0}[\varepsilon_A \mathbb{E}_{P_0}[\varepsilon \mid X, A]] = 0$.

The coefficient θ can be estimated by plugging estimators of h and w into the empirical analogue of the covariance ratio in (3.34); the construction below uses sample splitting.

Remark 3.B.1. If ε_A is small it is possible to observe the estimator $\widehat{\mathrm{cov}}_{P_0}(\varepsilon, \varepsilon_A)$ of similar size to $\widehat{\mathrm{cov}}_{P_0}(A, \varepsilon_A)$, resulting in an unstable estimator of θ.

Let $u^A = \frac{1}{2}(\delta_0 + \delta_1)$, then

$$\Phi_{P_0}^{0,u^A}(x) = h(x) + \frac{\theta}{2}.$$

If, in addition, $v_{P_0}(0,x) = v_{P_0}(1,x)$ for P_0^X-almost every x, then Corollary 3.4.3 gives $\Phi_{P_0}^{\beta,u^A}(x) = h(x) + \theta/2$ for every $\beta \in [0,\infty]$, including the worst-case endpoint $\Phi_{P_0}^{\infty}$. We propose the estimator $\hat{h}(x)$ of $h(x)$ as the one obtained by regressing $Y_i - \hat{\theta}_n A_i$ on features X_i, $i = 1, \ldots, n$, where $\hat{\theta}_n$ is determined by the double machine learning method as described below.

Partition the observed data into K disjoint subsets indexed by I_k such that $\bigcup_{k=1}^K I_k = \{1, \ldots, n\}$ and $\#I_k \approx \frac{n}{K}$, for all $k = 1, \ldots, K$. Denote by I_k^C the complement of I_k, i.e. $I_k^C = \{1, \ldots, n\} \setminus I_k$.

Recall that A is assumed binary such that,

$$\Phi_{P_0}^{0,u^A}(x) = \frac{1}{2}m_{P_0}(x,0) + \frac{1}{2}m_{P_0}(x,1).$$

Let $\tilde{\Phi}_{P_0}^{0,u^A}(x)$ be the plug-in estimator of $\Phi_{P_0}^{0,u^A}(x)$ on the data indexed by I_k^C, i.e.

$$\tilde{\Phi}_{P_0}^{0,u^A}(x)_k(x) = \frac{\#\{i \in I_k^C \mid A_i = 0\}}{\#\{i \in I_k^C\}}\tilde{m}_{P_0,k}(x,0) + \frac{\#\{i \in I_k^C \mid A_i = 1\}}{\#\{i \in I_k^C\}}\tilde{m}_{P_0,k}(x,1),$$

where $\tilde{m}_{P_0,k}(x,j)$, $j = 0,1$, $k = 1,\ldots,K$, is the estimator of $m_{P_0}(x,j)$ obtained by training a machine learning model to Y_i with features X_i, $i \in I_k^C$ on the event that $A_i = j$.

Let $\tilde{w}_k$, $k = 1,\ldots K$, be the plug-in estimator of w on the data indexed by I_k^C estimated by training a machine learning models to A_i with features X_i, $i \in I_k^C$.

Let $\widehat{\mathrm{cov}}_{I_k}$ be the empirical covariance over the observations indexed by I_k. Now, using the relation presented in Equation (3.34),

$$\hat{\theta}_{k,n} = \frac{\widehat{\mathrm{cov}}_{I_k}(Y_{\{I_k\}} - \tilde{h}_k(X_{\{I_k\}}), A_{\{I_k\}} - \tilde{w}_k(X_{\{I_k\}}))}{\widehat{\mathrm{cov}}_{I_k}(A_{\{I_k\}}, A_{\{I_k\}} - \tilde{w}_k(X_{\{I_k\}}))}, \qquad k = 1,\ldots,K.$$

We then estimate θ by

$$\widehat{\theta}_n = \frac{1}{K}\sum_{k=1}^{K}\widehat{\theta}_{k,n}.$$

3.B.2 Efficient estimation using double machine learning in the binary case

For binary A and $u^A = \frac{1}{2}(\delta_0 + \delta_1)$,

$$\Phi_{P_0}^{0,u^A}(x) = \frac{1}{2}\{m_{P_0}(0,x) + m_{P_0}(1,x)\}.$$

If $v_{P_0}(0,x) = v_{P_0}(1,x)$, the same midpoint equals $\Phi_{P_0}^{\beta,u^A}(x)$ for every $\beta \in [0,\infty]$, including $\Phi_{P_0}^{\infty}(x)$.

We wish to obtain a debiased estimate of the above quantity. To motivate this, first introduce

$$\varphi^0(Y,A,X) = m_{P_0}(0,X) + \frac{(1-A)(Y - m_{P_0}(0,X))}{1 - P(A=1|X)}, \quad \text{and}$$
$$\varphi^1(Y,A,X) = m_{P_0}(1,X) + \frac{A(Y - m_{P_0}(1,X))}{P(A=1|X)}.$$

Then

$$\mathbb{E}_{P_0}\left[\varphi^a(Y,A,X)|X=x\right] = m_{P_0}(a,x), \quad a = 0,1.$$

This suggest using the double machine learning method by first estimatíng $\phi^0 + \phi^1$ and then regressing on X.

Using the same approach as above, we split the data into K folds, I_k, $k = 1, \ldots, K$. On the complement I_k^C we calculate the plug-in estimators $\tilde{m}^k_{P_0}(0, X)$, $\tilde{m}^k_{P_0}(1, X)$ and $\tilde{P}^k_0(A = 1|X)$, $k = 1, \ldots, K$, e.g. by training regression/classification machine learning models. On the remaining fold we then estimate for $iin I_k$

$$\hat{\phi}^{0,k}_i(Y, A, X) = \tilde{m}^k_{P_0}(0, X_i) + \frac{(1 - A_i(Y_i - \tilde{m}^k_{P_0}(0, X_i))}{1 - \tilde{P}^k_0(A = 1|X_i)},$$
$$\hat{\phi}^{1,k}_i(Y, A, X) = \tilde{m}^k_{P_0}(1, X_i) + \frac{A_i(Y_i - \tilde{m}^k_{P_0}(1, X_i))}{\tilde{P}^k_0(A = 1|X_i)}.$$

From which we obtain the kth estimator $\hat{m}^k_{P_0}(0, x)+\hat{m}^k_{P_0}(1, x)$ by regressing $\hat{\phi}^{0,k}+\hat{\phi}^{1,k}$ on X, e.g. using machine learning. This gives us the final estimator

$$\Phi^{0,u^{\mathcal{A}},\texttt{DML}}_{P_0}(x) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{2}\left(\hat{m}^k_{P_0}(0, x) + \hat{m}^k_{P_0}(1, x)\right).$$

3.B.3 Estimating the conditional variance

In order to estimate the full solution given in Corollary 3.4.3 via a plugin solution, we need to estimate the conditional variance, $\mathrm{Var}_P[Y|A, X]$. We propose to train a random forest model to Y with features (A, X) keeping track of the observations landing in each leaf for every tree during training. The variance is then estimated by calculating the variance within each leaf and taking the mean over all trees. The splitting criteria instead of CART should be based on the conditional variance.

Chapter 4

Modelling Non-Life Insurance Customer Portfolio Changes via Hawkes Processes

Abstract

Customer churn in non-life insurance is driven by the evolution of customers' product portfolios, where purchases and cancellations occur irregularly in continuous time and may cluster around periods of increased customer engagement. We propose a bivariate Hawkes process for modelling customer churn and portfolio evolution where purchases and cancellations are represented as mutually-exciting Hawkes processes, allowing recent actions to increase the short-term likelihood of further portfolio changes. The model captures both clustering within event types and dependence between purchases and cancellations, while marks and covariates can represent product and portfolio characteristics. Its low-dimensional structure provides a simple continuous-time alternative to discrete-state and time-varying survival models. We illustrate that the model performs comparably to literature benchmarks using a real-world data set from the Wisconsin Local Government Property Insurance Fund. We also implement an algorithm for simulating event histories from the proposed model and conduct a simulation study using the generated data.

4.1 Introduction

Customer relationships in non-life insurance often involve households with multiple renewable policies. A household may add new policies, cancel existing ones while retaining others, or eventually terminate the relationship with the insurer altogether. We therefore distinguish between product-level churn, where a customer drops a particular product, and full churn, where the customer no longer holds any active product with the insurer. Hence, full churn is the terminal state of the customer

relationship, whereas product purchases and product cancellations are intermediate events that may reveal changes in engagement, risk-appetite, price sensitivity, and customer needs.

Understanding these dynamics is valuable both commercially and actuarially. From a customer management perspective, insurers want to identify policyholders for whom a retention action, discount, product recommendation, or service intervention is likely to be effective. From an actuarial perspective, changes in a customer's portfolio also change risk exposure as well as premium income and claims experience distribution.

Much of the churn-prediction literature treats churn as a supervised classification problem, where a customer is labelled as churned or not churned over a specified prediction horizon. If the probability of full churn within a fixed time-horizon is the target then this is a reasonable model framework. Still, even within this seemingly simple model, it is generally not clear how to incorporate the possibly rich and valuable historical information available about a customer. Going beyond simple full churn prediction over a fixed-time horizon, it can be of interest to have a dynamic picture of how full churn probabilities evolve over time and also how more granular portfolio changes are expected to evolve over time. However, the problem of how to incorporate rich customer information remains.

Different approaches to representing customer history have been proposed in non-insurance churn applications. In mobile-game churn prediction, Guo et al. (2024) use a fixed-window classification design: activity observed in an observation window is summarised into game-log-based features and used to predict inactivity in a subsequent churn window. For related online-service interaction data, Khodadadi et al. (2022) replace the binary churn label by a return-time problem, combining temporal point processes with recurrent neural networks to update the representation of user history and predict when the user will return. Yin et al. (2025) use an attention-based neural point process for digital-content consumption, modelling marked event sequences in order to predict future activity times, behaviour combinations, and consumption quantities. These papers illustrate a spectrum of modelling choices, from engineered fixed-window summaries to learned sequential representations of high-frequency digital interactions.

Non-life insurance data have a different structure. Customer interactions are typically of lower-frequency than in digital platforms and portfolio states are constrained by a finite set of products. Moreover, product purchases and cancellations are not some generic activity indicators but directly change the insured portfolio. Our goal is therefore not to import high-dimensional digital-interaction models directly, but to construct a event-history representation tailored to insurance portfolio evolution.

In the non-life insurance churn literature, Dong et al. (2022) model customer churn in non-life insurance through transitions between coverage states observed on a discrete time grid. Such models are well suited to periodically observed insurance

portfolios and explicitly represent movements between portfolio states. But the model can quickly become too complex if the number of states, i.e. possible portfolio configurations, is increased and the use of a longer customer history is desired.

Within a survival framework, Brockett et al. (2008) and Guillén et al. (2009) study household-level insurance loyalty and the time from a first dropped coverage to total customer defection. Guillén et al. (2012) extend this survival perspective by allowing time-dependent covariates and time-varying effects in an extended Cox model. Their empirical observation is particularly important here: after a customer drops one coverage, the risk of dropping another coverage is not constant over time, and the period shortly after a cancellation shows especially an increased risk. In their framework, this post-cancellation behaviour is captured through time-varying covariate effects. Our approach can be viewed as a natural refinement of this idea.

We propose a continuous-time model for portfolio evolution based on a bivariate Hawkes process. One component records purchases and the other records cancellations. The model is designed to capture short-term dependence in portfolio actions. For cancellations, this is motivated by the insurance-loyalty evidence of Brockett et al. (2008) and Guillén et al. (2012), who show that the period following an initial cancellation is especially informative for subsequent defection risk. In addition, our own empirical analysis on some confidential data suggests that both purchases and cancellations may cluster in time. In short, our proposed model is designed to capture the empirical observation that a customer is more likely to buy (drop) more products if they have just bought (dropped) others. Furthermore, mutual excitement includes the possibility that buying products might also make dropping products shortly after more likely and vice versa, e.g. replacing one product by another. The intuition here is that customers may be more likely to interact with the insurance company when they are first reminded that they have insurance and are unlikely to think about insurance in their daily lives. Compared to Guillén et al. (2012) our Hawkes specification represents post-drop feedback through a small number of excitation parameters and a pre-chosen decay function, rather than through a full set of time-varying coefficient functions that are more difficult to estimate.

The proposed model is related to recent Hawkes-based work in insurance and actuarial science. Lesage et al. (2024) use a multivariate Hawkes process for insurance recommendation, where subscriptions and life events help explain the evolution of customers' insurance needs. Jung et al. (2025) use a most-recent-event Hawkes process in a multi-state transition setting. Our focus is different. We use a lower-dimensional buy/drop Hawkes specification to study churn and portfolio evolution in non-life insurance. Product identity and current portfolio composition can still enter the model through marks and covariates, but the excitation structure is kept deliberately simple. This avoids the rapid growth in parameters that would arise from assigning a separate Hawkes component to every product or event type.

4.2 Preliminaries for mutually exciting Hawkes processes

4.2.1 Counting, intensity, and compensator processes

Given random arrival times $0 < T_1 < T_2 < \cdots$, the stochastic process $(N(t))_{t\geq 0}$

$$N(t) = \#\{\ell : T_\ell \leq t\},$$

is the counting process counting one every time an arrival occurs. Assuming that the limit below exists, the corresponding intensity process, $(\lambda(t))_{t\geq 0}$, describes the instantaneous rate of arrival,

$$\lambda(t) := \lim_{\Delta t \to 0^+} \frac{\mathbb{E}[N(t+\Delta t) - N(t)|\mathcal{F}(t)]}{\Delta t}.$$

Here, $\mathcal{F}(t)$ is the filtration generated by $(N(t))_{t\geq 0}$, i.e., $\mathcal{F}(t)$ is the history of $(N(t))_{t\geq 0}$. If the intensity is constant, $\lambda(t) \equiv \lambda$, then $(N(t))_{t\geq 0}$ is a homogenous Poisson process. Furthermore, the integral of $\lambda(t)$,

$$\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s,$$

is called the compensator (process).

4.2.2 Marked counting processes

Counting processes record when events occur, but in applications we usually also observe information attached to each event. For example, earthquakes have magnitudes and locations in addition to occurrence times. Such event-level information is called a mark. In our empirical data application, events are buy and drop bundles, and the mark records which coverage lines have been added or dropped.

Let each arrival time T_ℓ have an associated mark $M_\ell \in \mathcal{M}$. For a mark set $\mathcal{A} \subseteq \mathcal{M}$, the marked counting process is

$$N(t, \mathcal{A}) = \sum_{\ell : T_\ell \leq t} \mathbb{1}\{M_\ell \in \mathcal{A}\}, \qquad t \geq 0.$$

For simplicity, if we imagine the marks are scalar real values $M_\ell \in \mathbb{R}$, then the intensity process describing the arrival rate at time t of an arrival with a mark m is

$$\begin{aligned}\lambda(t, m) &:= \lim_{\Delta t \to 0^+, \Delta m \to 0^+} \frac{\mathbb{E}[N(t+\Delta t, [m+\Delta m)) - N(t, [m+\Delta m))|\mathcal{F}(t)]}{\Delta t\, \Delta m} \\ &= \lambda(t) f(m|t).\end{aligned}$$

Here, $\lambda(t)$ is the intensity of the unmarked counting process and $f(m|t)$ is the density of mark m at time t. For both quantities it is therefore assumed that conditioning on the filtration, $\mathcal{F}(t)$, generated by $\{(T_\ell, M_\ell) : T_\ell < t\}$ results in functions that

only depend on t. Later we also add predictable covariates. The event-time and mark components may use the same or different covariate vectors; below, X denotes the timing design and Z the mark design. The description above generalises in the expected way to marks which are vector-valued or to more general $\mathcal{M}$ mark spaces. For a finite mark space the formulation becomes

$$\lambda(t, m) = \lambda(t)p(m \mid t), \qquad \sum_{m \in \mathcal{M}} p(m \mid t) = 1.$$

In general, the unmarked event-time intensity $\lambda(t)$ can be estimated without specifying the conditional mark density or probability, provided the mark is observed. If a mark model with its own parameters is added, the marked log-likelihood contains the event-time likelihood plus an additional term

$$\sum_{\ell: T_\ell \leq T} \log p(M_\ell \mid T_\ell).$$

Unless parameters are deliberately shared between the event-time and mark models, this additional term can be estimated separately conditional on the observed event times and possible covariate values. Hence, to estimate λ, one does not need to specify a model for p. However, a conditional mark model is needed if one wants to simulate future paths. The coverage-set mark model used in the data application is specified in Section 4.3.2.

4.2.3 Univariate Hawkes processes

A counting process is said to be a Hawkes process if its intensity process is given by

$$\lambda(t) = \mu + \eta \sum_{\ell: T_\ell < t} \nu(t - T_\ell), \tag{4.1}$$

where μ is the background rate, $\eta > 0$ is called the branching ratio, and $\nu : (0, \infty) \to (0, \infty)$ is a probability density function, i.e. $\int_0^\infty \nu(u)\mathrm{d}u = 1$, called the excitation function. It is usually strictly decreasing. That is, $(N(t))_{t \geq 0}$ will have intensity μ until an event occurs after which the intensity will jump with $\eta\nu(0+)$ and then decay back towards μ according to the function ν until another event occurs, and so on. This is the most fundamental property of a Hawkes process, often called the self-exciting property, where an arrival will temporarily increase the chance of more arrivals in the near future, causing a cluster of events.

A Hawkes process and its parameters can also be understood through its immigration–birth representation. In this interpretation, a homogeneous Poisson process with rate μ generates baseline events, called immigrants. Each event then generates an expected number η of direct additional events, called offspring, not counting further events subsequently generated by them. The timing of the offspring is governed by the kernel ν. This representation dates back to Hawkes and Oakes

(1974), though in modern statistical terms we would say that the Hawkes process is an example of a Poisson cluster process.

Note that, for $\eta < 1$, the geometric series implies that $\eta/(1-\eta)$ is the expected total number of offspring, including both direct and indirect offspring, caused by a single event.

4.2.4 Hawkes processes with inhomogeneous background rates

The Hawkes process is extremely adaptable, and basically every component of (4.1) has been generalised to match a wide variety of domains. In this work we generalise the intensity process by allowing the background rate λ to depend on t and a set of p time-dependent covariates $X(t) \in \mathbb{R}^p$,

$$\lambda(t; X(t)) = \mu(t; X(t)) + \eta \sum_{\ell: T_\ell < t} \nu(t - T_\ell)\,.$$

When there is no ambiguity, we will omit the dependency on the covariates from the notation, and will write $\lambda(t)$ to represent $\lambda(t; X(t))$.

4.2.5 Mutually Exciting Hawkes Processes

We also consider a multidimensional version of the Hawkes process, called "mutually exciting Hawkes processes". That is, we take $d \in \mathbb{N}$ Hawkes processes — denoted $(N^j(t))_{t\geq 0}$ with arrivals $(T_\ell^j)_{\ell\in\mathbb{N}}$ and intensity λ^j for $j = 1, \ldots, d$ — and update each intensity process so that each one is affected by arrivals to the other processes, as in

$$\lambda^j(t; X(t)) = \mu^j(t; X(t)) + \sum_{k=1}^{d} \eta^{jk} \sum_{\ell: T_\ell^k < t} \nu^{jk}(t - T_\ell^k), \quad j = 1, \ldots, d.$$

Here the first index of η^{jk} identifies the affected process and the second identifies the process in which the past event occurred. Later, for the sake of simplification, we will impose the assumption that the excitation function depends only on the affected process, i.e. $\nu^{jk} \equiv \nu^j$.

4.2.6 Likelihoods for Hawkes processes

For estimating the set of parameters θ that specifies the component intensities $\lambda^j(t),\ j = 1, \ldots, d$, we maximize the likelihood of the multivariate counting process $N(t) = \big(N^1(t), \ldots, N^d(t)\big)$ over an observation window $[0, T]$. For each component j, let $0 < T_1^j < T_2^j < \cdots$ denote the arrival times of the process N^j. Assuming that simultaneous jumps do not occur, the log-likelihood of the multivariate Hawkes process on $[0, T]$ is

$$\ell(\theta) = \sum_{j=1}^{d} \sum_{\ell: T_\ell^j \leq T} \log\Big\{\lambda^j(T_\ell^j)\Big\} - \sum_{j=1}^{d} \int_0^T \lambda^j(s)\,\mathrm{d}s.$$

4.3 Modelling churn in insurance via mutually exciting Hawkes processes

We now specialise the general multivariate Hawkes framework to customer churn in non-life insurance. The object of interest is not only whether a customer fully churns within a fixed prediction window, but the continuous-time sequence of portfolio changes before censoring or full churn. For household $i = 1, \dots, n$, let $S_i(t) \subseteq \mathcal{P}$ denote the set of active products at time t, where $\mathcal{P}$ is the set of products offered by the insurer. Full churn corresponds to $S_i(t) = \emptyset$, which is treated as an absorbing state.

4.3.1 Modelling the event-time log-likelihood

When entry is observed, time can be measured from the first policy purchase. For a left-truncated panel, as in the data application below, $t = 0$ instead denotes the first available observation.

The entry-observed case is illustrated in Figures 4.1 and 4.2. In calendar time, customers enter the portfolio at different dates and may subsequently buy or drop covers. When entry is available, we shift each history so that the first purchase occurs at time zero. We condition on this purchase rather than model its arrival: it is omitted from the event contribution to the likelihood, but retained in the Hawkes history and can therefore excite later buy and drop events. This is the convention used in the simulation study. Observations end at full churn or at the administrative end of follow-up, so a customer who has not fully churned is right-censored in time since first policy.

The data application has a different observation scheme. Many customers are already active when the annual panel begins, so their first observed portfolio is not their first purchase. We therefore condition on the portfolio at the first observation and do not insert an artificial buy event at time zero. This is a left-truncated history: earlier actions and their remaining excitation are unobserved. We later represent possible excess activity at this origin by the transient background component in (4.4); it provides an origin correction but does not recover the missing event history.

We assume that actions occur in continuous time and that distinct buy and drop actions do not occur at exactly the same time. To keep the indices distinct, throughout this section i indexes customers, j indexes the Hawkes component whose intensity is being modelled, k indexes the component in which a past event occurred, and ℓ indexes events within a component. Both component indices take values in $\mathcal{H} = \{\mathrm{b}, \mathrm{d}\}$, corresponding to buy and drop. Thus, for example, η^{jk} measures the effect of a type-k event on the type-j intensity.

We observe arrivals $(T_{i\ell}^{j})_{\ell \in \mathbb{N}}$, $j \in \mathcal{H}$, corresponding to buy and drop events and let

$$N_i^{\mathrm{b}}(t) = \#\{\ell : T_{i\ell}^{\mathrm{b}} \le t\}, \qquad N_i^{\mathrm{d}}(t) = \#\{\ell : T_{i\ell}^{\mathrm{d}} \le t\},$$

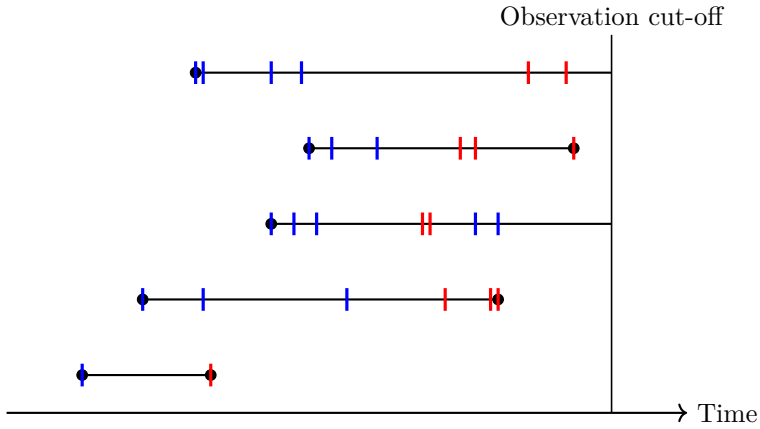


Figure 4.1: Entry-observed histories for five customers in calendar time. Blue and red ticks denote buy and drop events, respectively; the first filled point is the initial purchase. Histories reaching the observation cut-off are right-censored, whereas an earlier terminal drop represents full churn.

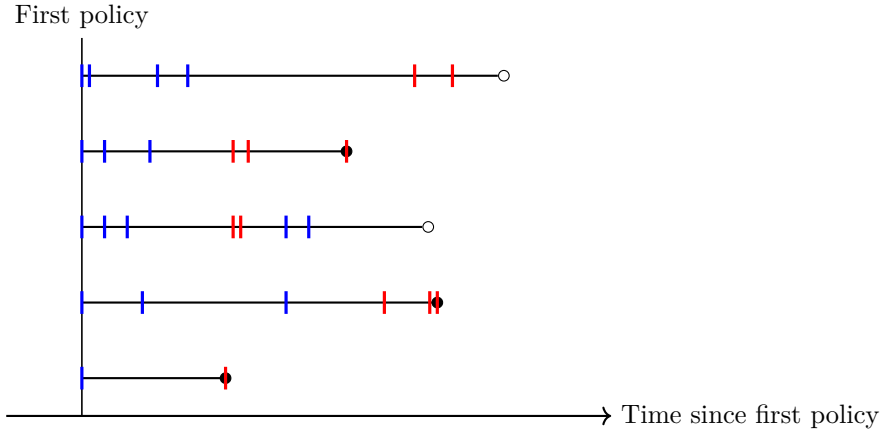


Figure 4.2: The same five entry-observed histories after shifting each initial purchase to time zero. Blue and red ticks denote buy and drop events, respectively; filled terminal points indicate full churn and open terminal points indicate administrative right-censoring.

be the corresponding counting processes. The counting processes themselves do not carry information about which products have been added or dropped, or about the current and historical portfolio composition.

A single event may change several products together. Its mark is therefore a non-empty subset $M_{i\ell}^{j} \subseteq \mathcal{P}$, rather than necessarily one product. At a buy event, $S_i(T_{i\ell}^{\mathrm{b}}) = S_i(T_{i\ell}^{\mathrm{b}}-) \cup M_{i\ell}^{\mathrm{b}}$; at a drop event, $S_i(T_{i\ell}^{\mathrm{d}}) = S_i(T_{i\ell}^{\mathrm{d}}-) \setminus M_{i\ell}^{\mathrm{d}}$. The event time drives the buy/drop intensity, while the exact changed subset is retained as the

mark.

We reiterate that we distinguish two event types. A buy event increases coverage by adding one or more products, while a drop event decreases coverage by cancelling one or more products. The affected subset is retained as a mark, while the Hawkes dimension is kept at two: one process for buys and one process for drops. While a higher-dimensional specification could assign a separate process to each product or life-event type, an m-dimensional Hawkes model requires m^2 excitation functions. With limited churn data this might be prohibitively large, and so the bivariate buy/drop formulation is intentionally kept simple. Product identity, current portfolio state, and customer characteristics can enter through marks and covariates.

Let C_i denote the right-censoring time and let

$$T_i^0 = \inf\{t > 0 : S_i(t) = \emptyset\}$$

be the time of full churn. The observed terminal time is

$$\tau_i = C_i \wedge T_i^0.$$

To account for censoring and feasibility of events, define the risk indicators

$$Y_i^{\mathrm{b}}(t) = \mathbf{1}\{t \le C_i,\ \emptyset \neq S_i(t-) \neq \mathcal{P}\}, \qquad Y_i^{\mathrm{d}}(t) = \mathbf{1}\{t \le C_i,\ S_i(t-) \neq \emptyset\}.$$

Thus, buying is switched off after censoring, full churn, or full coverage, while dropping is switched off after censoring or full churn.

The risk indicators encode the portfolio-state constraints illustrated in Figure 4.3. A customer can only buy if they are active and do not already hold all products, and can only drop if they hold at least one product. The drop that removes all remaining products is included as a drop event, after which churn is absorbing and both buy and drop intensities are switched off.

Extending the mutually exciting Hawkes-process framework introduced earlier to include the risk indicator, for $j \in \mathcal{H}$, we specify the conditional intensity

$$\lambda_i^j(t \mid \mathcal{F}_{i,t-}) = Y_i^j(t) \left[\mu_i^j\{t, X_i^j(t-)\} + \sum_{k \in \mathcal{H}} \eta^{jk} \sum_{\ell : T_{i\ell}^k < t} \nu^j(t - T_{i\ell}^k) \right]. \tag{4.2}$$

Here $X_i^j(t-)$ is the predictable timing-design vector and $\mathcal{F}_{i,t-}$ is the observed customer history just before time t. The covariates $X_i^j(t-)$ may contain both static and time-dependent customer covariates such as claims history, premiums, tenure, and calendar time, but also functions of the current portfolio state, for example

$$|S_i(t-)|, \qquad \mathbf{1}\{p \in S_i(t-)\}, \qquad p \in \mathcal{P},$$

or lower-dimensional summaries of these quantities. The first index of η^{jk} denotes the intensity being affected, while the second denotes the past event type. Hence,

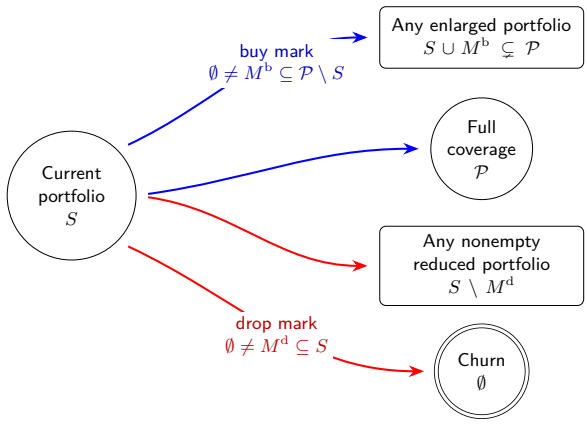


Figure 4.3: Feasible transitions from a current portfolio S. A buy may add any nonempty subset of products not currently held, while a drop may remove any nonempty subset of held products. Thus, a single marked event may move directly to any feasible union or set difference, including full coverage or churn; churn is absorbing.

η^{bb} and η^{dd} describe buy-to-buy and drop-to-drop self-excitation, whereas η^{bd} and η^{db} describe drop-to-buy and buy-to-drop cross-excitation.

This gives a compact way to combine a multi-state view of the portfolio with a low-dimensional Hawkes excitation structure. The affected subset at a buy or drop event can be treated as a mark. Thus, the model can distinguish, for example, dropping motor insurance from jointly dropping motor and pet insurance, without introducing a separate Hawkes process for every product-specific transition.

A convenient specification for the background intensity is

$$\mu_i^j\{t, X_i^j(t-)\} = \exp\{\beta_{\mu,0}^j + X_i^j(t-)^\top \beta_\mu^j\}, \tag{4.3}$$

where $\exp(\beta_{\mu,0}^j)$ is the component-specific long-run baseline rate. In the empirical studies below we use exponential excitation kernels,

$$\nu^j(u) = \gamma^j \exp(-\gamma^j u), \qquad u > 0,$$

so that the impact of a previous action is largest shortly after the event and then decays exponentially with rate γ^j towards zero. The model therefore allows that customer actions cluster in time. In practice, the excitation parameters η^{jk} are estimated from the data and may also be (partly) set to zero in simpler submodels.

An origin effect can be represented by augmenting the long-run log-linear background in (4.3) with an additive decaying term,

$$\mu_{i,\mathrm{tr}}^j(t) = \exp\{\beta_{\mu,0}^j + X_i^j(t-)^\top \beta_\mu^j\} + \exp(\xi_j)\exp(-\gamma^j t), \qquad t \geq 0. \tag{4.4}$$

Here ξ_j is a component-specific transient intercept. Meaning it models an additional initial intensity term shred with all customers that drops with the common

background rate γ_k. The transient term captures excess intensity at the origin that decays away over time. This can be useful for absorbing residual excitation from events before the modelled history begins.

Applying the likelihood from Section 4.2.6 to the bivariate censored process gives the individual event-time log-likelihood

$$\ell_i^{\text{time}}(\theta) = \sum_{j\in\mathcal{H}} \sum_{\ell: T_{i\ell}^j \le \tau_i} \log \lambda_i^j(T_{i\ell}^j \mid \mathcal{F}_{i,T_{i\ell}^j-}) - \sum_{j\in\mathcal{H}} \int_0^{\tau_i} \lambda_i^j(s \mid \mathcal{F}_{i,s-})\,\mathrm{d}s. \tag{4.5}$$

When the first purchase is used to define time zero, the likelihood is conditional on that initial event and its event contribution is omitted from the sum. With a left-truncated origin, it is instead conditional on the portfolio observed at that origin, and a transient term such as (4.4) may absorb residual pre-origin activity. The portfolio event-time log-likelihood is

$$\ell^{\text{time}}(\theta) = \sum_{i=1}^{n} \ell_i^{\text{time}}(\theta).$$

Several simpler models are nested in this specification. If $\eta^{\text{bb}} = \eta^{\text{bd}} = \eta^{\text{db}} = \eta^{\text{dd}} = 0$, the model reduces to a censored recurrent-event model with separate buy and drop background intensities. If $\eta^{\text{bd}} = \eta^{\text{db}} = 0$, then buys and drops may each be self-exciting, but neither process directly excites the other.

4.3.2 Modelling the conditional mark probabilities

The event-time intensity determines when a buy or drop occurs, but not which products are affected. Conditional on the event type, the mark selects an exact non-empty subset of products that is feasible for the portfolio immediately before the event. For a current portfolio $S \subseteq \mathcal{P}$, define

$$\mathcal{M}^{\text{b}}(S) = 2^{\mathcal{P}\setminus S} \setminus \{\emptyset\}, \qquad \mathcal{M}^{\text{d}}(S) = 2^{S} \setminus \{\emptyset\}. \tag{4.6}$$

These sets are conditional supports on the corresponding event risk set. A buy mark can contain only products not currently held, while a drop mark can contain only products currently held. In particular, dropping all products in S remains feasible and produces full churn. If $S = \mathcal{P}$, the buy support is empty and $Y_i^{\text{b}}(t) = 0$. If $S = \emptyset$, churn is absorbing and both risk indicators are zero.

As before, let $j \in \mathcal{H} = \{\text{b}, \text{d}\}$ be the event type: $j = \text{b}$ denotes a buy and $j = \text{d}$ a drop. The timing-design vector $X_i^j(t-)$ in (4.2) helps determine when a type-j event occurs. Conditional on such an event, let $Z_i^j(t-)$ denote the predictable mark-design vector used to determine which feasible bundle is affected. The current portfolio $S_i(t-)$ determines the feasible support and may also contribute coverage indicators or other summaries to either design.

To simplify notation, write $\lambda_i^j(t)$ for the event-time intensity in (4.2). Write β_p^j for the separate type-j mark-regression parameters. The marked intensity is

$$\lambda_i^j(t,m) = \lambda_i^j(t)\, p_j\Big\{m \mid Z_i^j(t-), S_i(t-); \beta_p^j\Big\}, \qquad j \in \mathcal{H}, \quad m \in \mathcal{M}^j(S_i(t-)). \tag{4.7}$$

By definition, p_j sums to one over the feasible support. It therefore distributes the already-specified type-j event-time intensity over the feasible bundles, and summing $\lambda_i^j(t,m)$ over those bundles recovers $\lambda_i^j(t)$. The event-time process remains bivariate even though the mark distinguishes many possible coverage bundles.

For each $j \in \mathcal{H}$, let $\mathcal{C}_j$ be the set of type-j bundles observed in the training data. For every class $m \in \mathcal{C}_j$, let $\beta_{p,0}^{j,m}$ and $\beta_p^{j,m}$ be its intercept and slope vector, and collect these parameters as

$$\beta_p^j = \left\{ \left(\beta_{p,0}^{j,m}, \beta_p^{j,m}\right) : m \in \mathcal{C}_j \right\}.$$

For a realized value z of the pre-event vector $Z_i^j(t-)$, define the class score

$$\eta_{j,m}(z) = \beta_{p,0}^{j,m} + z^\top \beta_p^{j,m}, \qquad m \in \mathcal{C}_j. \tag{4.8}$$

At portfolio S, only the classes in

$$\mathcal{A}_j(S) = \mathcal{C}_j \cap \mathcal{M}^j(S) \tag{4.9}$$

are both represented in the training data and feasible. Thus, the choice set varies with the pre-event portfolio. On the type-j event risk set, define the conditional mark distribution by

$$p_j(m \mid z, S; \beta_p^j) = \begin{cases} \dfrac{\exp\{\eta_{j,m}(z)\}}{\sum_{r \in \mathcal{A}_j(S)} \exp\{\eta_{j,r}(z)\}}, & m \in \mathcal{A}_j(S), \quad \mathcal{A}_j(S) \neq \emptyset, \\ \dfrac{1}{|\mathcal{M}^j(S)|}, & m \in \mathcal{M}^j(S), \quad \mathcal{A}_j(S) = \emptyset, \\ 0, & \text{otherwise.} \end{cases} \tag{4.10}$$

Thus, the fitted model assigns probability only to observed classes that are feasible from S. If no observed class is feasible, it defaults to a uniform distribution over the structurally feasible marks. For any $m, r \in \mathcal{A}_j(S)$,

$$\log \frac{p_j(m \mid z, S; \beta_p^j)}{p_j(r \mid z, S; \beta_p^j)} = \eta_{j,m}(z) - \eta_{j,r}(z) = \left(\beta_{p,0}^{j,m} - \beta_{p,0}^{j,r}\right) + z^\top \left(\beta_p^{j,m} - \beta_p^{j,r}\right),$$

so coefficient differences are log-odds.

For estimation, let $\mathcal{I}_j$ index the type-j training events, and let M_k, Z_k, and S_k denote the observed mark, mark-design vector, and pre-event portfolio for event k, respectively. Since $M_k \in \mathcal{A}_j(S_k)$ by construction, the mark parameters are estimated

by the penalized choice-set likelihood

$$\widehat{\beta}_p^j = \arg\max_{\beta_p^j} \left\{ \sum_{k \in \mathcal{I}_j} \left[\eta_{j,M_k}(Z_k) - \log \sum_{r \in \mathcal{A}_j(S_k)} \exp\{\eta_{j,r}(Z_k)\} \right] - \frac{\rho}{2} \sum_{m \in \mathcal{C}_j} \|\beta_p^{j,m}\|_2^2 \right\}, \tag{4.11}$$

$j \in \mathcal{H}, \rho > 0$, where the intercepts are not penalized. Because a common shift in all class scores leaves the probabilities unchanged, we impose $\sum_{m \in \mathcal{C}_j} \beta_{p,0}^{j,m} = 0$ and $\sum_{m \in \mathcal{C}_j} \beta_p^{j,m} = 0$ for identification.

Let $\beta_p = (\beta_p^{\mathrm{b}}, \beta_p^{\mathrm{d}})$. The marked model has individual joint log density

$$\ell_i^{\mathrm{marked}}(\theta, \beta_p) = \ell_i^{\mathrm{time}}(\theta) + \sum_{j \in \mathcal{H}} \sum_{\ell : T_{i\ell}^j \le \tau_i} \log p_j \Big(M_{i\ell}^j \mid Z_i^j(T_{i\ell}^j-), S_i(T_{i\ell}^j-); \beta_p^j \Big). \tag{4.12}$$

This decomposition specifies the joint model but is not maximized as a single fitting criterion. The two terms involve disjoint parameter blocks: θ is estimated by maximizing $\sum_i \ell_i^{\mathrm{time}}(\theta)$, while each β_p^j is estimated by the penalized conditional criterion in (4.11).

4.3.3 Comparing Related Models

Fixed-horizon churn models define a binary label such as

$$Y_i^{(h)}(t) = \mathbf{1}\{S_i(t+h) = \emptyset\},$$

and model the probability of churn over a pre-specified horizon. Such models can be useful for targeting, but they do not directly model the timing or ordering of intermediate buy and drop events. Survival models, such as those of Brockett et al. (2008), Guillén et al. (2009) and Guillén et al. (2012) instead model a time-to-event object, often the residual time from a first cancellation D_i to full churn,

$$U_i = T_i^0 - D_i.$$

A time-varying Cox model can then write the post-cancellation hazard as

$$\lambda_i(u) = Y_i(u)\lambda_0(u) \exp\{X_i(D_i + u)^\top \beta(u)\}, \qquad u > 0.$$

This captures the idea that the period immediately after a cancellation may be different from later periods. Our proposed Hawkes formulation represents a similar idea through event-triggered impulses: each previous drop contributes a decaying term to the future drop intensity,

$$\sum_{\ell : T_{i\ell}^{\mathrm{d}} < t} \eta^{\mathrm{dd}} \nu^{\mathrm{d}}(t - T_{i\ell}^{\mathrm{d}}),$$

so the dependence is updated after every observed drop, not only after the first one.

Discrete-time multi-state churn models, such as Dong et al. (2022), model transitions between coverage states on an observation grid, for example through

$$\Pr(s(S_{i,m+1}) = s' \mid s(S_{i,m}), s(S_{i,m-1}), X_{i,m}).$$

These models are well suited to annual renewal data and explicitly represent movements between portfolio states. Compared to our model, only the previous state enters the churn intensity and not the full history. Also, to keep the dimension low, only $s(S_i, m)$ (e.g. $s(S_i, m) =$ partial,full, churned) is modelled.

The closest Hawkes-based insurance work is Lesage et al. (2024), who also use a multivariate mutually exciting Hawkes process. In Lesage et al. (2024) the Hawkes components describe subscription and life-event processes for insurance recommendation, whereas here the components are the lower-dimensional buy and drop processes used to study churn and portfolio evolution.

4.4 Model selection

In the preceding section, we specified a fully parametric class of models for the buying and dropping intensities, λ^{b} and λ^{d}. Given a data set, the goal is to select the best model within this class. Specifically, we seek to determine which parameters and covariates should be included so that the resulting model provides an accurate description of the data and the underlying data-generating process.

Because we explicitly compute the event-time likelihood in (4.5), standard likelihood-based methods for selecting among the event-time intensity models, such as AIC, BIC, and likelihood-ratio tests, can be applied. These methods are designed to choose the model with the best goodness-of-fit while accounting for overfitting under the assumption that the data-generating process follows the specified model. They do not apply directly to the L2-penalized choice-set mark fit in (4.11).

Alternatively, instead of relying on a correct parametric specification, one may choose a model based on its predictive performance on non-parametric metrics. One way to assess predictive performance is to evaluate how well the model predicts future events. In practice, this is done by splitting the data into a training set on $[0, t']$ and a test set on $(t', t]$. The model is fitted on the training set and then used to generate predictions on the test set, which can be compared with the observed data on some performance metric.

As described in Section 4.6.1, we will later introduce an algorithm for simulating events from the fitted model parameters. Hence, we can predict the path in the future $(t', t]$ by simulation. We can evaluate the model by repeatedly simulating events from the fitted model and averaging for comparison on the observed data in $(t', t]$ under the chosen performance metric.

Taking advantage of our simulator that simulates entire paths for any inputs following the specified model, we can perform model evaluation of a range of

performance metrics on which we wish to evaluate and compare the models, choosing e.g. metrics such as time to next event, time to full churn, or simulating entire customer lifetimes. Later, in the simulation study, we consider the probability of full churn in $(t', t]$. Here we compute, for each fitted model, the mean squared error (MSE) between the full-churn probability estimated under the true data-generating process and the probability predicted by the model over B iterations, averaged across n policyholders, i.e.,

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{B}\sum_{r=1}^{B}\hat{\delta}_i^r(t', t] - \frac{1}{B}\sum_{r=1}^{B}\delta_i^r(t', t]\right)^2. \tag{4.13}$$

Here, $\hat{\delta}_i^r(t', t]$ is the indicator that policyholder i fully churns in $(t', t]$ under the fitted model in simulation replicate r, and similarly δ indicates full churn for the true model. The replicate index r is distinct from the roman event-type label b for buy. When the true data-generating process is unknown, its repeated-simulation churn probability is unavailable. We instead compare the predicted probability with the single observed binary outcome on the test set, in which case the squared loss is the binary Brier score.

Optimizing over the loss in (4.13) ensures that the model manages the difficult task of predicting well on a policyholder level. Insurance providers, however, also have an interest in making sure that their models perform well on the portfolio level,

$$\left(\frac{1}{nB}\sum_{i=1}^{n}\sum_{r=1}^{B}\hat{\delta}_i^r(t', t] - \frac{1}{nB}\sum_{i=1}^{n}\sum_{r=1}^{B}\delta_i^r(t', t]\right)^2. \tag{4.14}$$

Here we can obtain stable results even with $B = 1$ as long as n is large enough since we do not estimate individual full-churn probabilities. Although important for controlling the overall level of the model predictions, this measure is sensitive to changes in the portfolio mix because it does not differentiate among the characteristics of individual policyholders.

Simulating future events also requires a rule for covariates beyond the last observed annual record used to initialize the forecast (the forecast origin). In the application below, the transition from record y to record $y+1$ uses the covariates $A_{i,y}$ throughout; no covariates from record $y + 1$ enter the forecast. Portfolio holdings and the resulting state are different: they are predictable functions of the simulated path and are updated immediately after each simulated event. The forecasts are therefore conditional on this carry-forward assumption, rather than on future annual covariates being known.

4.5 Data application

4.5.1 Data, covariates, and evaluation design

We use the annual commercial-insurance panel from the Wisconsin Local Government Property Insurance Fund (LGPIF) studied by Dong et al. (2022). Its policyholders are local-government units rather than individuals, so personal variables such as age and gender do not apply. The fund offers six coverage lines: compulsory building and contents (BC); optional contractor's equipment or inland marine (IM); and comprehensive (PN, PO) and collision (CN, CO) cover for new and old vehicles. To reduce sparse transition cells, Dong et al. (2022) combine the four vehicle lines into Car and classify each annual portfolio as full (BC, IM, and Car), partial (active but not full), or churn; churn is absorbing. Car means that at least one vehicle line is active, so their full state need not contain all six lines. Thus, Dong et al. use one three-valued state variable, whereas our marked model retains the six individual coverage indicators.

The data contain the static and time-varying variables summarized in Table 4.1. Each row in the data is an annual policy record indexed by its year label y. Let $S_{i,y}$ denote the six-line coverage set and $A_{i,y}$ the time-varying covariates in that record; let E_i denote the static variables and $W_{i,y} = (E_i, A_{i,y})^T$. We write $s_{i,y}$ for the full/partial/churn state implied by $S_{i,y}$. Between annual observations, $C_i(t-)$ and $s_i(t-)$ denote the six coverage indicators and their implied state immediately before time t.

The data contain six-line portfolios through 2012 and a supplied full/partial/churn destination for 2013, but no six-line 2013 portfolio. In each comparison, the last observed annual record initializes the one-year forecast and the preceding records supply its historical fitting information. The primary models are fitted using the reconstructed history available through 2012; $S_{i,2012}$ and $W_{i,2012}$ then initialize prediction of the supplied 2013 destination for 1,039 eligible policies, of which 23 churn. Product-level forecasts are assessed only in an auxiliary 2011→2012 comparison where the models use history through 2011 for prediction of the observed 2012 portfolio. All remaining results and the policy illustration use the primary 2012→2013 comparison. Each marked-intensity forecast uses 100,000 simulated paths per policy. The data used in this analysis are confidential; a reduced data set without premium information is publicly available.

Table 4.1 lists the variables used in this analysis, rather than every available variable, and follows the selection of Dong et al. (2022). The four entity variables are binary encoded. Their exact collinearity makes the pooled Hawkes candidate design rank deficient, so it only retains `EntitySTV`, `TypeCity`, and `EntityCM`, but removes `EntityCSTV`. In the Dong et al. benchmarks, `EntitySTV` and `EntityCSTV` occur in different conditioning-state blocks (Table 4.2) and are therefore not redundant columns in those regression matrices. Rates are used alongside `TotalCoverage`

Notation	Contents and units	Temporal treatment
E_i	Four overlapping 0/1 indicators derived from entity type: `EntitySTV` (school, town, or village), `EntityCM` (city or miscellaneous), `EntityCSTV` (city, school, town, or village), and `TypeCity` (city).	Time-invariant.
$A_{i,y}$	`RateBC`, `RateIM`, and `RateCar`: premium in thousands of USD per USD 1 million of coverage; `RatioPremium`: total premium reported in record y divided by that reported in record $y-1$, minus one; `TotalCoverage`: insured value in hundreds of millions of USD; `IClaimBC`, `IClaimIM`, and `IClaimCar`: claim-occurrence indicators reported in record y; `FreqBC`: BC claim frequency; `FinaCris`: indicator for 2008.	Used as fixed annual input over the reconstructed transition from record y to record $y+1$.
$W_{i,y}$	The 14-variable pool $(E_i, A_{i,y})$, equal to the union used by the selected Dong et al. regression formulas.	Numerical features are standardized.
$C_i(t-)$	Indicators for the six exact coverages BC, IM, PN, PO, CN, and CO held immediately before t.	Updated after every event.
$s_i(t-)$	Current full/partial state implied by $C_i(t-)$.	Updated after every event.

Table 4.1: Variables used in the fitted LGPIF models. The index y denotes an annual record, while $t-$ denotes the instant immediately before an event.

to separate price from insured amount. Dong et al. (2022) also screened IM and Car claim frequencies, GDP, and the federal funds rate. Their reported regression formulas retained only BC frequency among the three frequency variables and `FinaCris` among the macroeconomic candidates. We therefore decided to omit those as well.

For the timing and mark models, every non-binary candidate column is centred and divided by its training-risk-set standard deviation, while binary indicators remain unscaled.

The annual records identify the two endpoint portfolios but not when changes occurred between them. For each adjacent pair $S_{i,y}$ and $S_{i,y+1}$, we reconstruct one bundled buy mark $S_{i,y+1} \setminus S_{i,y}$ and one bundled drop mark $S_{i,y} \setminus S_{i,y+1}$ whenever the respective set is non-empty. Each reconstructed event is placed at a randomly selected monthly position strictly between the two records. When both occur, their positions are distinct and their feasible order is selected at random. This preserves the observed endpoints but does not identify the true timing or order between observations. Customers already active in their first observed record are

left-truncated, and we therefore use the transient component in (4.4).
Bundled marks retain changes suppressed by the three-state encoding. In the primary histories, $\{\mathrm{PN}, \mathrm{CN}\}$ represents 119 of 224 buy marks and 135 of 455 drop marks. In the auxiliary holdout, 54 of the 113 changed exact portfolios remain in the same full/partial state.

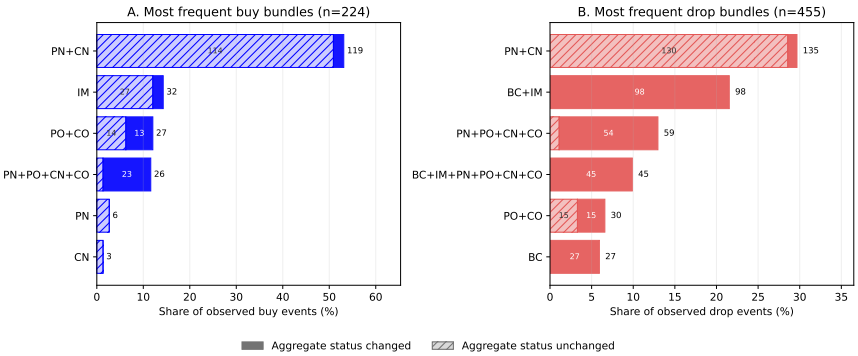


Figure 4.4: Observed coverage-set buy and drop bundles in the primary reconstructed histories through 2012. Bar lengths are shares of all 224 buy or 455 drop events, and printed values are event counts. Hatched segments leave the aggregate full/partial state unchanged; solid segments change it.

4.5.2 Models compared

We fit the two multinomial logistic regression (MLR) models employed by Dong et al. (2022) as discrete-time benchmarks; henceforth, MLR refers specifically to these two models. Their state-specific specifications are summarized in Table 4.2. For a transition from record y to record $y+1$, the Dong et al. (2022), first-order Markov MLR conditions on $s_{i,y}$, whereas the Dong et al. (2022), second-order Markov MLR conditions on the ordered pair $(s_{i,y-1}, s_{i,y})$; each first-order conditioning state and each second-order conditioning-state pair has its own intercept. The second order is empirically motivated: Dong et al. (2022) report that among policies currently full, 2.44% became partial after (full, full), compared with 11.63% after (partial, full). The latter cell contains only 43 observations, which also motivates limiting the two off-diagonal state pairs to two slopes each.

Table 4.3 gives the three timing designs and their paired restrictions. Constant and exactly collinear columns of the initial covariates are removed, leaving 18, 31, and 32 retained coordinates, respectively. The conditional mark vector $Z_i^j(t-)$ starts from the same covariates but is screened separately on observed buy or drop event rows, so its retained columns need not equal those in $X_i^j(t-)$.

Benchmark	Conditioning state(s)	Covariates with state-specific slopes
First-order	full	`EntitySTV`, `EntityCM`, `RateBC`, `RateIM`, `RateCar`, `IClaimBC`, `IClaimIM`, `IClaimCar`, `RatioPremium`, `TotalCoverage`
	partial	`EntitySTV`, `TypeCity`, `RatioPremium`, `FreqBC`, `FinaCris`
Second-order	(full, full)	`EntitySTV`, `EntityCM`, `RateBC`, `RateIM`, `RateCar`, `IClaimBC`, `IClaimIM`, `TotalCoverage`, `RatioPremium`
	(partial, full)	`IClaimCar`, `EntitySTV`
	(full, partial)	`EntitySTV`, `RateIM`
	(partial, partial)	`EntityCSTV`, `RatioPremium`, `FreqBC`, `FinaCris`

Table 4.2: Exact state-specific slopes in the two Dong et al. (2022) MLR benchmarks. Second-order pairs are $(s_{i,y-1}, s_{i,y})$, and every listed conditioning state or state pair also has its own intercept.

Timing model	Covariates	Retained dimension
$\text{Hawkes}_{C,W}$	(C, W)	18
$\text{NoExc}_{C,W}$	(C, W)	18
$\text{Hawkes}_{C,s\times W}$	$\big(C, (\mathbf{1}\{s = \text{partial}\}W, \mathbf{1}\{s = \text{full}\}W)\big)$	31
$\text{NoExc}_{C,s\times W}$	$\big(C, (\mathbf{1}\{s = \text{partial}\}W, \mathbf{1}\{s = \text{full}\}W)\big)$	31
$\text{Hawkes}_{C,s,s\times W}$	$\big(C, s, (\mathbf{1}\{s = \text{partial}\}W, \mathbf{1}\{s = \text{full}\}W)\big)$	32
$\text{NoExc}_{C,s,s\times W}$	$\big(C, s, (\mathbf{1}\{s = \text{partial}\}W, \mathbf{1}\{s = \text{full}\}W)\big)$	32

Table 4.3: Point-process timing designs and compact model names. Here C is the six-coverage indicator vector, W is the annual covariate vector, and s is the current full/partial state. Each NoExc row is the corresponding Hawkes specification with all excitation masses η^{jk} and both additive transient amplitudes fixed at zero. The reported dimension counts retained background-design coordinates after constant and collinear columns are removed.

4.5.3 Predictive performance: state and product-level outcomes

For the aggregate full/partial/churn endpoint, the marked point-process forecasts are competitive with the MLR benchmarks, but no model family dominates every score. Our models additionally describe the dynamic behaviour of the portfolios. The auxiliary holdout shows that they rank exact portfolio changes and provide useful distributions over their destinations, including changes that the three-state

outcome cannot see. Finally, the Hawkes and paired NoExc variants are generally close in this data application.

	Churn versus active				Full/partial/churn	
Model	Brier	Log loss	AUC	AP	Brier	Log loss
Dong et al. (2022), first-order Markov MLR	0.021570	0.1025	0.651	0.041	0.0617	0.1715
Dong et al. (2022), second-order Markov MLR	0.021840	0.1036	0.690	0.047	0.0581	0.1565
$\text{NoExc}_{C,W}$	0.021566	0.1028	0.608	0.040	0.0612	0.1537
$\text{Hawkes}_{C,W}$	0.021592	0.1034	0.603	0.042	0.0614	0.1517
$\text{NoExc}_{C,s\times W}$	0.021550	0.1026	0.628	0.047	0.0614	0.1532
$\text{Hawkes}_{C,s\times W}$	0.021570	0.1031	0.617	0.049	0.0611	0.1499
$\text{NoExc}_{C,s,s\times W}$	0.021540	0.1026	0.629	0.055	0.0614	0.1532
$\text{Hawkes}_{C,s,s\times W}$	0.021550	0.1030	0.616	0.057	0.0611	0.1502

Table 4.4: Out-of-sample endpoint prediction for the common 2013 holdout (1,039 policies; 23 churners). The three-state scores use the canonical full/partial/churn codebook. Lower Brier scores and log losses are better; higher AUC and average precision (AP) are better. Binary Brier scores are shown to six decimals because the low event rate compresses their differences; the constant-prevalence benchmark is 0.021647.

The primary 2012–2013 comparison asks whether a policy ends the year active or churned and, if active, whether its aggregate state is full or partial. The Dong et al. (2022), second-order Markov MLR has the best three-state Brier score (0.0581) and churn ROC AUC (0.690). The two state-interacted Hawkes designs, $\text{Hawkes}_{C,s\times W}$ and $\text{Hawkes}_{C,s,s\times W}$, instead have the lowest three-state log losses (0.1499 and 0.1502, compared with 0.1565 for the second-order MLR). For churn, $\text{NoExc}_{C,s,s\times W}$ and $\text{Hawkes}_{C,s,s\times W}$ have nearly identical Brier scores (0.021540 and 0.021550), both close to the constant-prevalence score of 0.021647. The latter Hawkes forecast has the highest average precision (AP) (0.057) and captures 5 of 23 churners (21.7%) in the top risk decile, compared with 2.3 expected under random ranking. An explanation of higher AP but lower AUC than the second-order MLR is that AUC averages rankings across all positive–negative pairs, whereas AP gives more weight to precision near the head of this rare-event ranking. At broader targeting depths, however, the second-order MLR captures 8 and 14 churners in the top 20% and 30%, compared with 6 and 9 for $\text{Hawkes}_{C,s,s\times W}$. With only 23 churners, one additional case changes sensitivity by 4.35 percentage points, so these differences are weak evidence rather than stable rankings. Exact six-product portfolios are unavailable in 2013, and the primary comparison is therefore not used to evaluate product destinations.

The auxiliary 2011–2012 holdout supplies both $S_{i,2011}$ and $S_{i,2012}$, so it can evaluate the additional product-level output. Scoring only the single most likely

Model	Top 5%	Top 10%	Top 20%	Top 30%
Dong et al. (2022), first-order Markov MLR	1 (4.3%)	2 (8.7%)	6 (26.1%)	11 (47.8%)
Dong et al. (2022), second-order Markov MLR	2 (8.7%)	2 (8.7%)	8 (34.8%)	14 (60.9%)
$\text{NoExc}_{C,W}$	2 (8.7%)	3 (13.0%)	5 (21.7%)	6 (26.1%)
$\text{Hawkes}_{C,W}$	2 (8.7%)	4 (17.4%)	5 (21.7%)	7 (30.4%)
$\text{NoExc}_{C,s\times W}$	2 (8.7%)	4 (17.4%)	7 (30.4%)	8 (34.8%)
$\text{Hawkes}_{C,s\times W}$	3 (13.0%)	5 (21.7%)	6 (26.1%)	9 (39.1%)
$\text{NoExc}_{C,s,s\times W}$	3 (13.0%)	4 (17.4%)	6 (26.1%)	9 (39.1%)
$\text{Hawkes}_{C,s,s\times W}$	3 (13.0%)	5 (21.7%)	6 (26.1%)	9 (39.1%)
Random ordering (expected)	1.2 (5.0%)	2.3 (10.0%)	4.6 (20.0%)	6.9 (30.0%)

Table 4.5: Observed churners captured at selected targeting depths in the 1,039-policy holdout (23 churners). Each fitted-model entry is the number captured, with the share of all churners in parentheses. The random-ranking row is an analytical expectation.

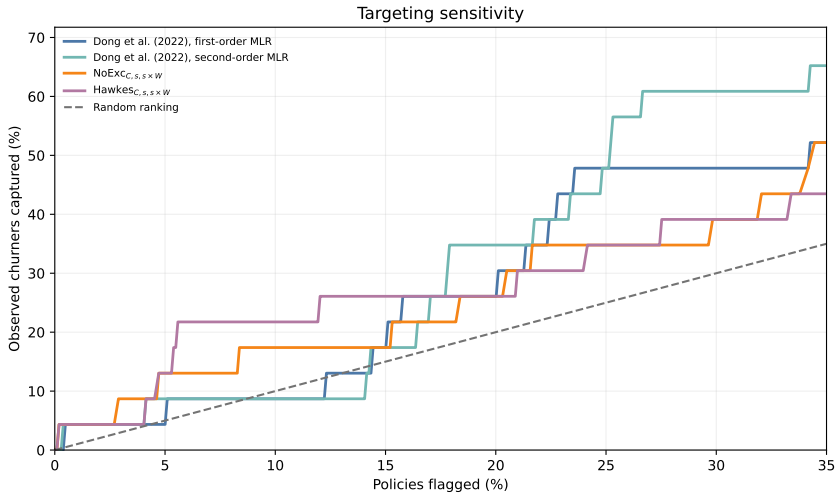


Figure 4.5: Churn captured as the targeted share of the primary 2012–2013 cohort increases. Cutoff ties receive fractional credit, and the dashed diagonal is the random-ranking expectation.

2012 portfolio would make persistence the dominant feature of the target: 962 of 1,075 policies do not change exact portfolio. We instead separate whether the 2012 portfolio differs from 2011 from what the 2012 portfolio is conditional on a change.

For policy i, define the probability of a different 2012 portfolio

$$q_i = 1 - \Pr(S_{i,2012} = S_{i,2011}).$$

Panel A: Whether the 2012 portfolio differs from 2011

	Change rate (%)				
Forecast	Observed	Predicted	Brier	AUC	AP
$\text{NoExc}_{C,s,s\times W}$	10.51	7.99	0.08289	0.774	0.353
$\text{Hawkes}_{C,s,s\times W}$	10.51	7.53	0.08257	0.775	0.361
Always no change (tied ranking)	10.51	0.00	0.10512	0.500	0.105

Panel B: Which 2012 portfolio, conditional on an observed change

Forecast	Conditional Brier	Mean probability assigned to realized destination	Conditional log loss
$\text{NoExc}_{C,s,s\times W}$	0.6224	43.6%	1.568
$\text{Hawkes}_{C,s,s\times W}$	0.6162	44.8%	1.448

Destination ranking

Forecast	Top 1	Top 3	Top 5
$\text{NoExc}_{C,s,s\times W}$	62/113 (54.9%)	94.0/113 (83.2%)	106.1/113 (93.9%)
$\text{Hawkes}_{C,s,s\times W}$	61/113 (54.0%)	96/113 (85.0%)	106.3/113 (94.1%)

Table 4.6: Auxiliary 2011–2012 product-level evaluation (1,075 policies; 113 changed exact portfolios). $\text{Hawkes}_{C,s,s\times W}$ is compared with $\text{NoExc}_{C,s,s\times W}$. Panel A evaluates whether the exact portfolio changed by reporting observed and mean predicted change rates, probability scores, and ranking scores; the always-no-change reference assigns zero change probability and has tied rankings. Panel B is restricted to the 113 observed changers and renormalizes each forecast over the 63 exact portfolios other than that policy's 2011 portfolio, including full churn. Conditional Brier averages the sum across destinations of squared differences between forecast probabilities and 0/1 indicators for the realized portfolio. Conditional log loss averages the negative natural log probability assigned to the realized portfolio; lower values are better. The mean-probability column simply averages the probability assigned to that destination; higher is better. Top k ranks destinations within each policy, not policies by risk: it gives the tie-adjusted number and share whose realized portfolio is among the k most probable alternatives. Cutoff ties receive fractional credit, so counts need not be integers. Thus the 96/113 Top 3 entry means that the realized destination is among the model's three leading alternatives for 96 changers.

Note that this is not one minus the probability of no latent event: a simulated path can contain offsetting buys and drops and still return to $S_{i,2011}$. Panel A of Table 4.6 evaluates q_i using the observed and mean predicted change rates, Brier score, ROC AUC, and AP. For $\text{Hawkes}_{C,s,s\times W}$, the AUC is 0.775 and AP is 0.361; its top risk decile contains 49 of the 113 observed changes, rather than the 11.4 expected under random ranking. $\text{NoExc}_{C,s,s\times W}$ captures 44. Their Brier scores are similarly close (0.08257 and 0.08289, respectively). The Hawkes forecast's mean predicted change probability is 7.53%, compared with the observed rate of 10.51%, so the ranking is informative but the overall change frequency is underestimated.

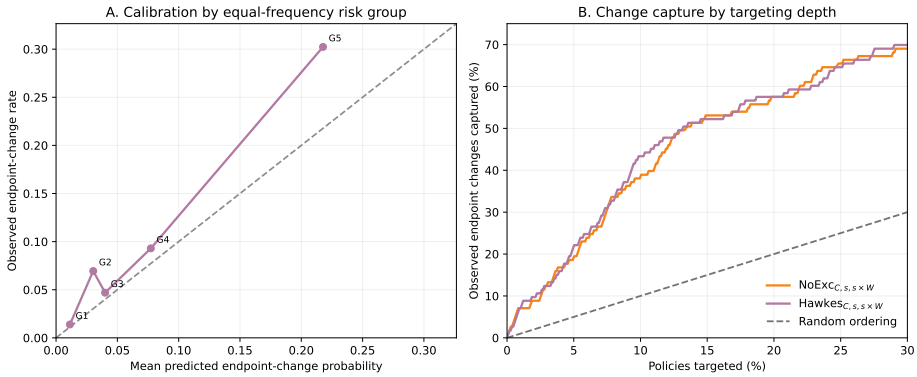


Figure 4.6: Auxiliary 2011–2012 endpoint-change diagnostics. Panel A assesses calibration for Hawkes$_{C,s,s\times W}$. The 1,075 policies are ordered by their predicted endpoint-change probability q_i and divided by quantiles into five approximately equal-frequency groups, from G_1 (lowest predicted risk) to G_5 (highest). Each point plots the group's mean predicted probability against its observed proportion of exact-portfolio changes. Panel B shows the cumulative share of observed changes captured as the targeted share increases. Cutoff ties receive fractional credit, and the diagonal denotes random ranking.

Figure 4.6 complements these scalar scores with calibration by risk group and capture across targeting depths. A retained-customer sensitivity assigns predicted change mass only to 2012 portfolios other than $S_{i,2011}$ and full churn, isolating switches among nonempty portfolios. Among the 69 such reconfigurations, Hawkes$_{C,s,s\times W}$ has AUC 0.860 and AP 0.332, captures 34 in the top risk decile, and places the realized retained destination among its first three alternatives for 60 (87.0%).

Let

$$\mathcal{I}_\Delta = \{i : S^{\text{obs}}_{i,2012} \neq S_{i,2011}\}, \qquad m = |\mathcal{I}_\Delta| = 113,$$

denote the observed exact-portfolio changers. For $i \in \mathcal{I}_\Delta$, Panel B of Table 4.6 uses the conditional destination distribution

$$r_i(s) = \Pr(S_{i,2012} = s \mid S_{i,2012} \neq S_{i,2011}), \qquad s \neq S_{i,2011}.$$

The reported conditional scores are

$$\text{Brier}_{\text{cond}} = \frac{1}{m} \sum_{i\in\mathcal{I}_\Delta} \sum_{s\neq S_{i,2011}} \left(r_i(s) - \mathbf{1}\{s = S^{\text{obs}}_{i,2012}\}\right)^2,$$

$$\text{LogLoss}_{\text{cond}} = -\frac{1}{m} \sum_{i\in\mathcal{I}_\Delta} \log r_i(S^{\text{obs}}_{i,2012}).$$

The table's mean realized-destination probability is $m^{-1}\sum_{i\in\mathcal{I}_\Delta} r_i(S^{\text{obs}}_{i,2012})$. Top k ranks the 63 alternative destination portfolios separately for each changed policy. A probability tie that straddles rank k receives fractional credit equivalent to random

ordering within the tied group, explaining non-integer counts. For example, the $\text{Hawkes}_{C,s,s\times W}$ Top 3 entry 96/113 means that the realized destination is among its three highest-probability alternatives for 96 of the 113 changers. The corresponding count is 94.0 for $\text{NoExc}_{C,s,s\times W}$; at Top 1, the counts are 61 and 62, respectively. The Hawkes forecast also assigns a slightly higher mean probability to the realized destination (44.8% versus 43.6%) and has the lower conditional Brier score (0.6162 versus 0.6224) and conditional log loss (1.448 versus 1.568). For logarithms only, the unconditional realized-destination probability is floored at 10^{-10} before the conditional log contribution is formed; this keeps the loss finite and preserves the decomposition below.

All in all this model comparison supports the value of our proposed models.

4.5.4 Interpretation of fitted values

The marked point-process forecast gives a distribution over the timing and sequence of intervening buys and drops. Note that the annual snapshots cannot validate within-year event times or ordering.

Excitation edge	Branching ratio η	Decay γ	One-year mass	Boundary status
Buy → buy	0.000	0.402	0.000	zero-excitation boundary
Drop → buy	0.277	0.402	0.092	interior
Buy → drop	2.517	0.010	0.025	γ at lower bound
Drop → drop	0.182	0.010	0.002	γ at lower bound

Table 4.7: Excitation and decay estimates for $\text{Hawkes}_{C,s,s\times W}$, in the normalized-kernel notation of (4.2). The fitted code uses jump sizes $\alpha^{jk} = \eta^{jk}\gamma^j$; the table reports $\eta^{jk} = \alpha^{jk}/\gamma^j$. The one-year mass $\eta(1 - \exp(-\gamma))$ is the unrestricted direct excitation falling in the next year before state and risk masking. Boundary status refers to the underlying constrained fit.

Table 4.7 reports the excitation parameters η^{jk} and γ^j. The branching ratio η^{jk} is the unrestricted excitation mass over all future time, whereas $\eta^{jk}(1-\exp(-\gamma^j))$ is the part falling within one year. For drop-to-buy, $\widehat{\eta}^{\text{bd}} = 0.277$ and $\widehat{\gamma}^{\text{b}} = 0.402$, giving a one-year mass of 0.092. For buy-to-drop, $\widehat{\eta}^{\text{db}} = 2.517$, but $\widehat{\gamma}^{\text{d}} = 0.010$ implies that only 0.025 falls within the next year. The drop-to-drop values are 0.182 and 0.002, respectively, while the buy-to-buy values are zero.

The estimate $\widehat{\eta}^{\text{bb}} = 0$ is at the zero-excitation boundary, while $\widehat{\gamma}^{\text{d}} = 0.010$ is at its optimization lower bound in all three Hawkes designs. That decay is shared by excitation into the drop intensity and the transient background.

The example in Table 4.8 illustrates the additional path information that an endpoint churn or product-change score discards.

Output	Forecast
Observed 2012 portfolio	BC+PN+PO+CN+CO
One-year churn probability	5.60%
Probability of no event	72.7%
Expected buy/drop event counts	0.113 / 0.231
Median time to churn, conditional on churn	0.47 years
Expected active coverage count in 2013	4.47
Most likely 2013 portfolios	BC+PN+PO+CN+CO (74.9%); BC+PO+CO (8.1%); BC+IM+PN+PO+CN+CO (5.7%)
Leading simulated buy/drop bundles	Buy: IM (47.9%); Drop: PN+CN (46.4%)

Table 4.8: Exemplary 2012–2013 path forecast under Hawkes$_{C,s,s\times W}$, fitted to the historical records available through 2012. The example is selected deterministically as the median-risk policy among the top 10% by predicted churn risk. Bundle percentages are conditional on the corresponding simulated event type.

4.6 Simulation study

The data application shows what the marked path model can report on real insurance data, but the annual snapshots do not reveal the true within-year event history or the data-generating parameters. We therefore complement that application with a controlled simulation in which the continuous-time event history and its parameters are known. The study asks three questions: whether the event-time likelihood recovers the excitation and decay parameters when the model is correctly specified; whether fitted excitation becomes small when the true excitation parameters are zero; and how accurately nested timing models recover the true one-year probability of full churn. A final, secondary comparison illustrates what happens when these continuous-time paths are aggregated to the annual states used by a discrete-time model.

In contrast to the data application, the simulated histories are entry-observed and condition on the initial purchase. Furthermore, each simulated buy or drop changes one cover, drawn from pre-specified feasible probabilities.

4.6.1 Simulation design

To isolate the event-time mechanism while retaining time-varying covariate paths, the simulation has two modular stages. First, an auxiliary insurance portfolio is generated at monthly resolution. Customer and product covariates are drawn from pre-specified distributions; product-specific Poisson frequency and gamma severity models generate claims and risk premiums; and a logistic model generates

the premium-increase indicator. Age-related variables are updated annually. The complete covariate inventories and coefficient settings are reported in Appendix 4.A. The event-time simulator is conditional on these covariate paths.

Buy and drop events are simulated from the bivariate intensities λ_i^j, $j \in \mathcal{H}$, by thinning. At the first active month, the simulator conditions on an initial buy. This event is not generated by the intensity and contributes no event term to the likelihood, but it is retained in the Hawkes history and can excite later buys and drops, as specified in Section 4.3. Event times are generated in absolute time and then shifted so that this initial purchase occurs at time zero.

Within each time step, the simulator uses an upper bound on the total intensity to propose an event time, accepts or rejects the proposal, and assigns an accepted event to the buy or drop component in proportion to their intensities. Infeasible actions are rejected: a customer cannot buy beyond the maximum number of products or drop a product that is not held. After an accepted event, the intensity and current portfolio are updated and one feasible cover is selected from the pre-specified buy or drop probabilities. The path ends either when the final held cover is dropped, producing full churn, or at the administrative endpoint. Thus, an administratively censored customer remains active at the endpoint, while an uncensored full-churn path terminates at its final drop event.

The simulator returns the event times and types, the affected cover at each event, the time-varying covariate and portfolio history, and the censoring status. These path-valued outputs support likelihood estimation on an observed interval and forward simulation over a prediction interval. The results below use the reduced timing design in Appendix 4.A.2; the largest portfolio is a reported fit, not a scalability experiment.

4.6.2 Recovery of the event-time parameters

The first question is whether the likelihood recovers the event-time parameters when the data are generated from the fitted model class. We report a fit to continuous-time event histories with monthly covariate updates at each of four portfolio sizes, $n = 10^2, 10^3, 10^4, 10^5$, over $T = 12 \cdot 10$ months. The reduced background design contains age, log income, region, the premium-increase indicator, and time; it excludes held-cover and state-transition covariates. We then maximize the event-time likelihood in (4.5).

Table 4.9 reports the parameters that govern excitation and decay. Each parenthesized value is the absolute relative error $|\widehat{\theta} - \theta|/|\theta|$. The complete table, including the background coefficients, is given in Appendix Table 4.15.

The Hawkes parameters are generally close to their generating values in the largest portfolio. Recovery is not monotone at every intermediate sample size, however, and Appendix Table 4.15 shows that several small background coefficients remain unstable even in the largest reported portfolio.

		Estimate (absolute relative error)			
Parameter	**True**	$n = 10^2$	$n = 10^3$	$n = 10^4$	$n = 10^5$
η^{bb}	0.10	0.104 (0.04)	0.106 (0.06)	0.0932 (0.07)	0.0981 (0.02)
η^{db}	0.009	$9.90 \cdot 10^{-9}$ (1.00)	0.00546 (0.39)	0.0116 (0.28)	0.00897 (0.003)
η^{bd}	0.007	0.0616 (7.80)	0.0329 (3.70)	0.00502 (0.29)	0.00606 (0.13)
η^{dd}	0.12	0.0470 (0.61)	0.123 (0.03)	0.125 (0.04)	0.123 (0.03)
γ^{b}	1	1.35 (0.35)	0.847 (0.15)	1.02 (0.02)	1.04 (0.04)
γ^{d}	1	1.01 (0.01)	0.758 (0.24)	0.882 (0.12)	0.929 (0.07)

Table 4.9: Recovery of the excitation and decay parameters in the correctly specified reduced event-time DGP at $n = 10^2, 10^3, 10^4, 10^5$. Parentheses give the absolute relative error $|\widehat{\theta} - \theta|/|\theta|$. Each column is one reported point estimate for a simulated portfolio observed for ten years; no replicate summary is reported.

4.6.3 Zero-excitation boundary case

The second question is whether the fitted model produces appreciable excitation when the true process has none. We repeat the reduced experiment with all four generating values $\eta^{jk} = 0$. The complete estimates are reported in Appendix Table 4.16. In the largest reported case, $n = 10^4$, three excitation estimates are zero and the remaining estimate is 0.0025. This pattern is consistent with the fitted excitation becoming small as the portfolio grows. Note that when all excitation coefficients are zero, the decay parameters γ^{b} and γ^{d} are unidentified and are therefore not reported.

4.6.4 Recovery of one-year full-churn probabilities

The third question is how accurately the fitted timing models recover each active policyholder's probability of full churn during the next year. We fit the reduced event-time model to nine years of continuous-time event histories with monthly covariate updates and reserve the tenth year for prediction, using simulated portfolios of $n = 1000$ and $n \approx 10^5$ policyholders. The displayed n is the simulated and fitting portfolio size; the MSE is averaged over the subset active at the forecast origin. Within this reduced design, three nested timing specifications are compared: the full Hawkes timing model; a covariate-background model with all excitation parameters fixed at zero; and a constant-background model with no excitation or background covariates.

For every policyholder active at the forecast origin, we simulate $B = 1000$ paths from each fitted model and a separate $B = 1000$ paths from the known data-generating parameters. The two sets of Monte Carlo paths estimate the fitted and oracle one-year full-churn probabilities, respectively. Table 4.10 reports their policyholder-level MSE from (4.13).

The correctly specified full Hawkes timing model has the lowest reported individual-level MSE at both portfolio sizes. The ordering of the two restricted models changes between the two sizes, and their errors are close relative to the comparison with the

Timing model	$n = 1000$	$n \approx 10^5$
Full Hawkes timing model	$1.78 \cdot 10^{-4}$	$8.28 \cdot 10^{-5}$
Covariate background, no excitation	$2.24 \cdot 10^{-4}$	$1.07 \cdot 10^{-4}$
Constant background, no excitation	$2.22 \cdot 10^{-4}$	$1.15 \cdot 10^{-4}$

Table 4.10: Policyholder-level MSE between fitted and oracle one-year full-churn probabilities under the correctly specified continuous-time DGP at simulated and fitting portfolio sizes $n = 1000$ and $n \approx 10^5$. The full Hawkes timing, covariate-background/no-excitation, and constant-background/no-excitation models are fitted to nine years of continuous-time event histories with monthly covariate updates; the tenth year is evaluated over the active-at-origin subset using $B = 1000$ fitted-model paths and a separate $B = 1000$ true-DGP paths for each eligible policyholder.

full timing model. The parameter estimates underlying this forecast comparison are retained in Appendix Table 4.17.

4.6.5 Yearly aggregation and a discrete-time benchmark

The final, secondary question is how a discrete-time benchmark behaves when the simulated continuous-time paths are observed only at annual endpoints. We aggregate the monthly paths to an analogous three-state annual representation for a Dong et al. (2022)-style benchmark. Each annual observation records the end-of-year portfolio: no held covers is churn, one or two covers is low coverage, and three or more covers is high coverage. Within this representation, the "second-order MLR with origin-specific covariates" includes covariate effects within the admissible two-state origins, while the "second-order MLR with origin indicators only" uses those origin indicators without additional covariates. The continuous-time comparator has a constant buy and drop background and no excitation.

This benchmark is deliberately favourable to the continuous-time comparator. The true paths are generated by the continuous-time mechanism in Section 4.6.1 and only subsequently reduced to annual endpoints for the MLR models. Moreover, the MLR risk set contains policies in low or high coverage at both $t-2$ and $t-1$, whereas the continuous-time comparator is fitted to the available event history on $[0, t-1]$. Table 4.11 is an illustration of model–DGP alignment under yearly aggregation, not a neutral test of whether continuous-time models generally outperform discrete-time models.

The constant-background continuous-time comparator has the lowest reported error at both sample sizes in this continuous-time DGP. At $n = 1000$, the origin-indicator MLR has lower error than the covariate-augmented MLR; at $n \approx 10^5$, the two MLR values are similar. Because the experiment changes the observation scale, risk set, and training history together, it does not isolate the effect of annual aggregation or establish a general ranking of the model classes.

Model	$n = 1000$	$n \approx 10^5$
Second-order MLR with origin-specific covariates	$1.68 \cdot 10^{-3}$	$9.16 \cdot 10^{-4}$
Second-order MLR with origin indicators only	$1.09 \cdot 10^{-3}$	$9.41 \cdot 10^{-4}$
Constant-background continuous-time model	$1.70 \cdot 10^{-4}$	$1.37 \cdot 10^{-4}$

Table 4.11: Policyholder-level MSE between estimated and oracle one-year full-churn probabilities under the continuous-time Hawkes DGP at simulated portfolio sizes $n = 1000$ and $n \approx 10^5$. The two second-order MLR models use their eligible annual-history subsets through $t-1$, while the constant-background comparator uses its eligible continuous event histories on $[0, t-1]$; each targets full churn in $(t-1, t]$. Oracle probabilities use $B = 1000$ true-DGP paths at each portfolio size.

Taken together, the experiments show descriptive recovery of the event-time dynamics in a correctly specified continuous-time setting, small fitted excitation in the largest reported zero-excitation setting, and lower oracle individual-level churn-probability error for the full Hawkes timing model when excitation is present in the DGP.

4.7 Further extensions

We shortly outline some ways to improve the modelling framework, focusing on settings with large data volumes, where noise is less of a concern.

Non-linearity. In the modelling above, the background intensity is assumed to comprise only linear effects. This assumption can be relaxed by incorporating non-linear components. For example, continuous covariates with potentially non-linear time effects can be modelled using spline functions.

In addition, interaction terms can be included in the background intensity to capture non-linear effects arising from the interplay between different covariates.

Machine learning methods. Extending this idea further, one can dispense with structural assumptions on the background intensity altogether. Instead, the background intensity may be modelled non-parametrically, for instance using machine learning methods. Such approaches are well suited to capturing complex patterns in the data, including interactions among covariates.

4.A Simulation design and supplementary results

This appendix records the auxiliary portfolio generator and the reduced event-time experiment reported in Section 4.6. The portfolio generator can supply a wider set of time-varying variables than is used in the reported event-time experiment. The latter is therefore stated separately below to avoid conflating the general simulator with the data-generating process underlying Tables 4.9–4.11.

4.A.1 Auxiliary portfolio and covariate generator

Table 4.12 lists the variables generated for the auxiliary monthly insurance portfolio. Table 4.13 lists the variables that can be passed to the event-time simulator. The reported reduced experiment uses only the timing covariates stated in Appendix 4.A.2.

Variable	Category	Type	Description
id	Identifier	Integer	Id of policy holder.
month	Identifier	Integer	Active month (from time zero).
age	General	Integer	Age of policy holder.
region	General	Character	Region (Urban/Suburban/Rural).
income	General	Integer	Income.
driving_experience	Motor	Integer	Years of driving experience.
vehicle_age	Motor	Integer	Age of car.
vehicle_value	Motor	Integer	Car value (EUR).
annual_km	Motor	Integer	Annual distance driven (km).
house_age	House/Contents	Integer	Age of the house (years).
building_sum_insured	House/Contents	Integer	Sum insured of house (EUR).
building_type	House/Contents	Character	Building type.
house_location_risk	House/Contents	Character	Location risk category (Low/Medium/High).
security_system	House/Contents	Integer	Indicator of security system (0/1).
square_meters	House/Contents	Integer	Living area of the property (m^2).
contents_sum_insured	House/Contents	Integer	Sum insured for contents (EUR).
pet_age	Pet	Integer	Age of the insured pet (years).
travel_zone	Travel	Character	Geographic travel zone (Zone 1/Zone 2).
BMI	Health	Integer	Body Mass Index of the policy holder.
pre_exist_conditions	Health	Integer	Number of pre-existing medical conditions.
health_sum_insured	Health	Integer	Sum insured for medical expenses (EUR).
family_history_risk	Health	Character	Family medical risk (Low/Medium/High).

Table 4.12: Variables generated for the auxiliary monthly insurance portfolio. Units, variable types, and insurance categories are shown in the table.

Variable	Category	Type	Description
`id`	Identifier	Integer	Id of policy holder. We simulate n policy holders.
`month`	Identifier	Integer	Active month for policy. Counting from time zero.
`age`	General	Integer	Age of policy holder.
`region`	General	Character	Region of policy holder (Urban/Suburban/Rural).
`income`	General	Integer	Income of policy holder.
`motor_freq`	Motor	Integer	Number of motor claims.
`house_freq`	House	Integer	Number of house claims.
`contents_freq`	Contents	Integer	Number of contents claims.
`pet_freq`	Pet	Integer	Number of pet claims.
`travel_freq`	Travel	Integer	Number of travel claims.
`health_freq`	Health	Integer	Number of health claims.
`prem_incr`	General	Binary	Indicator of premium increase.

Table 4.13: Variables passed from the auxiliary monthly portfolio generator to the event-time simulator. Month is counted from the first active month, and (n) denotes the number of simulated policyholders.

Portfolio-model design matrices.

$$
\begin{aligned}
X^{\text{motor}} &= \big(1,\ \text{age},\ \log(\text{income}),\ \text{region\_suburban},\ \text{region\_rural},\ \text{km\_thousands},\\
&\qquad \text{driving\_experience},\ \text{vehicle\_age},\ \text{vehicle\_value\_tenk}\big),\\
X^{\text{house}} &= \big(1,\ \text{age},\ \log(\text{income}),\ \text{region\_suburban},\ \text{region\_rural},\\
&\qquad \log(\text{building\_sum}),\ \tfrac{\text{house\_age}}{10},\ \text{security},\ \text{loc\_risk},\ \text{build\_type}\big),\\
X^{\text{contents}} &= \big(1,\ \text{age},\ \log(\text{income}),\ \text{region\_suburban},\ \text{region\_rural},\\
&\qquad \log(\text{contents\_sum})\big),\\
X^{\text{pet}} &= \big(1,\ \text{age},\ \log(\text{income}),\ \text{region\_suburban},\ \text{region\_rural},\ \text{pet\_age}\big),\\
X^{\text{travel}} &= \big(1,\ \text{age},\ \log(\text{income}),\ \text{region\_suburban},\ \text{region\_rural},\ \text{zone2}\big),\\
X^{\text{health}} &= \big(1,\ \text{age},\ \log(\text{income}),\ \text{region\_suburban},\ \text{region\_rural},\ \tfrac{\text{BMI}}{10},\\
&\qquad \log\big(1+\text{pre\_exist}\big),\ \text{history\_risk}\big),\\
X^{\text{price}} &= \big(1,\ \text{age},\ \tfrac{\text{income}}{10\,000},\ \text{tenure\_years},\ \text{region\_suburban},\ \text{region\_rural}\big).
\end{aligned}
$$

Frequency coefficients.

$$
\begin{aligned}
\beta^{\text{freq}}_{\text{motor}} &= \big(\log(0.05),\ 0.01,\ -0.10,\ 0.20,\ 0.30,\ 0.05,\ -0.02,\ 0.01,\ 0.02\big)^{\top},\\
\beta^{\text{freq}}_{\text{house}} &= \big(\log(0.05),\ 0.005,\ 0.020,\ 0.050,\ 0.050,\ 0.020,\ -0.010,\ -0.20,\ 0.10,\\
&\qquad 0.05\big)^{\top},\\
\beta^{\text{freq}}_{\text{contents}} &= \big(\log(0.05),\ 0.01,\ -0.05,\ 0.10,\ 0.20,\ 0.05\big)^{\top},\\
\beta^{\text{freq}}_{\text{pet}} &= \big(\log(0.05),\ 0.01,\ 0.10,\ 0.05,\ 0.10,\ 0.03\big)^{\top},\\
\beta^{\text{freq}}_{\text{travel}} &= \big(\log(0.05),\ 0.005,\ 0.020,\ 0.10,\ 0.20,\ 0.50\big)^{\top},\\
\beta^{\text{freq}}_{\text{health}} &= \big(\log(0.05),\ 0.01,\ 0.05,\ 0.10,\ 0.10,\ 0.10,\ 0.10,\ 0.05\big)^{\top}.
\end{aligned}
$$

Severity coefficients.

$$\begin{aligned}
\beta^{\text{sev}}_{\text{motor}} &= \big(\log(8000),\ 0.00,\ 0.02,\ 0.00,\ 0.00,\ 0.00,\ -0.002,\ 0.00,\ 0.25\big)^\top,\\
&\quad \alpha_{\text{motor}} = 2,\\
\beta^{\text{sev}}_{\text{house}} &= \big(\log(10000),\ 0.00,\ 0.01,\ 0.00,\ 0.00,\ 0.02,\ -0.005,\ -0.10,\ 0.05,\\
&\quad 0.00\big)^\top,\\
&\quad \alpha_{\text{house}} = 5,\\
\beta^{\text{sev}}_{\text{contents}} &= \big(\log(4000),\ 0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.05\big)^\top,\\
&\quad \alpha_{\text{contents}} = 3,\\
\beta^{\text{sev}}_{\text{pet}} &= \big(\log(2000),\ 0.00,\ 0.01,\ 0.00,\ 0.00,\ 0.01\big)^\top,\\
&\quad \alpha_{\text{pet}} = 2,\\
\beta^{\text{sev}}_{\text{travel}} &= \big(\log(5500),\ 0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.10\big)^\top,\\
&\quad \alpha_{\text{travel}} = 2,\\
\beta^{\text{sev}}_{\text{health}} &= \big(\log(9000),\ 0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.01,\ 0.05,\ 0.10\big)^\top,\\
&\quad \alpha_{\text{health}} = 8.
\end{aligned}$$

Premium-increase model.

$$\beta_{\text{price}} = \big(\text{logit}(0.02),\ 0.005,\ 0.02,\ 0.20,\ -0.10,\ -0.20\big)^\top.$$

4.A.2 Reduced event-time experiment

The results in the simulation section use the reduced timing design shown here, rather than all variables generated by the auxiliary portfolio model. For event type $j \in \{\text{b}, \text{d}\}$, the predictable background covariate vector is

$$\begin{aligned}
X_i^j(t-) = \big(&\text{age}_i(t-), \log(\text{income}_i(t-)), \mathbb{1}\{\text{suburb}_i(t-)\}, \mathbb{1}\{\text{rural}_i(t-)\},\\
&\text{prem_incr}_i(t-), t\big)^\top.
\end{aligned}$$

The intercepts and decay parameters are

$$(\beta^{\text{b}}_{\mu,0}, \beta^{\text{d}}_{\mu,0}) = (-4.0, -4.5), \qquad (\gamma^{\text{b}}, \gamma^{\text{d}}) = (1.0, 1.0).$$

With rows indexing the affected event type and columns indexing the past event type, the matrix of integrated excitation coefficients under the normalized kernel parametrization is

$$\eta = \begin{pmatrix} 0.10 & 0.007 \\ 0.009 & 0.12 \end{pmatrix}.$$

The background coefficients used for the buy and drop intensities are:

Covariate	Buy	Drop
Age	−0.005	−0.006
Log income	−0.001	−0.002
Suburb indicator	0.010	−0.010
Rural indicator	0.020	−0.020
Premium increase	−0.020	0.100
Time	0.001	−0.001

Table 4.14: Background coefficients in the reduced event-time data-generating process used for the reported simulation experiments.

4.A.3 Complete parameter estimates

Tables 4.15–4.17 provide the coefficient estimates moved from the main text. As in the main text, each column is one reported fit; the tables do not summarize repeated simulation runs.

		Estimate (absolute relative error)			
Parameter	**True**	$n = 10^2$	$n = 10^3$	$n = 10^4$	$n = 10^5$
η^{bb}	0.10	0.104 (0.04)	0.106 (0.06)	0.0932 (0.07)	0.0981 (0.02)
η^{db}	0.009	$9.90 \cdot 10^{-9}$ (1.00)	0.00546 (0.39)	0.0116 (0.28)	0.00897 (0.003)
η^{bd}	0.007	0.0616 (7.80)	0.0329 (3.70)	0.00502 (0.29)	0.00606 (0.13)
η^{dd}	0.12	0.0470 (0.61)	0.123 (0.03)	0.125 (0.04)	0.123 (0.03)
γ^{b}	1	1.35 (0.35)	0.847 (0.15)	1.02 (0.02)	1.04 (0.04)
γ^{d}	1	1.01 (0.01)	0.758 (0.24)	0.882 (0.12)	0.929 (0.07)
$\beta^{\mathrm{b}}_{\mu,0}$	−4	−1.30 (0.68)	−3.68 (0.08)	−4.53 (0.13)	−4.58 (0.15)
$\beta^{\mathrm{d}}_{\mu,0}$	−4.5	−2.86 (0.36)	−3.76 (0.16)	−4.99 (0.11)	−4.94 (0.10)
$\beta^{\mathrm{b}}_{\mu,\mathrm{age}}$	−0.005	0.0105 (3.10)	−0.00609 (0.22)	−0.00386 (0.23)	−0.00465 (0.07)
$\beta^{\mathrm{b}}_{\mu,\log(\mathrm{income})}$	−0.001	−0.280 (279.00)	−0.00930 (8.30)	−0.00540 (4.40)	0.0544 (55.40)
$\beta^{\mathrm{b}}_{\mu,\mathrm{suburb}}$	0.010	0.0153 (0.53)	−0.0774 (8.74)	0.0483 (3.83)	0.0282 (1.82)
$\beta^{\mathrm{b}}_{\mu,\mathrm{rural}}$	0.020	−0.137 (7.85)	−0.211 (11.55)	0.0313 (0.56)	0.0186 (0.07)
$\beta^{\mathrm{b}}_{\mu,\mathrm{prem}}$	−0.020	−0.167 (7.35)	−0.249 (11.45)	−0.00684 (0.66)	−0.0359 (0.79)
$\beta^{\mathrm{b}}_{\mu,t}$	0.001	−0.0177 (18.70)	−0.000838 (1.84)	−0.0496 (50.60)	0.000306 (0.69)
$\beta^{\mathrm{d}}_{\mu,\mathrm{age}}$	−0.006	−0.00542 (0.10)	−0.00284 (0.53)	−0.0236 (2.93)	−0.00527 (0.12)
$\beta^{\mathrm{d}}_{\mu,\log(\mathrm{income})}$	−0.002	−0.121 (59.50)	−0.0840 (41.00)	0.0158 (8.90)	0.0287 (15.35)
$\beta^{\mathrm{d}}_{\mu,\mathrm{suburb}}$	−0.010	−0.106 (9.60)	−0.0328 (2.28)	−0.0108 (0.08)	−0.0000561 (0.99)
$\beta^{\mathrm{d}}_{\mu,\mathrm{rural}}$	−0.020	0.380 (20.00)	−0.240 (11.00)	0.103 (6.15)	−0.00923 (0.54)
$\beta^{\mathrm{d}}_{\mu,\mathrm{prem}}$	0.100	0.0854 (0.15)	0.411 (3.11)	−0.000185 (1.00)	0.105 (0.05)
$\beta^{\mathrm{d}}_{\mu,t}$	−0.001	−0.0104 (9.40)	−0.000861 (0.14)	0.000023 (1.02)	−0.000956 (0.04)

Table 4.15: Complete coefficient recovery in the correctly specified reduced event-time DGP for the reported portfolio at each sample size, observed for ten years. Parentheses give $|\widehat{\theta} - \theta|/|\theta|$; the columns are point estimates, not averages over replicated datasets.

		Estimate (absolute relative error)		
Parameter	**True**	$n = 10^2$	$n = 10^3$	$n = 10^4$
η^{bb}	0	0.0127 (–)	0 (–)	0 (–)
η^{db}	0	0.0350 (–)	0 (–)	0 (–)
η^{bd}	0	0.0632 (–)	0.0284 (–)	0 (–)
η^{dd}	0	0 (–)	0.0175 (–)	0.0025 (–)
γ^{b}	–	–	–	–
γ^{d}	–	–	–	–
$\beta^{\mathrm{b}}_{\mu,0}$	−4	−0.5705 (0.86)	−2.7119 (0.32)	−4.1009 (0.03)
$\beta^{\mathrm{d}}_{\mu,0}$	−4.5	−4.1955 (0.07)	−6.5528 (0.46)	−4.9465 (0.10)
$\beta^{\mathrm{b}}_{\mu,\mathrm{age}}$	−0.005	−0.0027 (0.46)	−0.0038 (0.24)	−0.0056 (0.12)
$\beta^{\mathrm{b}}_{\mu,\log\mathrm{inc}}$	−0.001	−0.3090 (308.00)	−0.1186 (117.60)	0.0152 (16.20)
$\beta^{\mathrm{b}}_{\mu,\mathrm{suburb}}$	0.010	0.4904 (48.04)	0.0595 (4.95)	0.0199 (0.99)
$\beta^{\mathrm{b}}_{\mu,\mathrm{rural}}$	0.020	0.2480 (11.40)	0.1420 (6.10)	−0.0007 (1.03)
$\beta^{\mathrm{b}}_{\mu,\mathrm{prem}}$	−0.020	−0.8717 (42.59)	−0.2584 (11.92)	−0.0373 (0.87)
$\beta^{\mathrm{b}}_{\mu,t}$	0.001	−0.0114 (12.40)	0.0003 (0.70)	0.0008 (0.20)
$\beta^{\mathrm{d}}_{\mu,\mathrm{age}}$	−0.006	−0.0039 (0.35)	−0.0059 (0.02)	−0.0064 (0.07)
$\beta^{\mathrm{d}}_{\mu,\log\mathrm{inc}}$	−0.002	−0.0213 (9.65)	0.1885 (95.25)	0.0390 (20.50)
$\beta^{\mathrm{d}}_{\mu,\mathrm{suburb}}$	−0.010	−0.1311 (12.11)	0.0681 (7.81)	−0.0062 (0.38)
$\beta^{\mathrm{d}}_{\mu,\mathrm{rural}}$	−0.020	0.2482 (13.41)	−0.3682 (17.41)	−0.0703 (2.51)
$\beta^{\mathrm{d}}_{\mu,\mathrm{prem}}$	0.100	−1.0468 (11.47)	−0.0917 (1.92)	0.0824 (0.18)
$\beta^{\mathrm{d}}_{\mu,t}$	−0.001	−0.0051 (4.10)	−0.0013 (0.30)	−0.0016 (0.60)

Table 4.16: Complete coefficient estimates in the zero-excitation boundary DGP for the reported portfolio at each sample size, observed for ten years. Parenthesized errors are undefined for the zero excitation coefficients. The decay parameters are not reported because they are unidentified when every excitation coefficient is zero.

Parameter	True	Estimated parameters					
		Full Hawkes timing model		Covariate background, no excitation		Constant background, no excitation	
		$n = 10^3$	$n \approx 10^5$	$n = 10^3$	$n \approx 10^5$	$n = 10^3$	$n \approx 10^5$
η^{bb}	0.10	0.0986	0.0993	–	–	–	–
η^{db}	0.009	0.00707	0.00902	–	–	–	–
η^{bd}	0.007	0.0397	0.00582	–	–	–	–
η^{dd}	0.12	0.0856	0.121	–	–	–	–
γ^{b}	1.00	0.849	0.997	–	–	–	–
γ^{d}	1.00	0.658	0.921	–	–	–	–
$\beta^{\mathrm{b}}_{\mu,0}$	−4.00	−4.55	−4.63	−4.09	−3.63	−3.96	−3.97
$\beta^{\mathrm{d}}_{\mu,0}$	−4.50	−5.18	−4.88	−5.20	−4.74	−4.73	−4.83
$\beta^{\mathrm{b}}_{\mu,\mathrm{age}}$	−0.005	−0.00381	−0.00467	−0.00303	−0.00432	–	–
$\beta^{\mathrm{b}}_{\mu,\log\mathrm{inc}}$	−0.001	−0.00617	−0.00584	−0.00547	−0.00581	–	–
$\beta^{\mathrm{b}}_{\mu,\mathrm{suburb}}$	0.010	0.0591	0.0587	0.0536	0.0098	–	–
$\beta^{\mathrm{b}}_{\mu,\mathrm{rural}}$	0.020	0.0763	0.0263	0.0803	0.0239	–	–
$\beta^{\mathrm{b}}_{\mu,\mathrm{prem}}$	−0.020	−0.00120	0.01345	0.0253	0.0115	–	–
$\beta^{\mathrm{b}}_{\mu,t}$	0.001	−0.271	−0.007	−0.251	−0.019	–	–
$\beta^{\mathrm{d}}_{\mu,\mathrm{age}}$	−0.006	0.118	0.028	0.0424	0.0231	–	–
$\beta^{\mathrm{d}}_{\mu,\log\mathrm{inc}}$	−0.002	−0.199	−0.023	−0.153	−0.027	–	–
$\beta^{\mathrm{d}}_{\mu,\mathrm{suburb}}$	−0.010	−0.123	−0.024	−0.0326	−0.0191	–	–
$\beta^{\mathrm{d}}_{\mu,\mathrm{rural}}$	−0.020	0.00268	0.11362	−0.0160	0.1097	–	–
$\beta^{\mathrm{d}}_{\mu,\mathrm{prem}}$	0.100	−0.00304	0.00036	−0.0104	−0.0070	–	–
$\beta^{\mathrm{d}}_{\mu,t}$	−0.001	0.0000405	−0.000752	−0.000233	−0.00125	–	–

Table 4.17: True and fitted parameters for the three nested timing models in the correctly specified reduced event-time DGP. Each model is fitted to nine years of continuous-time event histories with monthly covariate updates at the indicated portfolio size; these fits underlie the one-year oracle probability comparison based on $B = 1000$ fitted-model paths and a separate $B = 1000$ true-DGP paths per eligible policyholder.

Bibliography

Adadi, A. and M. Berrada (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access 6*, 52138–52160.

Agarwal, A., M. Dudík, and Z. S. Wu (2019). Fair regression: Quantitative definitions and reduction-based algorithms. In *Proceedings of the 36th International Conference on Machine Learning*, Volume 97 of *Proceedings of Machine Learning Research*, pp. 120–129. PMLR.

Ancona, M., C. Oztireli, and M. Gross (2019). Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Value Approximation. In *Proceedings of the 36th International Conference on Machine Learning*, Volume 97 of *Proceedings of Machine Learning Research*, pp. 272–281. PMLR.

Apley, D. W. and J. Zhu (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.

Becker, B. and R. Kohavi (1996). Adult. UCI Machine Learning Repository.

Bischl, B., G. Casalicchio, M. Feurer, P. Gijsbers, F. Hutter, M. Lang, R. G. Mantovani, J. N. van Rijn, and J. Vanschoren (2021). Openml benchmarking suites.

Bordt, S. and U. von Luxburg (2023). From shapley values to generalized additive models and back. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, Volume 206 of *Proceedings of Machine Learning Research*, pp. 709–745. PMLR.

Brockett, P. L., L. L. Golden, M. Guillen, J. P. Nielsen, J. Parner, and A. M. Perez-Marin (2008). Survival analysis of a household portfolio of insurance policies: How much time do you have to stop total customer defection? *Journal of Risk and Insurance 75*(3), 713–737.

Chastaing, G., F. Gamboa, and C. Prieur (2012). Generalized Hoeffding-Sobol decomposition for dependent variables - application to sensitivity analysis. *Electronic Journal of Statistics 6*, 2420 – 2448.

Chen, H., J. D. Janizek, S. Lundberg, and S.-I. Lee (2020). True to the model or true to the data?

Chen, T. and C. Guestrin (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794. ACM.

Chen, Y., R. Raab, J. Wang, and Y. Liu (2022). Fairness transferability subject to bounded distribution shift. In *Advances in Neural Information Processing Systems*, Volume 35, pp. 11266–11278.

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

Ding, F., M. Hardt, J. Miller, and L. Schmidt (2021). Retiring Adult: New datasets for fair machine learning. In *Advances in Neural Information Processing Systems*, Volume 34, pp. 6478–6490.

Dong, Y., E. W. Frees, F. Huang, and F. K. C. Hui (2022). Multi-State Modelling of Customer Churn. *ASTIN Bulletin 52*(3), 735–764.

Duchi, J. C. and H. Namkoong (2021). Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics 49*(3), 1378–1406.

EIOPA Consultative Expert Group on Digital Ethics in Insurance (2021). Artificial intelligence governance principles: Towards ethical and trustworthy artificial intelligence in the European insurance sector. Technical report, European Insurance and Occupational Pensions Authority, Luxembourg.

European Commission (2012). Guidelines on the application of Council Directive 2004/113/EC to insurance, in the light of the judgment of the Court of Justice of the European Union in case C-236/09 (Test-Achats). Official Journal of the European Union, 2012/C 11/01, pp. 1–11.

Fischer, S. F., L. H. M. Feurer, and B. Bischl (2023). OpenML-CTR23 – a curated tabular regression benchmarking suite. In *AutoML Conference 2023 (Workshop)*.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics 29*(5), 1189–1232.

Fumagalli, F., M. Muschalik, E. Hüllermeier, B. Hammer, and J. Herbinger (2025). Unifying feature-based explanations with functional ANOVA and cooperative game theory. In *The 28th International Conference on Artificial Intelligence and Statistics*.

Guillén, M., A. M. Perez-Marin, and J. Perch Nielsen (2009). Cross-buying behaviour and customer loyalty in the insurance sector. *EsicMarket*. Available at SSRN: `https://ssrn.com/abstract=3421534`.

Guillén, M., J. P. Nielsen, T. H. Scheike, and A. M. Pérez-Marín (2012). Time-varying effects in the analysis of customer loyalty: A case study in insurance. *Expert Systems with Applications 39*(3), 3551–3558.

Guo, R., W. Xiong, Y. Zhang, and Y. Hu (2024). Enhancing game customer churn prediction with a stacked ensemble learning model. *The Journal of Supercomputing 81*(1), 178.

Hardt, M., E. Price, and N. Srebro (2016). Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, Volume 29, pp. 3315–3323.

Harsanyi, J. C. (1963). A Simplified Bargaining Model for the n-Person Cooperative Game. *International Economic Review 4*(2), 194–220.

Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Volume 2. Springer.

Hawkes, A. G. and D. Oakes (1974). A cluster process representation of a self-exciting process. *Journal of applied probability 11*(3), 493–503.

Henzi, A., X. Shen, M. Law, and P. Bühlmann (2025). Invariant probabilistic prediction. *Biometrika 112*(1), asae063.

Herren, A. and P. R. Hahn (2022). Statistical aspects of shap: Functional anova for model interpretation.

Hiabu, M., J. T. Meyer, and M. N. Wright (2023). Unifying local and global model explanations by functional decomposition of low dimensional structures. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, Volume 206 of *Proceedings of Machine Learning Research*, pp. 7040–7060. PMLR.

Hooker, G. (2007). Generalized Functional ANOVA Diagnostics for High-Dimensional Functions of Dependent Variables. *Journal of Computational and Graphical Statistics 16*(3), 709–732.

Janzing, D., L. Minorics, and P. Bloebaum (2020). Feature relevance quantification in explainable AI: A causal problem. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, Volume 108 of *Proceedings of Machine Learning Research*, pp. 2907–2916. PMLR.

Jethani, N., M. Sudarshan, I. C. Covert, S.-I. Lee, and R. Ranganath (2022). FastSHAP: Real-time shapley value estimation. In *International Conference on Learning Representations*.

Jung, J., K. Lee, and M. Xu (2025). Modeling multistate health transitions with a most-recent-event hawkes process. *North American Actuarial Journal 0*(0), 1–22.

Kallenberg, O. (2021). *Foundations of Modern Probability* (3rd ed.). Probability Theory and Stochastic Modelling. Cham: Springer.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.

Kennedy, E. H., Z. Ma, M. D. McHugh, and D. S. Small (2016). Non-parametric methods for doubly robust estimation of continuous treatment effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology 79*(4), 1229–1245.

Khodadadi, A., S. A. Hosseini, E. Pajouheshgar, F. Mansouri, and H. R. Rabiee (2022, April). Choracle: A unified statistical framework for churn prediction. *IEEE Transactions on Knowledge and Data Engineering 34*(4), 1656–1666.

Kilbertus, N., M. Rojas Carulla, G. Parascandolo, M. Hardt, D. Janzing, and B. Schölkopf (2017). Avoiding discrimination through causal reasoning. In *Advances in Neural Information Processing Systems*, Volume 30, pp. 656–666.

Kuhn, D., S. Shafiee, and W. Wiesemann (2025). Distributionally robust optimization. *Acta Numerica 34*, 579–804.

Kusner, M. J., J. R. Loftus, C. Russell, and R. Silva (2017). Counterfactual fairness. In *Advances in Neural Information Processing Systems*, Volume 30, pp. 4066–4076.

Lengerich, B., S. Tan, C.-H. Chang, G. Hooker, and R. Caruana (2020). Purifying Interaction Effects with the Functional ANOVA: An Efficient Algorithm for Recovering Identifiable Additive Models. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Volume 108 of *Proceedings of Machine Learning Research*, pp. 2402–2412. PMLR.

Lesage, L., M. Deaconu, A. Lejay, J. A. Meira, G. Nichil, and R. State (2024, October). A Recommendation System For Insurance Built With A Multivariate Hawkes Process Based On Customers' Life Events. preprint.

Lindholm, M., R. Richman, A. Tsanakas, and M. V. Wüthrich (2022). Discrimination-free insurance pricing. *ASTIN Bulletin: The Journal of the IAA 52*(1), 55–89.

Lindholm, M., R. Richman, A. Tsanakas, and M. V. Wüthrich (2024). What is fair? proxy discrimination vs. demographic disparities in insurance pricing. *Scandinavian Actuarial Journal 2024*(9), 935–970.

Liu, J., T. Steensgaard, M. N. Wright, N. Pfister, and M. Hiabu (2025). Fast estimation of partial dependence functions using trees. In *Proceedings of the 42nd International Conference on Machine Learning*, Volume 267 of *Proceedings of Machine Learning Research*, pp. 39496–39534. PMLR.

Lundberg, S. M., G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee (2020). From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence 2*(1), 56–67.

Maity, S., D. Mukherjee, M. Yurochkin, and Y. Sun (2021). Does enforcing fairness mitigate biases caused by subpopulation shift? In *Advances in Neural Information Processing Systems*, Volume 34, pp. 25773–25784.

Mitchell, S., E. Potash, S. Barocas, A. D'Amour, and K. Lum (2021). Algorithmic fairness: Choices, assumptions, and definitions. *Annual Review of Statistics and Its Application 8*, 141–163.

Molnar, C., T. Freiesleben, G. König, J. Herbinger, T. Reisinger, G. Casalicchio, M. N. Wright, and B. Bischl (2023). Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process. In *World Conference on Explainable Artificial Intelligence*, pp. 456–479. Springer.

Muschalik, M., F. Fumagalli, B. Hammer, and E. Hüllermeier (2024). Beyond TreeSHAP: Efficient Computation of Any-Order Shapley Interactions for Tree Ensembles. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (Technical Track 13)*, pp. 14388–14396.

Nabi, R., N. S. Hejazi, M. J. van der Laan, and D. Benkeser (2024). Statistical learning for constrained functional parameters in infinite-dimensional models. arXiv preprint arXiv:2404.09847.

Nabi, R. and I. Shpitser (2018). Fair inference on outcomes. *Proceedings of the AAAI Conference on Artificial Intelligence 32*(1), 1931–1940.

Pearl, J. (2009). *Causality: Models, Reasoning, and Inference* (2nd ed.). Cambridge University Press.

Peters, J., P. Bühlmann, and N. Meinshausen (2016). Causal inference by using invariant prediction: Identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology 78*(5), 947–1012.

Pope, D. G. and J. R. Sydnor (2011). Implementing anti-discrimination policies in statistical profiling models. *American Economic Journal: Economic Policy 3*(3), 206–231.

Rezaei, A., A. Liu, O. Memarrast, and B. D. Ziebart (2021). Robust fairness under covariate shift. *Proceedings of the AAAI Conference on Artificial Intelligence 35*(11), 9419–9427.

Rojas-Carulla, M., B. Schölkopf, R. E. Turner, and J. Peters (2018). Invariant models for causal transfer learning. *Journal of Machine Learning Research 19*(36), 1–34.

Rota, G.-C. (1964). On the foundations of combinatorial theory I. Theory of Möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiet*, 340–368.

Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association 100*(469), 322–331.

Sagawa, S., P. W. Koh, T. B. Hashimoto, and P. Liang (2020). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*.

Stone, C. J. (1994). The Use of Polynomial Splines and Their Tensor Products in Multivariate Function Estimation. *The Annals of Statistics 22*(1), 118 – 171.

Štrumbelj, E. and I. Kononenko (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research 11*(1), 1–18.

Taufiq, M. F., P. Blöbaum, and L. Minorics (2023). Manifold Restricted Interventional Shapley Values. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, Volume 206 of *Proceedings of Machine Learning Research*, pp. 5079–5106. PMLR.

Thams, N., S. Saengkyongam, N. Pfister, and J. Peters (2023). Statistical testing under distributional shifts. *Journal of the Royal Statistical Society Series B: Statistical Methodology 85*(3), 597–663.

Wang, G., Y.-N. Chuang, M. Du, F. Yang, Q. Zhou, P. Tripathi, X. Cai, and X. Hu (2022). Accelerating shapley explanation via contributive cooperator selection. In *Proceedings of the 39th International Conference on Machine Learning*, Volume 162 of *Proceedings of Machine Learning Research*, pp. 22576–22590. PMLR.

Wettenstein, R., A. Nadel, and U. Boker (2026). Woodelf++: A fast and unified partial dependence plot algorithm for decision tree ensembles.

Wu, Y., L. Zhang, X. Wu, and H. Tong (2019). PC-Fairness: A unified framework for measuring causality-based fairness. In *Advances in Neural Information Processing Systems*, Volume 32, pp. 3399–3409.

Yang, J. (2022). Fast treeshap: Accelerating shap value computation for trees.

Yin, J., Y. K. Feng, and Y. Liu (2025). Modeling behavioral dynamics in digital content consumption: An attention-based neural point process approach with applications in video games. *Marketing Science 44*(1), 220–239.

Yu, P., A. Bifet, J. Read, and C. Xu (2022). Linear tree shap. In *Advances in Neural Information Processing Systems*, Volume 35, pp. 25818–25828. Curran Associates, Inc.

Zern, A., K. Broelemann, and G. Kasneci (2023). Interventional SHAP Values and Interaction Values for Piecewise Linear Regression Trees. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (Technical Track 9)*, pp. 11164–11173.