Bayesian Learning in the Counterfactual World

Alberto Caron

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Statistical Science
University College London

August 29, 2023

Declaration of Authorship

I, Alberto Caron, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Recent years have witnessed a surging interest towards the use of machine learning tools for causal inference. In contrast to the usual large data settings where the primary goal is prediction, many disciplines, such as health, economic and social sciences, are instead interested in causal questions. Learning individualized responses to an intervention is a crucial task in many applied fields (e.g., precision medicine, targeted advertising, precision agriculture, etc.) where the ultimate goal is to design optimal and highly-personalized policies based on individual features.

In this work, I thus tackle the problem of estimating causal effects of an intervention that are heterogeneous across a population of interest and depend on an individual set of characteristics (e.g., a patient's clinical record, user's browsing history, etc..) in high-dimensional observational data settings. This is done by utilizing Bayesian Nonparametric or Probabilistic Machine Learning tools that are specifically adjusted for the causal setting and have desirable uncertainty quantification properties, with a focus on the issues of interpretability/explainability and inclusion of domain experts' prior knowledge. I begin by introducing terminology and concepts from causality and causal reasoning in the first chapter. Then I include a literature review of some of the state-of-the-art regression-based methods for heterogeneous treatment effects estimation, with an attempt to build a unifying taxonomy and lay down the finite-sample empirical properties of these models. The chapters forming the core of the dissertation instead present some novel methods addressing existing issues in individualized causal effects estimation: Chapter 3 develops both

a Bayesian tree ensemble method and a deep learning architecture to tackle interpretability, uncertainty coverage and targeted regularization; Chapter 4 instead introduces a novel multi-task Deep Kernel Learning method particularly suited for multi-outcome — multi-action scenarios. The last chapter concludes with a discussion.

Impact Statement

The desire for highly personalized decision-making, that requires to implicitly or explicitly hinge on causal reasoning and counterfactual statements, is pervasive across several disciplines both at the industrial and academic level. However, in most of these disciplines exploration of policies in the real-world through randomized experiments is often costly, harmful or simply not feasible, and thus researchers must rely on observational data. Following what has been briefly mentioned in previous paragraphs, I will describe some real-world case studies where causal machine learning methods, such as the ones developed in this work, are in high-demand and already extensively deployed.

The original motivating case study for this thesis work particularly concerned an application in the medical sciences, where the aim was to develop and deploy a model to output the best therapy among three types of treatments, two surgical and one non-surgical, for patients with history of cardiovascular diseases, based on patient-specific clinical features. This is just one of the many example in the clinical realm where causal learning is useful, as the methods developed here are fairly generalizable to any type of counterfactual questions about policy interventions, such as "what would have happened if individual i undertook treatment A instead of treatment B?", where one cannot typically rely on randomized experimental data. But this type of what-if question is typically found in several other domains. For example, recommendation systems are popular tools used in the tech and media advertising industry to suggest new contents to a user (e.g., a product to buy online, a new movie to watch, a new song/artist to listen to, etc.), where the underlying aim is to

choose an action (e.g., a product to suggest) that maximizes a certain type of reward/outcome (e.g., probability of buying the product suggested), based on the user-specific history. Another example is precision agriculture, where the aim is, e.g., to choose a combination of soil nutrients that maximize crop yield.

The ones cited above are just few of the many examples of real-world applications of causal learning, as these methods arouse interest from virtually any fields where individualized targeted policy making is ultimately of concern. This explains why they are increasingly high in demand both in the industries and in academia.

This work has led to the publication of several contributions in peer-reviewed journals, as listed in more details later.

Acknowledgements

First and foremost, I would like to thank my primary supervisor Prof. Ioanna Manolopoulou and my secondary supervisor Prof. Gianluca Baio, for their precious advice and uninterrupted encouragement throughout my PhD years, which is by no means limited to PhD related matters. It has been a pleasure to work with them and I could not recommend them highly enough.

Secondly, I would like to thank my mum and dad, and my whole family, for their unconditional love and support.

Last but not least, I extend my deep gratitude to all the staff and PhD students in the Department of Statistical Science at UCL and at the The Alan Turing Institute, and to all my close friends in London and back in Italy, who are indeed my second family and have contributed in making this journey memorable.

Publications & Working Papers

Publications from this work

- A. Caron, G. Baio and I. Manolopoulou. Estimating Individual Treatment Effects using Non-Parametric Regression Models: a Review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 2022.

- A. Caron, G. Baio and I. Manolopoulou. Shrinkage Bayesian Causal Forests for Heterogeneous Treatment Effects Estimation. *Journal of Computational & Graphical Statistics*, 2022.

- A. Caron, G. Baio and I. Manolopoulou. Interpretable Deep Causal Learning for Moderation Effects. *2nd Interpretable Machine Learning for Healthcare Workshop, ICML* 2022.

- A. Caron, G. Baio and I. Manolopoulou. Counterfactual Learning with Multioutput Deep Kernels. *Transaction on Machine Learning Research (TMLR)*, 2022.

- **[Short Discussion]** H. Zhu, X. Liu, A. Caron, I. Manolopoulou, S. Flaxman, FX. Briol. Contributed Discussion of "Bayesian Regression Tree Models for Causal Inference: Regularization, Confounding, and Heterogeneous Effects". *Bayesian Analysis*, 2020.

Under Review

- X. Liang, A. Caron, S. Livingstone and J. Griffin. "Structure Learning with Adaptive Random Neighborhood Informed MCMC". *Under Review*, 2023.

Contents

List of Figures

List of Tables

Main Nomenclature

$\boldsymbol{X}_i$	covariate vector of unit i
$\mathcal{D}_i$	dataset entry of unit i
$\mu(\cdot)$	prognostic effects function
$\pi(\cdot)$	propensity score
$\tau(\cdot)$	CATE or moderating effects function
POs	Potential Outcomes
A_i	action/treatment of unit i
$Y_i^{(a)}$	potential outcome of unit i, for action $A_i = a$
Y_i	outcome of unit i
CATE	Conditional Average Treatment Effects
PEHE	Precision in Estimating Heterogeneous Treatment Effects
SUTVA	Stable Unit Treatment Value Assumption

Chapter 1

Introduction

The use of advanced statistical learning tools (Hastie et al., 2001; Murphy, 2012) in causal inference has gained popularity in recent years, partly due to the fact that large datasets are becoming available at relatively lower costs (thanks to, e.g., electronic health records, social network data, etc.). One of the increasingly common objectives of causal inference in many disciplines is to draw inferences about *individual-level causal effects* and learn highly-personalized policies, as opposed to average across the population of interest. The importance of inferring individual-level causal effects lies in the fact that the impact of an intervention is very often heterogeneous across units of analysis, so that optimal "treatment" allocation policies need to be specifically tailored for different population's subgroups. Two such examples arise in precision medicine (Collins and Varmus, 2015; Hodson, 2016) and targeted advertising domains (Cheung et al., 2003), where the ultimate goal is to make different decisions for each specific patient or user. For instance, patients with high cholesterol levels respond differently to statin prescriptions based on their age, gender, comorbidities, etc. This type of analysis requires causal methods that can accurately estimate and predict the impact of such treatment at a fine resolution, as well as quantify uncertainty around it. To this end, popular statistical learning algorithms that exhibit excellent predictive performance, such as tree ensembles, kernel methods and neural networks, can be exploited also, with due adjustments, in these causal settings (Künzel et al., 2017; Caron et al., 2022a). Additionally,

Bayesian nonparametrics offer a toolbox of flexible and complex methods with desirable Bayesian coverage properties for uncertainty quantification. This work is therefore motivated by the growing number of (Bayesian) non-parametric regression methods for modelling **Individual (Causal) Treatment Effects** (ITE) using large datasets to answer individual-level counterfactual questions.

The fundamental problem of causal inference is that the random variable of interest — namely the causal effect — is never directly observable, thus standard supervised learning techniques cannot be directly applied. Moreover, in the context of observational studies, the mechanism driving treatment allocation is usually unknown and can obscure the causal effect of interest through confounding factors (Dawid, 2000; Pearl, 2009a; Imbens and Rubin, 2015). Confounding factors are observed or unobserved common causes of both treatment selection and outcome. The "latent" treatment effect needs to be inferred then by reconstructing counterfactual statements through randomization in the experimental design phase, or in the case of observational data resort to, e.g., importance sampling (via weighting methods), matching, and/or regression adjustment, provided that identification is actually assumed to be possible.

Randomized experiments, where treatment is randomly allocated, marginally on confounding factors (such as clinical history or socio-economic characteristics), are considered to be the gold standard in causal inference. However, exploration of policies in the real-world through randomized studies is generally costly, harmful, or simply unfeasible. Furthermore, randomized data sometimes suffer from problems such as non-compliance and missingness that might invalidate the randomization mechanism. In contrast, data of observational nature, where policy allocation is not randomized, are more easily accessible and abundantly present in many applied fields. However, observational studies present several drawbacks, largely attributable to *sample selection bias*, which manifests when the treatment allocation mechanism is not under the researcher's control, but determined by other factors, which may be observable and/or unobservable. This constitutes a potential source of confounding, if

related also to the outcome of interest, that needs to be controlled for, as it generates structural differences between the different intervention arms/groups. Similarly, selection bias also generates *partial overlap* problems, that occur when there are regions in the space of relevant confounding covariates where units belonging to a specific treatment arm are virtually absent. As a result, this creates imbalances due to the fact that, in these regions, the researcher does not have access to appropriate comparison units, with those underlying characteristics. Finally, although we do not specifically address this scenario in this work, selection bias might also generate problems related to the fact that treatment allocations and corresponding outcomes may not be independent across individuals, i.e., sample displays interference between the units; we shall briefly discuss this case of network interference in the last sections of this work.

We proceed by briefly summarizing contents in this manuscript. The current chapter, Chapter 1, introduces the problem of causal learning and causal effect identification, accompanied by the relative possible mathematical notations. In particular, we describe three different frameworks for causal modelling: i) the Neyman-Rubin Causal Model (Rubin, 1974, 1978; Imbens and Rubin, 2015); ii) *do*-calculus and graphical causal models (Pearl, 2009a; Koller and Friedman, 2009; Peters et al., 2017); iii) the decision-theoretic approach (Dawid, 2000, 2015). The focus will be primarily on the first two. Chapter 2 then specifically outlines the problem of estimating Individual Treatment Effects (ITEs) via (non-parametric) regression-based techniques and reviews some of the most recent methods, by casting them under the same unifying taxonomy. Implied assumptions and finite-sample properties of each method are empirically assessed, and their performance compared, via simulated experiments. A practical demonstration of the two best performing methods emerging from the simulations is presented as well; this is based on a real-world study on the effect of participation to school meal programs on students' health indicators. Chapter 3 focuses specifically on Bayesian nonparametric and Probabilistic Machine Learning regression techniques and presents two

novel methods that combine three main desiderata when it comes to highly-personalized policy making, that most existing black-box causal methods failed to comply with: i) **Interpretability**: existing causal ML models do not produce any interpretable measure of importance as to what are the main moderators, among the observed covariates, of the heterogeneity behind the response to a treatment; ii) **Targeted regularization/shrinkage**: most causal ML models are incapable to convey carefully tailored regularization, and prior knowledge in a Bayesian perspective, directly on the quantity of interest, i.e., causal effects, and often end up generating unintended bias in the estimates. iii) **Uncertainty quantification**: for similar reasons to point ii), these models do not directly produce appropriate Bayesian uncertainty intervals around causal effects point estimates. The two models presented are respectively based on a recently developed Bayesian tree-ensemble model, named Bayesian Additive Regression Trees (BART) (Chipman et al., 1998, 2010), and on a more interpretable variant of Neural Networks (NNs), theoretically grounded as Generalized Additive Model (GAM), called Neural Additive Models (NAMs) (Agarwal et al., 2021), coupled with approximate Bayesian deep learning inference techniques (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Pearce et al., 2020; Abdar et al., 2021). Chapter 4 then introduces a novel Bayesian nonparametric causal model based on multitask Gaussian Processes (Rasmussen and Williams, 2005; Alvarez et al., 2012) and Deep Kernel Learning (Wilson et al., 2016), to specifically tackle settings with high-dimensional data along multiple axes, i.e., settings with large covariate space, multiple treatment arms and also multiple outcomes of interest, where existing methods would incur in sample efficiency losses and scalability issues. The manuscript concludes with Chapter 5 by summarizing and outlining, existing or potential, interesting research directions, mainly revolving around how to relax some of the main assumptions and discussing the problem of "causal discovery" or Bayesian structure learning.

1.1 Statistical and Causal Learning

In this section we briefly highlight and formalize the fundamental philosophical differences underpinning statistical and causal learning, in order to give more sound foundations to the concepts introduced in the earlier paragraphs of this chapter (Peters et al., 2017). Statistical Learning (Vapnik, 1999, 2000), very loosely speaking, is concerned with discovering the statistical properties of random variables in terms of their dependence, leveraging a given (finite) sample of data indexed by $\{1, ..., n\}$, where n denotes sample size. For example, given an independent and identically distributed (iid) sample consisting in the pair of random variables $\{X_i, Y_i\}_{i=1}^n \overset{\text{iid}}{\sim} p_{X,Y}(\cdot)$, where $p_{X,Y}(\cdot)$ is a joint probability distribution, the goal is to learn a predictor for estimating Y_i given X_i. Here $X_i \in \mathcal{X}$ is a covariate (or input/explanatory variable) and $Y_i \in \mathcal{Y}$ is the outcome (or label). This predictor $f : \mathcal{X} \to \mathcal{Y}$ is typically learnt as the the minimizer, among a set of possible functionals $\mathcal{F}$, of a type of associated "risk" (**risk minimization**), i.e.:

$$f^* \in \arg\min_{f \in \mathcal{F}} \mathcal{R}^{\text{true}}_{p_{X,Y}}(f)$$

$$\mathcal{R}^{\text{true}}_{p_{X,Y}}(f) = \mathbb{E}_{p_{X,Y}}\big[\mathcal{L}(f(x), y)\big] = \int_{(\mathcal{X},\mathcal{Y})} \mathcal{L}(f(x), y)\, p(x, y)\, dxdy \ ,$$

where $p(\cdot)$ is the true distribution over x and y and $\mathcal{L}(f(x), y)$ is a loss function that depends on the type of statistical problem at hand (e.g., regression, classification, etc.). Now, since the true joint distribution $p_{X,Y}(\cdot)$ is not observable, one has to resort to approximation with its sample equivalent $\hat{\mathcal{R}}^{\text{emp}}_{p_{X,Y}}(f) = \frac{1}{n}\sum_{i=1}^n \mathcal{L}(f(x_i), y_i)$ (hence the term **empirical risk minimization**), and train the model based on this estimate instead of the ground-truth. The available sample $\{x_i, y_i\}_{i=1}^n$ consists of observations that are realizations of a random experiment whose statistical and probabilistic properties, captured by the joint probability $p_{X,Y}(\cdot)$, are unknown. This what defines an **inverse problem**. If the data satisfy the iid assumption, and provided that the minimizer f^* of the true risk lives in the (possibly restricted) function class $f^* \in \mathcal{F}$,

a desirable asymptotic property of a certain supervised learning algorithm, that selects a certain function $f \in \mathcal{F}$, is consistency, i.e. empirical risk converges to the true risk $\hat{\mathcal{R}}^{\text{emp}}_{p_{X,Y}}(f) \xrightarrow{p} \mathcal{R}^{\text{true},*}_{p_{X,Y}}(f^*)$, and the true minimizer $f^* \in \mathcal{F}$ gets selected, as $n \to \infty$ and the *estimation error* goes to zero. Restricting the class of candidate functionals helps as asymptotic convergence might be very slow for large functional spaces, but on the opposite side, if the restricted class of functions $\mathcal{F}$ does not contain the true function f^* one incurs into an *approximation error* in addition to an *estimation error*. Typical restrictions to the functional class space $\mathcal{F}$ include additive separability (e.g., of noise term), regularizers, prior distributions, etc. The success of the empirical risk minimization paradigm relies then on the satisfaction of the iid assumption, trade-off between faster asymptotic convergence and low approximation error in choosing the restricted class $\mathcal{F}$, and on the complexity of the true distribution $p_{X,Y}(\cdot)$.

In Causal Learning (Pearl, 2009a; Imbens and Rubin, 2015; Peters et al., 2017), where the goal is to ultimately retrieve the cause-effect relationships (Blaisdell and Miller, 2012) not just purely probabilistic ones, one is faced with the challenge of a typical statistical learning inverse-problem, plus an additional layer of complexity given by the fact that, even under full knowledge of the true distribution $p_{X,Y}(\cdot)$, the underlying causal model might not be learnable. In fact, $p_{X,Y}(\cdot)$ is effectively an observational distribution, which implies that data from $p_{X,Y}(\cdot)$ are collected, loosely speaking, by passively observing the outcome of a random experiment in an environment. Instead, causal modelling is concerned with learning about **interventional distributions**, that is, about (probabilistic, still) effects of manipulative interventions or distribution shifts on a system (Peters et al.,

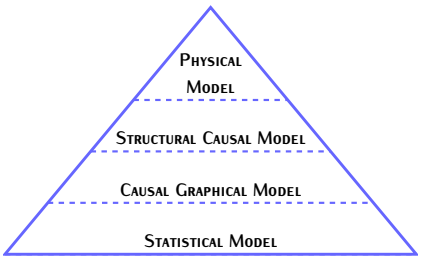


Figure 1.1: A simple hierarchy of models, in terms of their accuracy in depicting relationships between variables.

2017), thus relating to a deeper level of understanding than just the statistical properties. This locates causal models a step closer to the most accurate type of knowledge of a system, represented by the mechanical/physical model (Peters et al., 2017; Schölkopf et al., 2021).

Figure 1.1 on the side depicts a simple hierarchy of models, departing from the most accurate mechanical/physical models, which includes (from top to bottom): i) pure statistical model, that cannot answer counterfactual questions, nor predict under a distribution shift, but can be learnt from observational data; ii) causal graphical model, i.e., graphs where arrows represent causal relationships, that can predict under a distributional shift additionally, meaning when an observed intervention changes the joint distribution $p_{X,Y}(\cdot)$; iii) structural causal models, i.e., system of equations expressing approximate functional relationships between the variables of a system, that can also answer counterfactual "what-if" questions (about why things happened); iv) physical/mechanical model, i.e., usually a system of differential equations, that can do all the previous things, plus fully describe physical properties of a system. Physical models become increasingly hard, if not impossible, to even formulate as complexity of the system grows. Thus, causal models lie somewhere in the middle of the hierarchy, and have additional desirable properties compared to statistical ones. However, the causal learning problem inherits the hardness of a statistical learning task, where one ought to make a first set of assumptions to e.g. have iid data, restrict the functional class, etc., and, on top of that, necessitates additional assumptions to *identify* the underlying causal properties of a system from purely observational data, flawed by confounding sample bias. These causal assumptions are laid out and discussed in the following sections, under three equivalent mathematical frameworks for causal modelling.

1.2 Causal Modelling Frameworks

Identifying causal effects with fully randomized experimental data is relatively less burdensome than with observational ones, as we get access to the (uncon-

founded) "interventional" state of the world, where the causal effects (effects of manipulative interventions) of interest are, by construction, isolated from any other spurious associations (e.g., correlations between two non-stationary time series). In the realm of observational data, assumptions are instead necessary in order to remove confounding and effectively derive unbiased estimates of causal effects by using observational quantities. In this section, we mainly introduce the mathematical notation to represent causal notions, which has some additional features compared to the purely statistical/probabilistic notation, and we discuss causal assumptions necessary for the identification of causal effects. The mathematical notation utilized for causal reasoning is not unique. There are different, albeit equivalent, ways, or frameworks, for representing interventional distributions. We begin by introducing the Neyman-Rubin Causal Model framework (also known as the Potential Outcomes framework), popular in e.g., the statistical, econometric and social sciences literature. Then we proceed by discussing Pearl's graphical approach, which relies on graphical representation of Structural Causal Models via Bayesian Networks (Koller and Friedman, 2009) and *do*-calculus, more popular, e.g., in the computer science literature. Finally, we briefly comment on a third framework, represented by Dawid's decision-theoretic approach.

1.2.1 The Neyman-Rubin Causal Model

The Neyman-Rubin Causal Model, also known as the Potential Outcomes (POs) causal model, (Neyman, 1923; Rubin, 1974, 1978; Splawa-Neyman et al., 1990; Imbens and Rubin, 2015), very loosely speaking, conceives causal inference as a structural missing data problem. For each unit of analysis $i \in \{1, ..., n\}$, we first define $A_i \in \mathcal{A}$ as the action (or treatment) variable, which consists of a manipulative variable that we can artificially intervene on. We will restrict our analysis to discrete $\mathcal{A}$ spaces, in particular to scenarios with a binary action indicator $\mathcal{A} = \{0, 1\}$, where $A_i = 1$ indicates exposure to, e.g., a treatment and $A_i = 0$ indicates no exposure. However, notice that most of the regression-adjustment methods reviewed in later sections can

potentially be extended to the case of multi-arm discrete action space. Given $A_i \in \mathcal{A} = \{0, 1\}$, the framework defines the quantities $\left(Y_i^{(0)}, Y_i^{(1)}\right)$ as Potential Outcomes, where $Y_i^{(1)}$ corresponds to unit i's outcome under exposure to the treatment, while $Y_i^{(0)}$ corresponds to its outcome under no exposure. The fundamental problem of causal inference is that, for each i, only one of the two POs is actually observable, the "factual" outcome, while the other potential outcome is effectively a "counterfactual". The observed factual outcome Y_i then is the one corresponding to the specific realization of the action variable (i.e., the action performed): $Y_i = A_i Y_i^{(1)} + (1 - A_i)Y_i^{(0)}$. We will generally consider, throughout this work, a setting where the outcome variable of interest is continuous, i.e., $\left(Y_i^{(0)}, Y_i^{(1)}\right) \in \mathbb{R}^2$, but again most of the ideas and methods presented can in principle be generalized to other types of outcome. We denote then by $\boldsymbol{X}_i \in \mathcal{X}$ a potentially high-dimensional set of observed covariates (e.g., patient's or user's characteristics). In the case of observational studies, a subset of observed covariates $\tilde{\boldsymbol{X}}_i \subseteq \boldsymbol{X}_i$ constitute a possible source of confounding, as that they may simultaneously determine the action A_i and the outcome Y_i (common causes). There could also be unobserved sources of confounding $U_i \in \mathcal{U}$, still common causes of Y_i and A_i, that might impede identification. Scenarios with unobserved confounders need to be addressed in a different way, which entails a different set of assumptions (e.g., via instrumental variables). However, throughout this work we will assume that we observe sufficient variation in the high-dimensional covariates to capture the confounding effects. Finally, we denote the main quantity of interest $Y_i^{(1)} - Y_i^{(0)}$ as the *Individual Treatment (or Causal) Effects* (ITEs), that is, the (never observable) effect of intervening on A_i, artificially setting it to $A_i = 1$ instead of 0, on the same individual i. Provided we have access to either observational or randomized experimental data $\mathcal{D}_i = \{\boldsymbol{X}_i, A_i, Y_i\}$, with $i \in \{1, ..., n\}$, the general aim is to efficiently derive estimates for moments of the $Y_i^{(1)} - Y_i^{(0)}$ interventional distribution. Two moments of primary interest are the Conditional Average

Treatment Effect (CATE) and the Average Treatment Effect (ATE), defined as

$$\text{CATE:} \quad \tau(\boldsymbol{x}_i) = \mathbb{E}\Big[Y_i^{(1)} - Y_i^{(0)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\Big] = \mu_1(\boldsymbol{x}_i) - \mu_0(\boldsymbol{x}_i) \quad \text{and} \tag{1.1}$$

$$\text{ATE:} \qquad \tau = \mathbb{E}_X\big[\tau(\boldsymbol{x}_i)\big] = \mathbb{E}_X\big[Y_i^{(1)} - Y_i^{(0)}\big] \ , \tag{1.2}$$

where the two quantities $\mu_1(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i^{(1)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\big]$ and $\mu_0(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i^{(0)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\big]$ in (1.1) are the conditional average potential outcomes. The intuition behind the estimation of CATE $\tau(\boldsymbol{x}_i)$ is the following. In case both potential outcomes were observed, then $Y_i^{(1)} - Y_i^{(0)}$ (ITE) would be modelled as the response variable in a supervised regression framework where $\boldsymbol{X}_i$ are the d regressors, and where the aim is to estimate the conditional mean of the outcome, namely $\tau(\boldsymbol{x}_i) = \mathbb{E}[Y_i^{(1)} - Y_i^{(0)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i]$. However, since only $Y_i = A_i Y_i^{(1)} + (1 - A_i) Y_i^{(0)}$ is observable, direct application of supervised regression models is not possible. The set of regressors $\boldsymbol{X}_i$ here does not necessarily include all the available covariates, but only moderators of treatment effects responsible for heterogeneity in the response. We will discuss in later sections how detecting moderators is a particularly insightful part of the analysis of heterogeneous moderation effects. Figure 1.2 provides a graphical representation of a simple single-covariate example, where coloured dots show observed values $Y_i = Y^{(A_i)}$ of the response, while grey dots their corresponding (unobservable) counterfactuals $Y_i^{(1-A_i)}$. Notice that the example is purely illustrative and serves as visual aid to introduce the reader to the key concepts in the Rubin-Neyman framework.

An additional quantity of interest under this framework is the *propensity score*, which is defined, for each unit of analysis i, as the probability of being selected into treatment ($A_i = 1$), given a set of observed covariates that we denote by $\tilde{\boldsymbol{X}}_i = \tilde{\boldsymbol{x}}_i$, to stress the fact that it might be different from the set of covariates used for the estimation of $\tau(\boldsymbol{x}_i)$; more formally:

$$\pi(\boldsymbol{x}_i) = \mathbb{P}(A_i = 1 \mid \boldsymbol{X}_i = \boldsymbol{x}_i) \ .$$

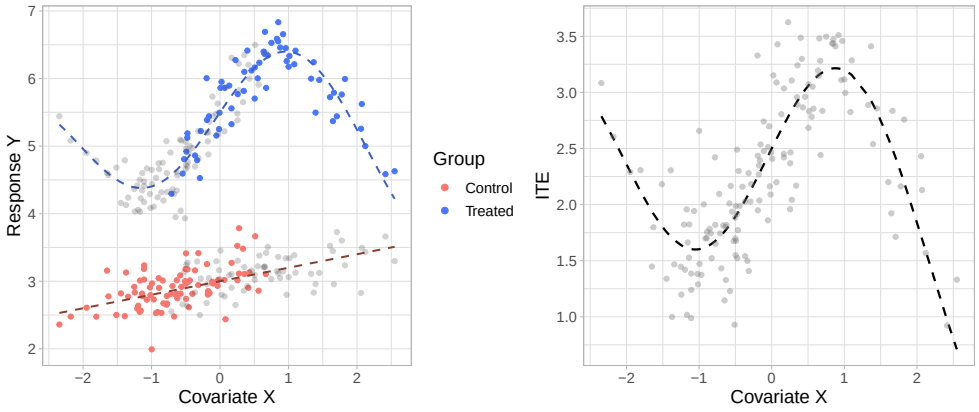


Figure 1.2: Simulated example with one single covariate X. Potential outcomes are generated respectively as $Y_i^{(0)} \sim \mathcal{N}(3 + 0.2X_i, 0.25)$ and $Y_i^{(1)} \sim \mathcal{N}(5.5 - 0.1X_i^2 + \sin(1.5X_i), 0.25)$, while the propensity score is $\pi(X_i) = \Phi(-0.2 + 1.5X_i)$, where $\Phi(\cdot)$ is the standard normal cdf. Left panel: observed outcomes for the treated (blue dots) and control (red dots) groups with underlying conditional mean function (dashed lines), and unobservable counterfactual outcomes (grey dots). Right panel: unobservable true ITE (grey dots) and corresponding conditional mean function (dashed line).

Thus, for each unit, the binary treatment assignment A_i can be seen as the outcome of a Bernoulli experiment where $A_i \sim \text{Bernoulli}\big(\pi(\boldsymbol{x}_i)\big)$. In the simple example of Figure 1.2, the propensity score is generated as a monotone function of the covariate X. This is why treated units are more frequently observed for higher values of X, while control units are more frequent for lower values of X. The propensity score distribution in this case is also slightly skewed to the right; as a consequence, there are more units in the control group compared to the treated one.

The least stringent set of assumptions needed to identify and estimate ATE and CATE, in fully randomized experiments, is made of:

a) **Stable Unit Treatment Value Assumption (SUTVA)** . Under POs notation SUTVA is formally defined as $Y_i^{(a_1,\dots,a_i,\dots,a_n)} = Y_i^{(a_i)}$, and states that the potential response to treatment of unit i is not affected by other units' assignment to treatment $\forall i \in \{1, \dots, n\}$. When SUTVA fails to holds,

identification and estimation of causal effects become more challenging. Nonetheless, many contributions in the literature have attempted to relax this assumption (Hudgens and Halloran, 2008; Tchetgen and VanderWeele, 2010; Aronow and Samii, 2017). In the last chapter of the manuscript, we will briefly introduce and discuss possible extension to recent work on causal inference under networked interference (Toulis and Kao, 2013; Forastiere et al., 2021; Ma and Tresp, 2021).

b) **Consistency**. Consistency assumption guarantees that the observed outcome corresponds to one realization of the treatment only, formally if $A_i = a_i$ then $Y_i = Y_i^{(a_i)}$, thus ensuring that there are no multiple versions of the same treatment.

The two assumptions above imply absence of any interference between the units of analysis, and are necessary and sufficient for identification in completely interventional scenarios, where A_i is under the control of the researcher, but not in observational ones. In the case of observational studies, an additional set of assumptions is required to address the problem generated by sample selection bias. These are:

c) **Unconfoundedness**. Also known as POs conditional independence, it states that there is no *unobserved* common cause affecting selection into treatment and outcome simultaneously. Under the POs framework this translates into

$$\left(Y_i^{(0)}, Y_i^{(1)}\right) \perp\!\!\!\perp A_i \mid \boldsymbol{X}_i \ , \tag{1.3}$$

thus ensuring that conditioning on $\boldsymbol{X}$ suffices to derive unbiased estimates of the causal effect $A \to Y$, provided that the *Common Support* assumption, discussed in the next point below, also holds. A well-known result concerning the propensity score, derived by Rosenbaum and Rubin (1983), is that if (1.3) holds, then

$$\left(Y_i^{(0)}, Y_i^{(1)}\right) \perp\!\!\!\perp A_i \mid \pi(\boldsymbol{X}_i) \ . \tag{1.4}$$

Conditional independence in (1.4) represents an equivalent way of expressing unconfoundedness, by conditioning only on the propensity score rather than on the full set of covariates. In a context where $\boldsymbol{X}_i$ is high-dimensional, $\pi(\boldsymbol{X}_i)$ represents a 1-dimensional representation of a d-dimensional covariate set that, in theory, equally achieves conditional independence between Y_i and A_i; its success in practice, though, is highly dependent on how well we can approximate the propensity via a supervised model $\hat{\pi}(\boldsymbol{X}_i)$. Unconfoudedness might be, in some cases, a strained assumption to make (in complex systems such as socio-economic ones), but represents less of a threat in settings where $\boldsymbol{X}_i$ is sufficiently rich and thus likely include the relevant confounders, or least a sufficient set of proxies for them. Formal identification strategy via proxies is thoroughly described e.g. in Tchetgen et al. (2020), and addresses a setting where some confounders in $\boldsymbol{X}_i$ are unmeasurable but their variability is well enough explained by some measurable proxy W_i. A scenario where unconfoundedness then fails to hold is when there are unobserved common causes U_i of A_i and Y_i, that do not admit proxies. In these cases, identification might be achievable via Instrumental Variables (IV) (Angrist et al., 1996; Pearl, 2009a), where generally an observed covariate L_i (or generated (Wang and Blei, 2019)) that does not share any unobserved common cause with Y_i and sufficiently explains variability in A_i is utilized as an instrument instead of A_i. We will not address proximal nor instrumental settings in this work, that merit their own specific discussion.

d) **Common Support**, also known as Positivity or Overlap. It states that each unit i, identified by a given set of covariates $X_i = x_i$, has non-zero probability of being observed in the two treatment groups. In other words, common support means there is no deterministic selection into treatment, thus ensuring that there are no regions of the covariate space where only treated or control units are observable. Under the framework outlined above, common support assumption implies that propensity score satisfies $0 < \pi(\boldsymbol{x}_i) < 1$, for each i. This guarantees the existence of CATE for every

$X_i = x_i$. As highlighted in later paragraphs, overlap does not have to hold for the whole support $\mathcal{X}$, but causal effects are identifiable only when in regions where it does.

Under unconfoundedness and common support assumptions, CATE (and consequently ATE) can be estimated from purely observational data in $\mathcal{D}_i = \{\boldsymbol{X}_i, A_i, Y_i\}$ $i \in \{1, ..., n\}$. In fact, under unconfoudedness, we have that

$$\begin{aligned} \mu_A(\boldsymbol{x}_i) &= \mathbb{E}\left[Y^{(A_i)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\right] = \mathbb{E}\left[Y^{(A_i)} \mid A_i = a_i, \boldsymbol{X}_i = \boldsymbol{x}_i\right] \\ &= \mathbb{E}\left[Y_i \mid A_i = a_i, \boldsymbol{X}_i = \boldsymbol{x}_i\right] \ , \end{aligned} \tag{1.5}$$

for $a_i \in \{0, 1\}$, where the second equality arises from unconfoundedness, while the last one from the identity $Y_i = A_i Y_i^{(1)} + (1 - A_i) Y_i^{(0)}$ and the consistency assumption. As a straightforward implication of (1.5), one can identify CATE as

$$\begin{aligned} \tau(\boldsymbol{x}_i) &= \mathbb{E}\left[Y_i^{(1)} - Y_i^{(0)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\right] = \\ &= \mathbb{E}\left[Y_i^{(1)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\right] - \mathbb{E}\left[Y_i^{(0)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\right] \\ &= \mathbb{E}\left[Y_i^{(1)} \mid A_i = 1, \boldsymbol{X}_i = \boldsymbol{x}_i\right] - \mathbb{E}\left[Y_i^{(0)} \mid A_i = 0, \boldsymbol{X}_i = \boldsymbol{x}_i\right] \\ &= \mathbb{E}\left[Y_i \mid A_i = 1, \boldsymbol{X}_i = \boldsymbol{x}_i\right] - \mathbb{E}\left[Y_i \mid A_i = 0, \boldsymbol{X}_i = \boldsymbol{x}_i\right] \ . \end{aligned} \tag{1.6}$$

where, as in (1.5), the third equality is given by unconfoundedness, and the last one by the identity $Y_i = A_i Y_i^{(1)} + (1 - A_i) Y_i^{(0)}$. Common support assumption is then needed to guarantee that the two conditional average potential outcomes $\mu_1(\boldsymbol{x}_i) = \mathbb{E}\left[Y_i^{(1)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\right]$ and $\mu_0(\boldsymbol{x}_i) = \mathbb{E}\left[Y_i^{(0)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\right]$ theoretically exist for each values $\boldsymbol{x}_i$ in their supports, and thus can be estimated through the observed quantities in the conditional expectations $\mathbb{E}\left[Y_i \mid A_i = 1, \boldsymbol{X}_i = \boldsymbol{x}_i\right]$ and $\mathbb{E}\left[Y_i \mid A_i = 0, \boldsymbol{X}_i = \boldsymbol{x}_i\right]$ respectively. To clarify, suppose that common support does not hold for $\boldsymbol{X}_i = x^*$ and that $\pi(x^*) = 0$ (without loss of generality), then the conditional average potential outcome $\mu_1(x^*) = \mathbb{E}\left[Y^{(1)} \mid \boldsymbol{X}_i = x^*\right]$ does not theoretically exist, and it would not make sense to attempt to even estimate it.

In empirical studies of observational nature, common support sometimes

fails to hold for the whole support of $\boldsymbol{X}_i$. Hence, CATE can only be identified and reliably estimated in some specific regions only. In particular, we might refer to estimands such as *Average*, or *Conditional Average*, *Treatment Effect on the Treated* (ATT and CATT, respectively) to indicate when treatment effects are identifiable only on the treated group, or equivalently when common support only holds for the treated units. The same holds for *Average*, or *Conditional Average*, *Treatment Effect on the Control* (ATC and CATC). In practice, empirical methods such as propensity score re-weighting and trimming are generally utilized in order to focus estimation on overlap regions only; also, Bayesian inference can be used as tools for assessing the lack of overlap (Hill and Su, 2013).

As mentioned briefly in earlier paragraphs, this work particularly focuses on estimation of causal effects in observational scenarios where unconfoundedness and overlap assumptions hold. In the next two section, we will discuss how identification of causal effects admits equal representation under two other frameworks different than the Rubin Causal Model.

1.2.2 *do*-calculus and causal DAGs

The use of causal graphical models was mainly pioneered by computer scientist Judea Pearl (Pearl, 2009a,b; Geffner et al., 2022), and together with the Potential Outcomes framework remains one of the most popular tools for causal reasoning. We will give an overview of the main concepts in probabilistic graphical modelling (Koller and Friedman, 2009) and their causal declination, coupled with Pearl's *do*-calculus.

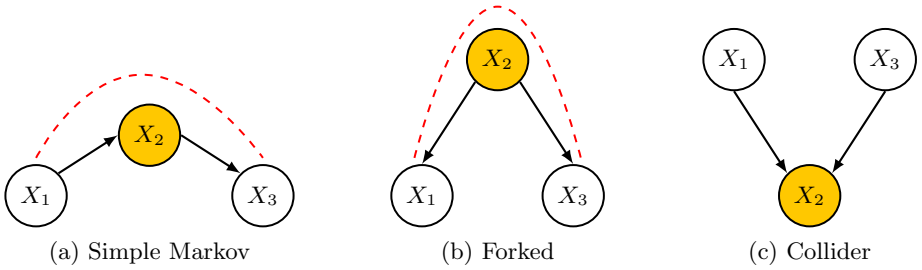
A **graph** is a mathematical structure consisting of vertices $\mathcal{V}$ and edges $\mathcal{E}$, $G = (\mathcal{V}, \mathcal{E})$, suitable to represent probabilistic dependencies (edges) between random variables (nodes) X_v, $v \in \mathcal{V}$ through a probabilistic graphical model object. In particular, one can distinguish between undirected graphical models (or Markov random fields) and directed graphical models, based on whether the edges $\mathcal{E}$ are directional. Within the class of directed graphs then, **Directed Acyclic Graphs** (DAGs) assume an important role in describing

causal relationships, via Bayesian networks. A **Bayesian Network** $\mathcal{B} = (G, \Phi)$ then is a probabilistic graphical model capturing conditional independence relationships among a collection of random variables, that can be represented by a DAG G; Φ represents the parameter space (Koller and Friedman, 2009). Hence, given a collection of d random variables $(\boldsymbol{X}_1, ..., \boldsymbol{X}_d)$, a Bayesian network is generally utilized to represent a factorization of the joint probability $p(\boldsymbol{X}_1, ..., \boldsymbol{X}_d)$ through DAG G, where nodes represent the variables and edges their relationships; while Φ are the parameters of the conditional distributions:

$$p(\boldsymbol{X}_1, ..., \boldsymbol{X}_d) = \prod_{i=1}^{d} p\big(\boldsymbol{X}_i \mid pa(\boldsymbol{X}_i)\big)\,, \tag{1.7}$$

where $pa(\boldsymbol{X}_i)$ denotes all parent nodes of $\boldsymbol{X}_i$.

Thus, the two main underlying assumptions required to convert a normal DAG $\mathcal{G}$ into a causal one are: i) **Local Markov Assumption**, which states that a node/variable X_i is independent of all its non-descendant, given its parent nodes; this essentially allows to express the joint probability of nodes $X_1, ..., X_n$ through the Bayesian network factorization as $p(\boldsymbol{X}_1, ..., \boldsymbol{X}_d) = \prod_{i=1}^{d} p\big(\boldsymbol{X}_i \mid pa(\boldsymbol{X}_i)\big)$, where $pa(\boldsymbol{X}_i)$ denotes the "parents" of node X_i (**faithfulness** assumption, needed for example in causal discovery, instead points in the opposite direction: conditional independencies in the data imply d-separation in the corresponding DAG, i.e. $X_i \perp\!\!\!\perp_{\mathbb{P}} X_j \mid X_k \implies X_i \perp\!\!\!\perp_{\mathcal{G}} X_j \mid X_k$); ii) **Causal Edges Assumption**, which simply states that parent nodes are direct causes of their children nodes — or, more informally, that arrows in a DAG exclusively represent causal relationships. Given these two assumptions, we generally say that there is a flow of association between two nodes X_i and X_j if there is a path of directional edges (or arrows) connecting them (also indirectly via other nodes). Figure 1.3 depicts three different cases of relationships between the variables X_1, X_2 and X_3. The dashed red line in the first two DAGs indicates that there is a flow of association between X_1 and X_3, while this is not present in (c).



(a) Simple Markov (b) Forked (c) Collider

Figure 1.3: Examples of DAGs representing different types of relationship. The dashed red line in (a) and (b) represents flow of association (null in the last case). The node containing X_2 is coloured differently to highlight the fact that conditioning on X_2 blocks the flow of association between X_1 and X_3 in (a) and (b), while it unblocks it in (c).

In the causal DAG language, conditioning allows one to block (or unblock) flows of association, and consequently single out desired paths of causality. For instance, in Figure 1.3b) conditioning on X_2 blocks the path, and thus the flow of association, between X_1 and X_3 (dashed red line). Equivalently said, X_2 *d-separates* the other two nodes, written as $X_1 \perp\!\!\!\perp_{\mathcal{G}} X_3 \mid X_2$ (where $\mathcal{G}$ stands for "in the DAG"). D-separation implies conditional independence in distribution $X_1 \perp\!\!\!\perp_{\mathcal{G}} X_3 \mid X_2 \implies X_1 \perp\!\!\!\perp_{\mathbb{P}} X_3 \mid X_2$ under the Local Markov assumption.

The DAG in Figure 1.3c) instead depicts a case where X_2 is known as a "collider", i.e. a child with more than one parent. Conditioning on X_2 here unblocks flow of association between X_1 and X_3, $X_1 \not\perp\!\!\!\perp_{\mathcal{G}} X_3 \mid X_2$, that would be otherwise absent, $X_1 \perp\!\!\!\perp_{\mathcal{G}} X_3$. This example is useful to understand why particular care is required when deciding the conditioning set to identify a specific causal path, as conditioning on a collider could lead to unblocking of secondary non-causal associational paths (known as *collider bias*). This explains why it is always advised not to condition on *post-treatment* covariates, as they might be common children of A and Y. However, it is neither

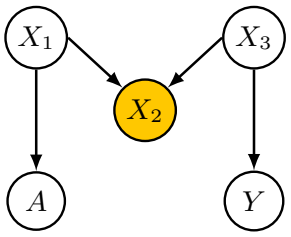


Figure 1.4: DAG representing M-bias, where X_2 is a collider.

generally advised to condition on all the pre-treatment covariates, as collider bias might arise also among them. The typical example is in causal DAGs with an M-structure (hence the name "M-bias"), like the one depicted in Figure on the side, where conditioning on X_2 unblocks an otherwise non-existent path from treatment A to outcome Y.

After having introduced few basic concepts about causal DAGs above, we will see how they offer a rather intuitive framework for causal reasoning and causal effects identification. We do so by first defining the quantity $p\big(Y = y \mid do(A = a)\big)$ as the interventional distribution, where the notation $do(A = a)$ (the *do*-operator) indicates that we intervene on A by "manually" setting it equal to a. The intervetional distribution, conceptually different than the conditional distribution $p(Y = y \mid A = a)$, is what we seek to derive. In particular, considering a binary action/treatment $A_i \in \{0, 1\}$, we are interested in ATE and CATE as moments of this distribution, as defined in the POs framework above:

$$\begin{aligned} \text{ATE:} \quad & \mathbb{E}\big[Y \mid do(A = 1)\big] - \mathbb{E}\big[Y \mid do(A = 0)\big] \\ \text{CATE:} \quad & \mathbb{E}\big[Y \mid do(A = 1), X\big] - \mathbb{E}\big[Y \mid do(A = 0), X\big] \ . \end{aligned} \tag{1.8}$$

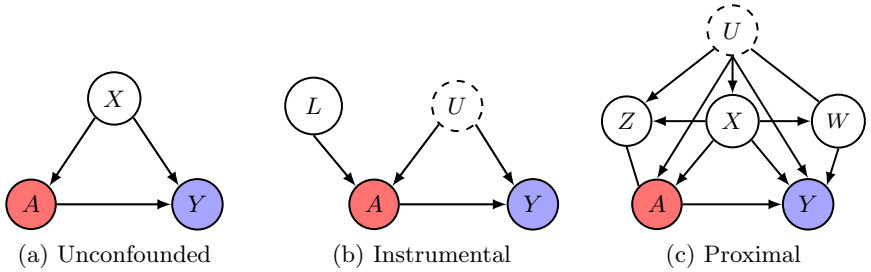
Intervening on A by setting its value equal to a — indicated using the notation $do(A = a)$ — implies overriding the causal mechanism behind it, determined by its parent nodes via $p\big(A \mid pa(A)\big)$. This translates graphically into all the incoming directional edges to A (and to A only, every other mechanism is unchanged; this is known also as "modularity assumption")) being "pruned" or canceled out, such that resulting causal DAG no longer represents the factorization of the full joint probability in (1.7). In fully randomized experiments, the conditional and interventional distributions are equivalent by construction, as one essentially has intervened on A in the experimental design phase, thus: $p(Y = y \mid A = a) \equiv p\big(Y = y \mid do(A = a)\big)$. In observational studies this is not the case, and in order to identify $p\big(Y = y \mid do(A = a)\big)$ through observational quantities we need to condition on all the direct common causes (common

parents) of A and Y to block flows of association different than the direct causal association between them — exactly as in the POs framework. In causal DAG terminology this is often refered to as **backdoor adjustment**, in that to achieve identification we need to block all the "backdoor paths" from A to Y to isolate the direct causal relationship (Pearl, 2009a). In the example of Figure 1.5a) in particular, X is said to satisfy the *backdoor criterion*, i.e. it blocks all the backdoor paths between A and Y, thus ruling out the presence of hidden commom causes that would invalidate the identification mechanism. For this reason, in this case, conditioning on X is sufficient to identify $p\big(Y \mid do(A = a)\big)$ as

$$\begin{aligned} p\big(Y \mid do(A = a)\big) &= \sum_{x \in \mathcal{X}} p\big(Y \mid do(A = a), X\big) p\big(X \mid do(A = a)\big) = \\ &= \sum_{x \in \mathcal{X}} p\big(Y \mid a, X\big) p\big(X \mid do(A = a)\big) = \\ &= \sum_{x \in \mathcal{X}} p(Y \mid a, X) p(X) \ , \end{aligned} \tag{1.9}$$

where the first equality is by the law of total probability, the second is given by conditioning on X (which allows to recover the interventional distribution) and the last by the fact that intervening on A cancels out the causal edge incoming to A from X.

Identification is possible also in settings where the *backdoor criterion* is not satisfied, such as: i) the instrumental variable case depicted by Figure 1.5b), where L can be used as an instrument (e.g., in the classic econometric studies with non, or partial, compliance) in lieu of A, by exploiting the fact that L does not share any unobserved confounder with Y (Angrist et al., 1996); ii) the proximal scenario of Figure 1.5c), where the coexistence of the three types of 'proxy' variables (Z, X, W) and U makes conditioning on X insufficient to achieve identification, and a 'proximal strategy' is needed (Tchetgen et al., 2020; Mastouri et al., 2021; Cui et al., 2023). Furthermore, another case of instrumental variable application is represented by the **frontdoor adjustment**,



(a) Unconfounded (b) Instrumental (c) Proximal

Figure 1.5: Example of causal DAGs where the aim is to identify the causal relationship $A \to Y$: a) (fully observable) unconfounded scenario, the focus of this work, where conditioning on X is sufficient to identify $A \to Y$; b) Instrumental scenario, where we have some unobserved confounder U that we cannot condition on, but we can use L as an "instrument" for A; c) Proximal, where the coexistence of three types of proxies $L = (Z, X, W)$ makes conditioning insufficient, and 'proximal identification' strategy is needed (Tchetgen et al., 2020).

where we assume we can access a "mediator" variable M which fully mediates the path from A and Y, but does not share unobserved confounders with Y, i.e., $A \to M \to Y$.

Given a causal DAG (and again under the local markov assumption), we can define an associated **Structural Causal Model** (SCM), which is a set of functionals describing the causal relationships (the "arrows") between the variables in the DAG. More formally, a structural causal model is defined as a 4-tuple $\langle E, \mathcal{V}, \mathcal{F}, p(\varepsilon) \rangle$ consisting of (subscript j indicates a random element in the set):

1. E: denoting a set of exogenous variables $\varepsilon_j \in E$, defined as variables determined outside of the model.
2. $\mathcal{V}$: denoting a set of endogenous variables $V_j \in \mathcal{V}$, defined as variables determined inside the model.
3. F: a set of functions $f_j \in \mathcal{F}$ mapping each element $\varepsilon_j \in E$ and every parent variables of $V_j \in \mathcal{V}$, $pa(V_j)$, to the endogenous variables $V_j \in \mathcal{V}$, $f_j : \varepsilon_j \cup pa(V_j) \mapsto V_j$.
4. $p(\varepsilon_j)$: a probability distribution over $\varepsilon_j \in E$.

Both ε_j and V_j are nodes in the causal graph, while the functional causal relationships $f_j \in \mathcal{F}$ are depicted by arrows. As an example, the SCM associated with the causal DAG in Figure 1.5a) is fully described by the following set of equations:

$$\begin{aligned} X_i &= f_X(\varepsilon_{i,X}), \\ A_i &= f_A\big(pa(A_i), \varepsilon_{i,A}\big) = f_A(X_i, \varepsilon_{i,A}) \\ Y_i &= f_Y\big(pa(Y_i), \varepsilon_{i,Y}\big) = f_Y(X_i, A_i, \varepsilon_{i,Y}) \ , \end{aligned} \tag{1.10}$$

where f_A and f_Y are potentially complex, non-linear functions, while ε_A and ε_Y are the unobservable exogenous variables representing noise. Especially f_Y describes how outcome of interest Y changes in response to changes in the manipulative variable A.

1.2.3 Dawid's Decision-Theoretic Approach

The last identification framework we briefly present is based on the seminal work of Dawid (2000, 2015). Dawid's work emphasizes the distinction between two substantially different causal problems, the **Effects of Causes** (EoC) and the **Causes of Effects** (CoE), and that, in the former, one does not necessarily needs to resort to the idea of "counterfactuals". Learning EoC is concerned with quantifying the future effects of an intervention in a causal systems where the cause-effect relationship is known. This represents a setting where the causal DAG structure is known, or partially known, (or, better, assumed) and the focus is on estimating the causal effects of known causes (for instance, it is scientifically proven that taking an aspirin can be good against migraine, but e.g. we do not know whether this is always the case in the population as some may suffer moderate to severe side effects).

On the other hand, CoE is instead concerned with learning what might have been the causes of an observed effect. In this case, the causal DAG is unknown (or only partially known), and the aim is to actually learn its causal structure and effects through observed data (randomized or observational, or

mixed). This is also known as a **Causal Discovery** task (Glymour et al., 2019) — different than **Structure Learning**, that more generally consists in learning graph structures (Drton and Maathuis, 2017). Dawid (2000, 2015) thus underline the need for separate causal frameworks for these two problems. In particular, they argue that while counterfactual statements are unavoidable in the case of CoE studies, they are not strictly necessary in EoC ones, and consequently discuss a different approach to causal effect identification under the latter.

The framework is decision-theoretic as we assume that we deal with a decision-making problem regarding e.g. what type of treatment $A = a$ to give to a patient. In the case of fully randomized studies, A is a "decision variable", not a random one, and thus does not follow a probability distribution $p(A)$. The quantity of general interest is the "hypothetical" distribution $p(Y = y \mid A = a)$. In particular, assuming $A \in \{0, 1\}$, we are interested in comparing $p(Y = y \mid A = 1)$ with $p(Y = y \mid A = 0)$, e.g. through their expected values $\mathbb{E}(Y \mid A = 1) - \mathbb{E}(Y \mid A = 0)$ (ATE), with the goal of choosing $A = a$ that minimizes some loss function $\mathcal{L}(y)$ or maximizes some reward function $\mathcal{R}(y)$. The decision-theoretic (DT) model is then simply made of a decision variable A and the conditional hypothetical distributions $p(Y = y \mid A = a)$. Notice that a specific DT model, such as the one just described, does not admit a unique representation through functional models $Y = f(A, \varepsilon, \cdot)$ but many. In the case of fully randomized experiments, two groups of exchangeable units are administered $A = 1$ and $A = 0$ respectively, and can be viewed as being drawn directly from the hypothetical conditional distributions $p(Y = y \mid A = 1)$ and $p(Y = y \mid A = 0)$ at random.

In the case of observational studies instead, A is no longer a decision variable and is now associated with a probability distribution $p(A)$. The problem here is that units under the two treatment arms can no longer be viewed as exchangeable due to confounding stemming from some covariate X, thus we seek to derive an approximation for the hypothetical conditional

distribution. To this end, we define a regime indicator variable F_A as

$$F_A = \begin{cases} 1 & \text{interventional with } A = 1 \\ 0 & \text{interventional with } A = 0 \\ \varnothing & \text{observational ,} \end{cases} \tag{1.11}$$

which simply describes under which regime one is operating. Under the no unobserved confounding, randomized scenario we have that $p(Y \mid F_A = a) = p(Y \mid A = a, F_A = a)$, while in observational scenarios we have to assume that

$$\begin{aligned} p(Y \mid F_A = a) \overset{\text{def}}{=} p(Y \mid A = a, F_A = a) \; &= \int_x p\big(Y \mid A = a, X = x, F_A = \varnothing\big) dp(x) \,, \\ &= p(Y \mid A = a, F_a = \varnothing) \end{aligned}$$

which implies the conditional independence $Y \perp\!\!\!\perp F_A \mid A$, that is the exact equivalent of unconfoundedness assumption. The main difference is that the assumption in a DT model does not rely on any notion of potential or counterfactual outcomes, but only requires the definition of the regime indicator variable F_A and confounder X. In the rest of the work, we will rely mostly on the POs and the Causal Bayesian Network (or Structural Causal Models) frameworks instead of the decision-theoretic approach, as they are the most widely adopted ones in the literature.

Chapter 2

Regression Adjustment Methods for Causal Effects Learning

Contributions *The majority of the contents in this chapter are based on a paper submitted and accepted to the Journal of the Royal Statistical Society: Part A (Statistics in Society). Reference: Caron et al. (2022a).*

2.1 Introduction

Provided that causal effects are identifiable from data, their empirical estimation can be tackled in different ways in practice (Imbens, 2004). For example, matching methods (Rubin, 1973; Stuart, 2010) are concerned with pairing each treated unit with a suitable "similar" control (or multiple ones) in terms of some definition of distance between their covariate realization $\boldsymbol{X}_i = \boldsymbol{x}^*$. Weighting methods focuses on re-balancing observations with propensity score estimates (Horvitz and Thompson, 1952; Hirano et al., 2003; Li et al., 2018), with the goal to approximate sampling generated under an experimental setting. However, throughout this second chapter, we will focus on the problem of estimating Conditional Average Treatment Effects (CATE), $\tau(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i^{(1)} - Y_i^{(0)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\big]$ in the POs framework or equivalently $\tau(\boldsymbol{x}_i) = \mathbb{E}\big[Y \mid do(A = 1), \boldsymbol{X}_i = \boldsymbol{x}_i\big] - \mathbb{E}\big[Y \mid do(A = 0), \boldsymbol{X}_i = \boldsymbol{x}_i\big]$ in Pearl's notation, through regression

adjustment (or supervised learning) methods, with a particular focus on non-parametric techniques, which allow to flexibly model complex functionals for the outcome Y compared to parametric ones that are nonetheless still widely used in applied disciplines. As described in the previous chapter, we consider observational settings where we assume we observed all direct common causes of A and Y in the set of covariates $\boldsymbol{X}$ (unconfoundedness), or at least enough proxies that capture confounding effects (proximal scenarios). While this can be a challenging assumption in some applied studies, e.g., in the socio-economic sciences, it is less so in others, such as medical/health ones.

We begin by reviewing some of the most recent, and most popular, regression adjustment methods, providing an overview of their implied assumptions and developing a novel unifying taxonomy for them. After briefly describing the challenges related to model selection in causal effects learning, we then proceed to compare models' performance and demonstrate their empirical finite-sample properties through a number of simulated experiments. Finally, we illustrate a practical real-world application of some of the methods by analyzing the NHANES data (Chan et al., 2016), with the aim of investigating the presence of heterogeneous effects of school meal participation on children's BMI, and detecting the moderators responsible for heterogeneity.

2.2 Regression-Based Setup

The (non-parametric) regression approaches we will review generally model the outcome surface Y_i as a function of the covariates-action pairs ($\boldsymbol{X}_i$, A_i) and some unobservable error term ε_i : $Y_i = g(\boldsymbol{X}_i, A_i, \varepsilon_i)$. More specifically, the reviewed methods typically restrict the functional space capacity/complexity in the corresponding structural causal model of the outcome by assuming that the error term ε_i is additive and with mean zero, which leads to the following setup:

$$Y_i = f(\boldsymbol{X}_i, A_i) + \varepsilon_i \ , \qquad \text{where} \quad \mathbb{E}(\varepsilon_i) = 0 \ , \tag{2.1}$$

and $f(\boldsymbol{X}_i, A_i) = \mathbb{E}\big[Y_i \mid \boldsymbol{X}_i, A_i\big]$ is left unspecified and learnt from the data. The strength of non-parametric regression models is that they are less prone to misspecification of the functional form of $f(\cdot)$ (e.g., tree-based methods model $f(\cdot)$ as piece-wise constant, splines as piece-wise polynomial, etc.). As mentioned earlier in this work, the covariates $\boldsymbol{X}_i \in \mathcal{X}$ represent a potential source of confounding to be controlled for. In terms of model-building, this means that ideally confounding variables contained in $\boldsymbol{X}_i$ need to be included in the outcome model — and in the propensity score model, if employed — while different subsets of other non-confounding covariates might be included in the propensity and outcome models if they are relevant predictors of either A_i or Y_i, in order to increase precision of the estimates. In addition, in a setting where the number of available covariates is high, one might want to resort to some sort of shrinkage or regularization when estimating of $f(\cdot)$. However, as explained in both Hahn et al. (2018) and Hahn et al. (2020), regularization should be handled carefully in this context, as we will explain further in later sections.

In the regression setup illustrated in (2.1), some of the frameworks reviewed in the next section reserve a specific role for the propensity score (X-, R- and τ-Learners) — these are often referred to as "propensity methods". As for the remaining methods, which do not explicitly envisage any use of the propensity score, in the simulated studies that we conduct in later sections, we follow the suggestion of Hahn et al. (2020) and incorporate PS estimates as an additional covariate, according to the following two-stage regression framework:

$$\begin{aligned} \pi(\boldsymbol{X}_i) &= \mathbb{P}\big(A_i = 1 \mid \boldsymbol{X}_i\big) \\ Y_i &= f\Big(\big[\boldsymbol{X}_i \;\; \pi(\boldsymbol{X}_i)\big], A_i\Big) + \varepsilon_i \; . \end{aligned} \tag{2.2}$$

The first stage in (2.2) involves estimating the propensity score, while the second embeds it as an extra covariate in the covariate set (Heckman, 1979). Any probabilistic classifier is suitable for use in the first stage regression (e.g. logistic regression, neural networks, etc.). As explained in Rosenbaum and Rubin

(1983); Hahn et al. (2020), and as we will describe in later sections, the addition of $\pi(\boldsymbol{X}_i)$ to the covariate set in (2.2) represents an effective way to tackle bias deriving from *targeted selection*. Targeted selection is a type of selection bias that arises when individuals are selected into treatment based on the prediction of otherwise adverse outcome (or of large gains under treatment), i.e., when $\pi(\boldsymbol{X}_i)$ is a strictly monotone function of $\mathbb{E}[Y_i^{(0)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i]$, and is common in many observational studies (e.g. medical or socio-economic studies).

It is worth mentioning here that particular care should be taken when building a propensity score model, especially in presence of high-dimensional covariate spaces $\boldsymbol{X} \in \mathcal{X}$. Indeed, the use of conventional prediction-driven 'automated' variable selection methods in high-dimensional settings may not result into an optimal propensity model in terms of causal effect estimation. For instance, Brookhart et al. (2006) argue that including variables related to the outcome, but not the treatment, should always be included in the PS model, as these are shown to increase precision of causal effects estimates, without increasing bias, in small samples. Furthermore, Pearl (2011) shows that accidentally selecting in instrumental variables can be detrimental to both bias and precision, but also the inclusion of near-Instrumental Variables (IV), which are variables that act both as confounders and as instruments, in terms of bias-variance trade-off.

Now, estimators for CATE $\tau(\boldsymbol{x}_i) \in \mathcal{T}$ (where by $\mathcal{T}$ we denote CATE's own functional space) can be directly derived from the representations in (2.1) and (2.2). There are currently several different approaches for deriving an estimator for CATE from (2.1) and (2.2), that will be analyzed in the next section.

2.3 CATE Estimators

Given the framework outlined above, various meta-algorithms designed to derive a CATE estimator have been proposed in the literature. These meta-algorithms are often referred to as "Meta-Learners", in that they are subroutines of "base-learners", which are common supervised learning algorithms (e.g. tree

Table 2.1: Summary of meta-learners discussed in this work.

	References	**CATE estimator**
S-Learner	Hill (2011); Foster et al. (2011)	$\tau(\boldsymbol{x}_i) = f(\boldsymbol{x}_i, 1) - f(\boldsymbol{x}_i, 0)$
T-Learner	Athey and Imbens (2016); Lu et al. (2018), Powers et al. (2018)	$\tau(\boldsymbol{x}_i) = f_1(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)$
X-Learner	Künzel et al. (2017)	$\tau(\boldsymbol{x}_i) = \pi(\boldsymbol{x}_i)\tau_0(\boldsymbol{x}_i) + \big(1 - \pi(\boldsymbol{x}_i)\big)\tau_1(\boldsymbol{x}_i)$
R-Learner	Nie and Wager (2020)	$\tau(\boldsymbol{x}_i) = \arg\min_\tau \Big\{ L_n\big(\tau(\cdot)\big) + \Lambda_n\big(\tau(\cdot)\big) \Big\}$
Multitask-Learner	Alaa and van der Schaar (2017, 2018)	$\tau(\boldsymbol{x}_i) = \mathbf{f}^\top(\boldsymbol{x}_i)\mathbf{e}$
τ-Learner	Hahn et al. (2020)	$\tau(\boldsymbol{x}_i)$ as explicit model parameter

ensembles, neural networks, gradient boosting methods, etc.). In what follows, we attempt to build a unifying taxonomy of these "Meta-Learners" approaches in Section 2.3.1, while in Section 2.3.2 we present an overview on the problem of model selection for CATE estimation, which is a substantially hard, arguably impossible, problem.

2.3.1 Meta-Learners

As mentioned in the earlier sections, we will partly build on top of the work by Künzel et al. (2017) and expand it by including the most recent contributions stemming from both the statistics and computer science literature. A concise summary of the presented "Meta-Learners", together with the relevant references, can be found in Table 2.1.

2.3.1.1 S-Learners

"Single-Learners", shortened to S-Learners, have been implicitly proposed, among others, in the two early contributions of Hill (2011) and Foster et al. (2011), and derive an estimator for CATE by including treatment assignment as "just another covariate" in the covariate set $\boldsymbol{X}_i \in \mathcal{X}$, which means that CATE

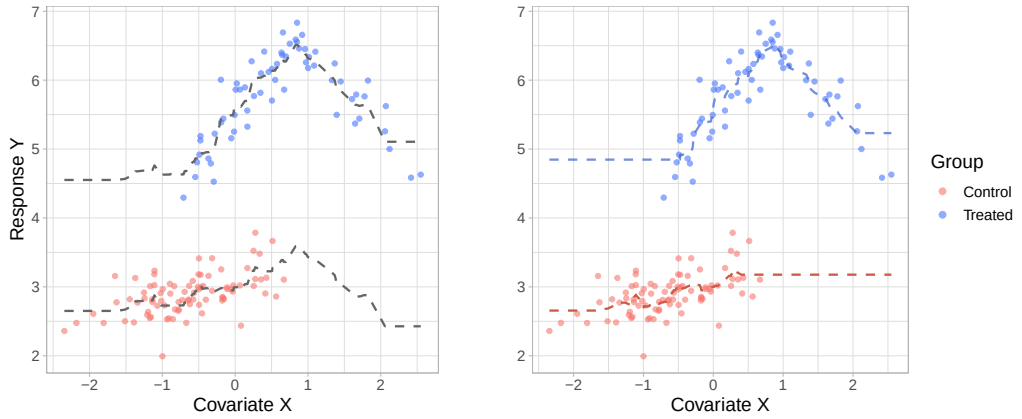


Figure 2.1: Simulated one-covariate data from Section 1.2. Left panel: conditional mean fit from a S-Learner BART (dashed grey line). Right panel: conditional mean fit from a T-Learner BART (blue and red dashed lines).

estimator take the form

$$\tau(\boldsymbol{x}_i) = f\big([\boldsymbol{x}_i \;\; 1]\big) - f\big([\boldsymbol{x}_i \;\; 0]\big) \; . \tag{2.3}$$

An S-Learner fits a single surface $f(\cdot)$, employing the regressors $[\boldsymbol{X}_i \;\; A_i]$, through a base-learner and derives CATE estimates by taking the difference between the two conditional average potential outcomes, which are represented by the fitted $\widehat{f}(\cdot)$ with $A_i = 1$ and $A_i = 0$ respectively. The underlying assumption is that the group-specific conditional average potential outcomes stem from the same probabilistic model, whose conditional mean function is $f(\cdot)$ and error term is ε_i. Regression trees are popular base-learners employed in the context of S-Learners. For instance, Hill (2011) advocates the use of Bayesian Additive Regression Trees (BART), while Foster et al. (2011) of random forests. The left panel plot of Figure 2.1 shows a S-Learner BART fit for the conditional mean $\widehat{f}(\cdot)$ of the single-covariate simulated example already encountered in Figure 1.2. Notice that the dashed line representing $\widehat{f}(\cdot)$ has a unique color (grey) to emphasize the fact that S-Learner fits a unique surface.

Since an S-Learner fits a single regression, it is quite restrictive in the way it accounts for the variation in Y attributable to A; this is because it

does not accurately target the causal functional of interest and thus performs sub-optimal regularization. This issue regarding targeted regularization will be further investigated in later sections and in the next chapter, particularly in the experiments of Section 3.2.4. This is relatively more of an issue when working with observational data, while it is less so with randomized experiments where selection bias is not a threat. Alaa and van der Schaar (2018) and Hahn et al. (2020) have both identified that the main drawback of S-Learners is their lack of ability in adapting to different levels of sparsity and smoothness across the treatment arms, since they impose the same regularizing conditions for both treated and control groups. A S-Learner will then perform poorly in a situation where the outcome surface complexity is very different across the two groups. On the contrary, S-Learner is expected to perform well when CATE is of simple form, as the complexity of the conditional average potential outcomes surfaces $\mu_1(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i^{(1)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\big]$ and $\mu_0(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i^{(0)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\big]$ does not vary much across treatment groups. For example, consider the case of a S-Learner employing a tree ensemble base-learner, such as BART. Since a tree ensemble method like BART picks splitting variables at each node in each tree randomly, it might not even choose A as splitting variable in some of the trees in the ensemble, so that A will possibly be included in most of the trees fitting the response Y, but not necessarily in all of them. The exclusion of A from the splitting rules of a tree in BART is more likely to happen as the number of covariates $\boldsymbol{X}_i$ grows larger, in that the model has a larger set of splitting variables to pick from (Caron et al., 2022b). This intuitively explains why S-Learners turn out to be appropriate in situations where the complexity of ground-truth CATE functional is reasonably low, relative to the variation in outcome attributable to the covariates only ($\mathbb{E}\big[Y_i \mid \boldsymbol{X}_i\big]$). It may happen in real world applications, such as clinical studies, that the researcher possesses some domain knowledge regarding the complexity of the treatment effect functional $\tau(\boldsymbol{x}_i) \in \mathcal{T}$. As we will explain in the following sections, this is a useful piece of information when it comes to choose a suitable method and to

reduce the capacity of the CATE functional space $\mathcal{T}$ through incorporation of prior knowledge.

2.3.1.2 T-Learners

"Two-Learners", shortened to T-Learners, derive an estimator for CATE by fitting two separate surfaces for the treated and control groups and computing their difference:

$$\tau(\boldsymbol{x}_i) = f_1(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i) \ . \tag{2.4}$$

Versions of T-Learners can be found in many contributions in the literature. For instance, Athey and Imbens (2016), Lu et al. (2018) and Powers et al. (2018) offer some examples employing decision trees, random forests and gradient boosted trees as base-learners respectively. In contrast to S-Learners, T-Learners separate the two treatment groups when modelling response variable Y, and assume that group-specific conditional average potential outcomes are derived from separate probabilistic models, characterized by different conditional mean functionals $f_1(\cdot) \in \mathcal{F}_1$ and $f_0(\cdot) \in \mathcal{F}_0$, and independent error terms ε_{1i} and ε_{0i}. This allows to preserve distributional differences across the two groups that might originate from selection bias, and to take into account different degrees of sparsity and smoothness that vary with A, when regressing Y against X. On the other hand, a shortcoming of T-Learners is that, as a result of splitting the sample in two, they do not allow sharing common underlying information between the groups when estimating the two surfaces. This is particularly sub-optimal in a scenario where units in the two groups share the same distributional characteristics in terms of the conditional distribution $p(y|x)$, which is regardless of the administered treatment A_i, and when the treatment arms are particularly unbalanced, which might lead to dangerous under/over-fitting.

The right panel plot in Figure 2.1 displays a T-Learner BART fit for the conditional mean functions of the two treatment groups $f_1(\cdot) \in \mathcal{F}_1$ and $f_0(\cdot) \in \mathcal{F}_0$, on the same one-covariate simulated example of Figure 1.2. Notice

that fitted $\widehat{f_1}(\cdot)$ and $\widehat{f_0}(\cdot)$ are differentiated by colors (blue and red dashed lines respectively), emphasizing the fact that T-Learner fits two separate regression models with independent error terms.

T-Learners are expected to work particularly well when complexity of the response surface is very different across treatment groups, i.e., depending on A_i, and so CATE itself turns out to be a rather complex function. Ideally, they preserve good asymptotic properties if sampling is balanced across groups (Alaa and van der Schaar, 2018), i.e., if larger sample size implies both groups are larger.

However, this is not often the case in many real world designs, where one of the arms is typically smaller. In the presence of such highly imbalanced designs, splitting the sample into subgroups inevitably leave too few observations for the estimation of $f_A \in \mathcal{F}_a$ in the smaller group. In the following subsections we will see how this issue is addressed by other Meta-Learners that extend the T-Learner framework (X-Learners and Multitask-Learners). On the contrary, T-Learners tend to perform quite poorly in settings where CATE function is relatively simple and heterogeneity patterns are not so sophisticated, i.e., situations where S-Learner usually performs better. Hence if subject-matter prior knowledge, to restrict the CATE functional space capacity $\mathcal{T}$, suggests that treatment impact is likely to be significantly complex, a T-Learner might be the preferred choice (i.e., in a Bayesian sense, higher probability mass is assigned in the choice of a suitable functional - we will see this more in details in the model selection section).

2.3.1.3 X-Learners

X-Learners have been proposed by Künzel et al. (2017) as an extension of T-Learners, and derive a CATE estimator in three steps. In the first step, conditional average potential outcomes are fitted as in a T-Learner approach, that is by using two separate regression models for the conditional means $f_1(\boldsymbol{x}_i) \in \mathcal{F}_1$ and $f_0(\boldsymbol{x}_i) \in \mathcal{F}_0$, assuming independent error structures. Then in the second step, "imputed treatment effects" are computed for each group

separately; these are defined as the differences between the group-specific observed outcome Y_i^A, and the estimated unobservable conditional average potential outcome $\widehat{Y}_i^{(A)}$ derived in the first step, more formally:

$$\begin{aligned} \tilde{D}_i^1 &= Y_i^1 - \widehat{Y}_i^{(0)} \qquad \text{if} \quad A_i = 1 \\ \tilde{D}_i^0 &= \widehat{Y}_i^{(1)} - Y_i^0 \qquad \text{if} \quad A_i = 0 \ . \end{aligned} \tag{2.5}$$

The second step thus attempts to recover the unobservable differences $D_i = Y_i^{(1)} - Y_i^{(0)}$ (ITE) separately for the treated and control group by replacing the unobservable counterfactual outcomes with the relative conditional average potential outcome estimates $\widehat{Y}_i^{(1-A)}$, but using the observed outcome for the other "actual" outcome $Y_i^A = Y_i^{(A)}$, whereas a T-Learner would just use fitted values for both instead. In the last step, $\tilde{D}_i^1$ and $\tilde{D}_i^0$ are utilized as response variables in two separate regressions, employing the chosen base-learner (linear regression, random forest, BART, etc.), to obtain estimates of $\hat{\tau}_1(\boldsymbol{x}_i) \in \mathcal{T}_1$ and $\hat{\tau}_0(\boldsymbol{x}_i) \in \mathcal{T}_0$, using covariates $\boldsymbol{X}_i$ as regressors. These two quantities are group-specific CATE estimates. The two independent regressions then take the following form

$$\begin{aligned} \tilde{D}_i^1 &= \tau_1(\boldsymbol{X}_i) + \eta_{1i} \qquad \text{if} \quad A_i = 1 \\ \tilde{D}_i^0 &= \tau_0(\boldsymbol{X}_i) + \eta_{0i} \qquad \text{if} \quad A_i = 0 \ , \end{aligned} \tag{2.6}$$

where $\mathbb{E}[\eta_{a,i}] = 0$ and $\text{cov}(\eta_{1,i}, \eta_{0,i}) = 0$. The final CATE estimate is then obtained through a weighted average of the two group-specific CATE estimates,

$$\widehat{\tau}(\boldsymbol{x}_i) = g(\boldsymbol{x}_i)\widehat{\tau}_0(\boldsymbol{x}_i) + \big(1 - g(\boldsymbol{x}_i)\big)\widehat{\tau}_1(\boldsymbol{x}_i) \ , \tag{2.7}$$

where $g(\boldsymbol{x}_i) \in [0, 1]$ is a given weighting function. The authors propose to set $g(\cdot)$ equal to a propensity score estimate $g(\boldsymbol{x}_i) = \widehat{\pi}(\boldsymbol{x}_i)$, but this can also take other forms (e.g. $g(\boldsymbol{x}_i) = 1$ or $g(\boldsymbol{x}_i) = 0$ if one of the groups is very unbalanced).

The intuition behind the last weighting step, that particularly characterizes X-Learners, is the following. When we are in presence of an unbalanced

design, with potential issues of poor overlap, and we fit a T-Learner, we would ideally want to pick a more complex (possibly non-parametric) model for the large treatment group and a simpler one for the small treatment group, to avoid over-fitting and allow better within-group generalization. While the model for the larger group is likely to be well-specified, since we observe a lot of data points, the model for the smaller group might not be very representative of the true conditional average potential outcome function, $\mu_1(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i^{(1)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\big]$ or $\mu_0(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i^{(0)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i\big]$, as we only observe a handful of data points. Nonetheless, the simpler model, which is then also employed to obtain group-specific CATE estimates of the smaller group $\hat{\tau}_A(\boldsymbol{x}_i)$, might be highly representative of the CATE function instead, so that $\hat{\tau}_A(\boldsymbol{x}_i)$ is actually very close to the true $\tau(\boldsymbol{x}_i) = \mu_1(\boldsymbol{x}_i) - \mu_0(\boldsymbol{x}_i)$.

The choice of $g(\boldsymbol{x}_i) = \widehat{\pi}(\boldsymbol{x}_i)$ allows to properly assign higher weight to the simpler model's CATE estimates, since e.g. if the treated group is small, then $\widehat{\pi}(\boldsymbol{x}_i)$ will generally be skewed to the right, and the final CATE estimates $\hat{\tau}(\boldsymbol{x}_i)$ will thus be close to $\hat{\tau}_1(\boldsymbol{x}_i)$. Choosing $g(\boldsymbol{x}_i) \in \{0, 1\}$ is useful in scenarios where the groups are very unbalanced, where more extreme values are necessary to "nudge" the final CATE estimates $\hat{\tau}(\boldsymbol{x}_i)$ towards the smaller group's estimates $\hat{\tau}_A(\boldsymbol{x}_i)$.

For this reason, in unbalanced studies (with poor overlap) where T-Learners would yield unnecessarily complex estimates of CATE, X-Learners attempt to improve accuracy by re-balancing group-specific CATE estimates through propensity score weighting. In this way, they avoid overfitting and revert back to simpler CATE patterns. A final remark about X-Learners, which is naturally valid for all propensity methods, is that careful specification of the propensity model is required to effectively improve precision in CATE estimates through the last balancing step. Poor propensity estimates might not deliver the desired results.

Figure 2.2 offers a simple example of a X-Learner, with BART as base-learner, applied to the one-covariate simulated data encountered in Figure 1.2

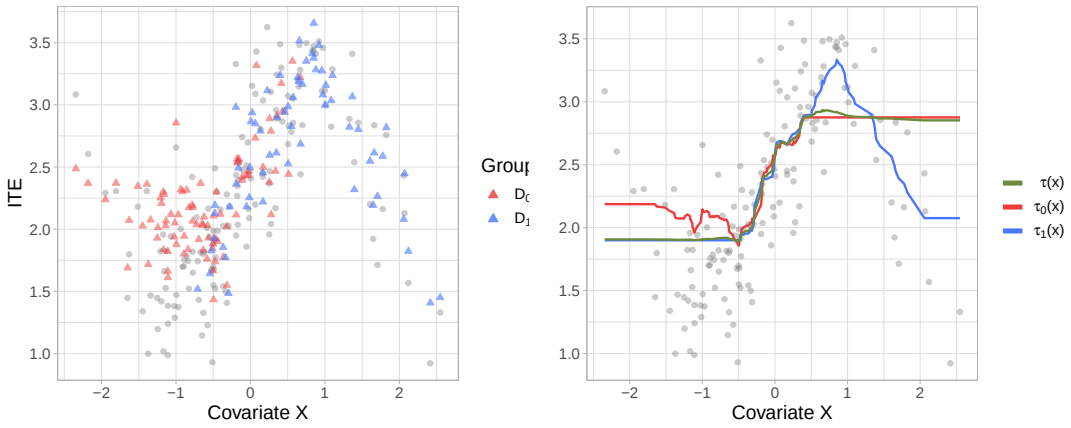


Figure 2.2: X-Learner BART applied to the simulated one-covariate example. Left panel: unobservable ITE (grey dots) and imputed treatment effects D^1 and D^0 (blue and red triangles), estimated as in (2.5) using T-Learner BART. Right panel: group-specific CATE estimates (blue and red dashed lines) obtained from the two regressions in (2.6), and final weighted CATE estimates (green dashed line) obtained from the re-balancing step in (2.7).

and Figure 2.1. X-Learner's first step essentially derives, via T-Learner, the same BART estimates seen in the right panel plot of Figure 2.1. The output of the second step, namely the imputed treatment effects $\tilde{D}_i^1$ and $\tilde{D}_i^0$, are depicted in the left panel plot of Figure 2.2 (red and blue triangles), together with the true ITE (grey dots). The graph on the right instead shows the estimated group-specific CATE $\hat{\tau}_1(\boldsymbol{x}_i)$ (blue dashed line) and $\hat{\tau}_0(\boldsymbol{x}_i)$ (red dashed line), derived from the two regressions in (2.6), and the final CATE estimate $\widehat{\tau}(\boldsymbol{x}_i)$ (green dashed line), obtained from the weighting step in (2.7). Propensity score estimates employed for the weighting step were retrieved via probit version of BART. Notice that the final CATE estimates $\widehat{\tau}(\boldsymbol{x}_i)$ lie in between the two fitted group-specific $\hat{\tau}_1(\boldsymbol{x}_i)$ and $\hat{\tau}_0(\boldsymbol{x}_i)$.

2.3.1.4 R-Learners

R-Learner was originally proposed by Nie and Wager (2020) as a two-stage meta-algorithm, and aims at minimizing a loss function specifically defined on CATE through parameter tuning. The derivation of the two-step procedure

stems from Robinson (1988) decomposition of the outcome model in (2.1). We start by defining the following two quantities:

$$\begin{aligned} Y_i &= \mu(\boldsymbol{X}_i) + \tau(\boldsymbol{X}_i)A_i + \varepsilon_i \\ m(\boldsymbol{X}_i) &= \mathbb{E}(Y_i \mid \boldsymbol{X}_i) = \mu(\boldsymbol{X}_i) + \tau(\boldsymbol{X}_i)\pi(\boldsymbol{X}_i) \end{aligned} \tag{2.8}$$

as the outcome model and the *conditional mean outcome* model, respectively. Under this setup, unconfoudedness assumption implies that the error term is such that $\mathbb{E}\big[\varepsilon_i \mid \boldsymbol{X}_i, A_i\big] = 0$. Notice that under this semi-parametric additive parametrization $\tau(\boldsymbol{X}_i)$ (CATE) enters explicitly in the outcome regression model. By combining the two quantities above, Robinson (1988) noticed that the outcome model can be parametrized as:

$$Y_i - m(\boldsymbol{X}_i) = \Big(A_i - \pi(\boldsymbol{X}_i)\Big)\tau(\boldsymbol{X}_i) + \varepsilon_i \ . \tag{2.9}$$

Starting from this decomposition, Nie and Wager (2020) derive a loss function that can be used for parameter tuning in the estimation of CATE; the optimal CATE estimates are defined as the minimizer of the following loss function:

$$\tau(\boldsymbol{X}_i) = \arg\min_{\tau} \left\{ \mathbb{E}\left[\Big(\big(Y_i - m(\boldsymbol{X}_i)\big) - \big(A_i - \pi(\boldsymbol{X}_i)\big)\tau(\boldsymbol{X}_i) \Big)^2 \right] \right\} . \tag{2.10}$$

The intuition behind equation (2.10) is the following. Suppose that an individual i is characterized by an extreme propensity value $\pi(\boldsymbol{x}_i) \approx 0$; thus its realized treatment assignment is almost deterministically $A_i = 0$. In this extreme scenario, eq. (2.10) boils down to simple Mean Squared Error (MSE) minimization for the conditional mean outcome model $m(\boldsymbol{x}_i)$, as in a standard regression problem. Hence, the term $\big(A_i - \pi(\boldsymbol{X}_i)\big)\tau(\boldsymbol{X}_i)$ subtracted serves as a de-biasing term that grows larger with the discrepancy between the realized treatment assignment A_i and the propensity score $\pi(\boldsymbol{x}_i)$, and is supposed to tackle selection into treatment imbalance through propensity score re-weighting.

The idea is that a base-learner that relies on parameter tuning (e.g. random

forest or gradient boosted trees) can be tuned on the modified parametrization of the outcome model in (2.9), which includes a version of the outcome net of the baseline impact of the covariates $\boldsymbol{X}_i$ on Y_i, $m(\boldsymbol{X}_i)$, and propensity score balancing, instead of being tuned on the raw outcome Y_i as one would do in an S- or T-Learner framework. Since we cannot observe directly the quantities in (2.10) for the minimization problem, the R-Learner replaces them with cross-validated estimates, through the following two-step approach:

1. Split the data into k-folds (5 or 10 suggested). Fit nuisance functions $\widehat{m}(\boldsymbol{x}_i)$ and $\widehat{\pi}(\boldsymbol{x}_i)$ (on a portion of left-out data) by minimizing usual prediction errors via cross-validation

2. Plug in estimates from the first step to estimate $\hat{\tau}(\boldsymbol{x}_i)$, by minimizing the regularized sample equivalent of (2.10) via parameters tuning on the k-folds, that is:

$$\widehat{\tau}(\boldsymbol{X}_i) = \arg\min_{\tau} \left\{ \widehat{L}_n\big(\widehat{\tau}(\boldsymbol{X}_i)\big) + \Lambda_n\big(\widehat{\tau}(\boldsymbol{X}_i)\big) \right\}, \quad \text{where} \tag{2.11}$$
$$\widehat{L}_n\big(\widehat{\tau}(\boldsymbol{X}_i)\big) = \frac{1}{n}\sum_{i=1}^{n} \left(\Big(Y_i - \widehat{m}^{-i}(\boldsymbol{X}_i)\Big) - \Big(A_i - \widehat{\pi}^{-i}(\boldsymbol{X}_i)\Big)\widehat{\tau}(\boldsymbol{X}_i) \right)^2,$$

and where $\Lambda_n\big(\hat{\tau}(\cdot)\big)$ is a term representing regularization (e.g. L1 or L2 regularization, splines smoothness penalization, dropout, etc.). The super-script $(-i)$ refers to the i-th observation being held-out from the estimation subsample, and used for k-fold cross-validation (or even more computationally intense leave-one out cross-validation).

The R-Learner setup resembles, and is in fact inspired by, the doubly robust approach to the estimation of average treatment effects (Cassel et al., 1976; Bang and Robins, 2005; Dudík et al., 2011; Dudík et al., 2014; Chernozhukov et al., 2018; Athey and Wager, 2021), with the main difference that the second stage in the R-Learner is specifically designed for heterogeneous/conditional treatment effects estimation with potentially non-linear models (Athey and Wager, 2019; Nie and Wager, 2020). The strength of R-Learners lies in their

two-stage procedure, where the first stage takes care of predicting the nuisance functions $\widehat{m}(\boldsymbol{x}_i)$ and $\widehat{\pi}(\boldsymbol{x}_i)$, while the second focuses on CATE estimation by constructing a direct loss function on it. In this way, R-Learners ensure that regularization is targeted and implemented separately for the nuisance functions and for CATE, and lowers the risk of over/under fitting. This is intuitively a desirable feature when CATE is of different complexity (in most cases smoother and sparser) compared to the nuisance functions, in that more (or less) aggressive regularization can be conveyed when estimating it, compared to that conveyed in estimating the baseline marginal effect of the covariates on the outcome, $m(\boldsymbol{x}_i)$. This particular feature is also shared by τ-Learners, introduced in the next subsection, where direct regularization on CATE is instead applied in the form of Bayesian shrinkage priors (Hahn et al., 2020; Caron et al., 2022b,c). The loss minimization procedure described by the R-Learner framework can generally involve any supervised learning method. The original work of Wager and Athey (2018) and Nie and Wager (2020) focuses particularly on the use of parametric LASSO linear regression, random forests and gradient boosted trees, whose parameters are tuned to minimize the R-loss in (2.11).

A final remark about R-Learners concerns the popular Causal Forest model for CATE estimation (Wager and Athey, 2018; Athey and Wager, 2019), which we are going to include in our empirical comparison of methods based on simulated studies. As stated in Athey and Wager (2019), Causal Forest can be viewed as a regression forest method motivated by the R-Learner setup. Indeed, its latest implementation uses separate regression forests to fit the nuisance functions and then trains another forests on the CATE loss function in (2.11).

2.3.1.5 Multitask-Learners

The idea of multitask-learning, or multi-output learning, for causal inference was explicitly introduced by Alaa and van der Schaar (2017) and Alaa and van der Schaar (2018), in the context of Gaussian Processes. The multitask perspective on CATE estimation consists in viewing the two potential outcomes $Y_i^{(0)}$ and

$Y_i^{(1)}$ as output of a vector-valued function $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^2$, with d-dimensional input space and 2-dimensional output space, where the output space is indexed by A_i, that acts as "task identifier". CATE estimator is defined as the difference between the elements of the 2-dimensional output of $\boldsymbol{f}(\cdot)$, i.e.,

$$\widehat{\tau}(\boldsymbol{x}_i) = \widehat{f}_1(\boldsymbol{x}_i) - \widehat{f}_0(\boldsymbol{x}_i) = \hat{\mathbf{f}}^\top(\boldsymbol{x}_i)\boldsymbol{\xi} \ , \quad \text{where} \quad \boldsymbol{\xi}^\top = [-1 \ \ 1] \ . \tag{2.12}$$

Equation (2.12) displays a very similar formulation to a T-Learner; and as in the T-Learner procedure, the sample is practically split into the two subgroups for the estimation. However, the advantage of viewing CATE estimation as a multitask problem is that, instead of estimating the two potential outcomes independently, as originating from separate conditional mean function spaces $\mathcal{F}_1, \mathcal{F}_0$ and distributions, as one would do in a T-Learner or X-Learner, they are estimated "jointly", through the specification of hyperparameters that trigger a joint loss minimization problem for the two "tasks": learning $f_{A=1}$ and $f_{A=0}$. Hence, while this approach separates the groups in subsamples (as in a T-Learner), at the same time attempts to recover common underlying patterns between them (as an S- or R-Learner would do) that would be otherwise lost due to the sample splitting. A side advantage of multioutput learning is related to the fact that joint estimation is convenient in cases where a treatment arm features a substantially smaller number of units, in that the process of borrowing information from the larger group becomes beneficial in fitting the conditional average potential outcome of the other group — this is potentially a major issue in presence of multiple treatment arms, and we will discuss it in more details in Chapter 4.

In the case of Alaa and van der Schaar (2017), multitask learning is induced through the specification of a particular structure in the kernel function of a Gaussian Process regression prior. This specific type of Gaussian Process prior kernel is often known as "coregionalization" kernel, and it is designed to induce correlation in the estimation of vector-valued functions $\boldsymbol{f}(\cdot)$ that map to multiple outcomes; this forces the underlying functions constituting

$\boldsymbol{f}(\cdot) = [f_0 \;\; f_1]$ to share similar patterns. We refer to Chapter 4 in this work, and furthermore Alvarez et al. (2012), for a more detailed discussion on multioutput learning in Gaussian Processes. The method is labelled as *Causal Multitask Gaussian Process* (CMGP). Alaa and van der Schaar (2018) then proposed a similar method where multitask learning is induced via a non-stationary version of the Gaussian Process kernel function (*Non-Stationary Gaussian Process* - NSGP).

Alaa and van der Schaar (2017) and Alaa and van der Schaar (2018) are two example of Multitask-Learners employing Gaussian Process regression as a base-learner, but there are different ways of inducing multitask learning using other types of base-learners (e.g. linear regression, tree-based methods, deep learning, etc.). In particular, a popular recent contribution that implicitly falls into the Multitask-Learners category is represented by Johansson et al. (2016) and Shalit et al. (2017) implementation of representation learning for CATE estimation (the methods are known with the name of Balancing Neural Network and/or Counterfactual Regression). The idea behind this method is to specify a deep learning model that learns "balancing" representations of the covariates by simultaneously minimizing a distance metric between the two distributions of the group-specific latent representations and a loss function on the fitted conditional average potential outcomes $f_A(\cdot)$. The goal of this "double-loss" deep neural network structure is to produce counterfactual outputs that generate from an approximation of a randomized study (expressed by the balancing representations learnt by minimizing the discrepancy loss). This model can be easily viewed as a form of Multitask-Learner, since the parameters in the deep learning model are shared across the A tasks ("hard parameter-sharing").

Due to their similarity with T-Learners in deriving a CATE estimator, we expect Multitask-Learners to perform better when complexity of the response surfaces f_1, f_0 varies across groups, and CATE itself turns out to be a rather complex function. Nonetheless, Multitask-Learners prevent from over-fitting

simpler CATE functions as T-Learners via borrowing of information across groups/tasks; and in this way (similarly to X-Learners) they also address the issue of unbalanced groups, and potentially lack of common support, in a Bayesian fashion.

2.3.1.6 τ-Learners

The last type of Meta-Learner reviewed in this chapter was implicitly developed by Hahn et al. (2020), under the name of "Bayesian Causal Forest". In a similar fashion to the R-Learner framework, the authors exploit the Robinson (1988) parametrization, but address the problem in a Bayesian way. Particularly, they noticed that the parametrization

$$Y_i = \mu(\boldsymbol{X}_i) + \tau(\boldsymbol{X}_i)A_i + \varepsilon_i \ , \tag{2.13}$$

can be viewed as a Bayesian regression framework where the *prognostic score*, defined as the impact of the covariates $\boldsymbol{X}_i \in \mathcal{X}$ on the outcome Y_i in absence of the treatment, $\mu(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i \mid A_i = 0, \boldsymbol{X}_i = \boldsymbol{x}_i\big]$, plays the role of the intercept, while $\tau(\boldsymbol{x}_i)$ the role of the slope. In this perspective, CATE can be regarded as an explicit "parameter" of the model (in the homogeneous treatment effects extreme case, this would be a scalar parameter associated with ATE) and thus can be treated in a Bayesian fashion through the specification of a (non-parametric) prior distribution $p\big(\tau(\cdot)\big)$, which restrict CATE functional space $\mathcal{T}$ capacity and can be shaped to convey prior knowledge, as well as more targeted regularization that can capture even simple patterns of heterogeneity (Caron et al., 2022b). Bayesian Causal Forest of Hahn et al. (2020) is composed by a pair of separate and independent BART priors placed on $\mu(\cdot)$ and $\tau(\cdot)$ respectively, but the parametrization in (2.13) can be exploited using other Bayesian regression methods (e.g. Gaussian Process, Dirichlet Process regression, etc.).

In addition to the parametrization shown in (2.13), Hahn et al. (2020) make use of the two-stage procedure seen in (2.2). The two-stage approach

is motivated by the presence of a particular type of confounding, which the authors in Hahn et al. (2018) and Hahn et al. (2020) call *Regularization Induced Confounding* (RIC). The intuition behind RIC is the following: regularization applied directly on the two curves $f_1, f_0 \in \mathcal{F}_1, \mathcal{F}_0$ featuring in a T-Learner (eq. (2.4)) may have unintended consequences on the induced regularization on $\tau(\cdot) \in \mathcal{T}$, leading to bias in the estimates of CATE (stemming from over-fitting). RIC is shown to have a stronger effect when there is strong confounding, such as in presence of *targeted selection*, that is when individuals are selected into treatment based on the prediction of otherwise adverse outcome. Targeted selection implies a potential strictly monotone relationship between the propensity score $\pi(\boldsymbol{x}_i)$ and the prognostic score $\mu(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i \mid \boldsymbol{X}_i, A_i = 0\big]$, and is rather common in studies of observational nature. The proposed way to tackle confounding from targeted selection is precisely to use the two-stage representation illustrated in (2.2), where a probabilistic estimate of the propensity score $\widehat{\pi}(\boldsymbol{x}_i)$, obtained from the first stage regression, is added to the covariates for the estimation of $\mu(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i \mid \boldsymbol{X}_i, A_i = 0\big]$ in the second stage, to account for their potential relationship.

We name the above approach τ-Learner, as it involves an explicit parametrization in terms of $\tau(\boldsymbol{x}_i)$ (similar to R-Learners) and a direct Bayesian approach to CATE estimation. Hahn et al. (2020) specifically make use of BART for estimation of $\mu(\boldsymbol{x}_i)$ and $\tau(\boldsymbol{x}_i)$, but any other Bayesian method could potentially work as mentioned before.

As a further advantage, the direct Bayesian approach returns full predictive posterior distribution on CATE, which conveniently allows the computation of point estimates as well as credible intervals with nice uncertainty coverage properties. This feature is shared also by Bayesian implementation of S-Learners (Hill, 2011) and can be usefully employed to check for causal common support, as showed by Hill and Su (2013). Meta-Learners that explicitly model CATE, such as S- and R-Learners, can naturally provide confidence intervals to accompany point estimates (Athey and Imbens, 2016; Athey and Wager,

2019; Caron et al., 2022a,b), while T-Learners and their extensions (X- and Multitask-Learners), which indirectly model CATE as the difference between two separately fitted surfaces, must resort to re-sampling techniques such as jackknife or bootstrapping to produce confidence intervals (Künzel et al., 2017).

2.3.2 Model Selection

Model selection is a challenging problem in causal inference, the main reason being that one cannot observe counterfactual outcomes $Y_i^{(1-A_i)}$ and thus the ITE difference $Y_i^{(1)} - Y_i^{(0)} = \tau(\boldsymbol{x}_i) + \eta_i$, $\forall i \in \{1, ..., n\}$, which distinguishes it from other classical statistical learning problems (Chapter 1). The aim here would ideally be to select a model $\mathcal{T}^* \in \{\mathcal{T}_1, ..., \mathcal{T}_d\}$ within the (restricted) class of possible functions, that minimizes a loss (or risk) function on the estimated CATE $\hat{\tau}(\boldsymbol{x}_i)$, in a supervised learning manner. CATE squared loss function, of the type $\mathbb{E}_p[(\hat{\tau} - \tau)^2]$, is referred to as *Precision in Estimating Heterogeneous Treatment Effects* (PEHE) (Hill, 2011) and takes the following form: $\mathbb{E}_p\big[\big(\hat{\tau}(\boldsymbol{x}_i) - \tau(\boldsymbol{x}_i)\big)^2 \mid \boldsymbol{X}_i = \boldsymbol{x}_i\big]$. Thus the statistical learning problem, under utopic full observability of counterfactuals and PEHE loss function, could be depicted as

$$\tau^* \in \underset{\tau \in \mathcal{T}}{\arg\min}\ \mathbb{E}_p\Big[\ell\big(\tau(x), (y^{(1)} - y^{(0)})\big)\Big] \qquad \text{where}$$

$$\ell\big(\tau(x), (y^{(1)} - y^{(0)})\big) = \big[\tau(x) - (y^{(1)} - y^{(0)})\big]^2 .$$

Thus, PEHE would be an ideal loss function to use, but cannot be computed because of partial observability of the POs. Moreover, typical loss functions for standard regression problems (MSE, MAE, cross-entropy loss, etc.) computed directly against the observed outcome Y_i for estimating conditional average PO functions f_1, f_0 are not reliable measures for goodness of resulting CATE estimates for selection and confounding bias issues. For example, as discussed in Section 2.3.1.3 in the context of X-Learners, fitting good POs models in terms of their prediction error is not sufficient to guarantee good CATE estimates in presence of unbalanced designs. Attempts have been made in the literature to

develop a feasible way to select CATE models (e.g., Schuler et al. (2018); Alaa and Van Der Schaar (2019)), but none of the very few model selection procedures has been widely adopted for the reasons above. The R-loss encountered in (2.11) in the R-Learner framework could be in principle utilized to evaluate CATE estimates $\hat{\tau}(\cdot)$ stemming from any other Meta-Learning framework. However, this implicitly entails assuming Robinson (1988)'s and R-Learner's parametrization $Y_i = \mu(\boldsymbol{x}_i) + \tau(\boldsymbol{x}_i)A_i + \varepsilon_i$, with common error structure across groups.

From a more high-level perspective, the problem of model selection for CATE inference can be decomposed into three main tasks:

1) **Causally sufficient variable selection**. By causally sufficient variable selection we refer to a step which is aimed at ideally partitioning covariates into four distinct categories, namely: i) confounders, i.e. common causes of A_i and Y_i, to be included in both outcome and propensity model; ii) predictors of A_i, to be included in the propensity model only, with particular care given to the inclusion of IV and near-IV covariates (Pearl, 2011); iii) predictors of Y_i, to be included in both the outcome model and in the propensity model as well (Brookhart et al., 2006); iv) moderators of the treatment effects, which are a (not necessarily strict) subset of the outcome's predictors entering the CATE model only. In the case of "direct methods" not relying on propensity score adjustment, the problem naturally reduces to the specification of the outcome model only. The "causally sufficient" terminology here relates to the inclusion of confounders, which represents the smallest set of covariates to condition on that guarantees unbiased CATE estimates (Hill and Su, 2013), while variable selection in propensity and outcome model is meant to improve estimates' precision instead.

   Naturally, in empirical applications with large datasets, manual variable selection is not feasible, so one typically resorts to regularization techniques, after assuming unconfoundedness (i.e. we observe and include all

confounders in the model). Nonetheless, we once again stress how this requires some particular care given to some specific categories of PS covariates (see the end of Section 2.2 for a brief discussion). The interesting sub-task for policy-making purposes in heterogeneous treatment effects estimation is that of detecting the main moderators, possibly amongst several covariates. As we will discuss in detail in later sections and in Chapter 3, R- and τ-Learners have the comparative advantage to other Meta-Learners that they provide a straightforward framework to do so (Caron et al., 2022b). As also described earlier, by exploiting Robinson (1988) parametrization they specify a direct regularized model on CATE, that can easily return interpretable measures of "moderators importance". For example, a LASSO regression implementation of R-Learner (Nie and Wager, 2020) would return a sparse vector of moderators' coefficient; a shrinkage prior implementation of Bayesian Causal Forest can return posterior splitting probabilities on moderators instead (Caron et al., 2022b), as we will discuss in Chapter 3.

2) **Base-learner selection** refers to the problem of finding the best supervised algorithm for fitting the surfaces of interest via the outcome and propensity models. This sub-problem is essentially similar to a standard statistical learning one. A first step might be related to determining whether a parametric model is sufficient for adequately approximating relationships in the data. Non-parametric regression models provide flexibility to capture more complex patterns. Among non-parametric models, the functional space might be restricted further to satisfy parsimony and better generalization principles by applying regularization techniques, and also via domain knowledge incorporation by placing a prior distribution on the functional space. For example, one might consider splines or Gaussian Processes as more appropriate than tree-based methods for certain type of data, as they are better suited for fitting smooth functions. Some Meta-Learners, namely T-, X- and R-Learners offer the opportunity of

employing more than just one base-learner for different groups or different nuisance function (PS or outcome). For example, if the treated group has very few instances compared to the control, a parametric model is more likely to deliver better generalization properties; in R-Learners, different models can be adopted for fitting $m(\boldsymbol{x}_i)$ and then $\tau(\boldsymbol{x}_i)$.

3) **Meta-Learner selection**. Finally, the chosen base-learner has to be paired to one of feasible the Meta-Learners subroutines described in earlier subsections. By "feasible" here we refer to the fact that some of the Meta-Learners presented above do not support all types of base-learners. While S-, T-, X- and R-Learners allow for a high degree of flexibility in the choice of a base-learners, other frameworks are a bit more selective. τ-Learners envisage the use of Robinson's re-parametrization and a prior distribution (e.g., BART, GP, BNN, etc.) to be placed on $\mu(\cdot)$ and $\tau(\cdot)$, for a fully Bayesian inference; So far, to the best of our knowledge, they have been implemented in the context of BART (Hahn et al., 2020; Caron et al., 2022b) and (approximate) Bayesian Deep Learning (Caron et al., 2022c). Multitask-Learners only allow for multi-output learning algorithms (e.g BART have not been extended to multi-output problems yet). As we will discuss in more detail in the next section, the choice of a Meta-Learner is in practice primarily based on domain knowledge about the study at hand and on study-specific characteristics (i.e. suspected complexity of heterogeneity patterns, treatment groups imbalance, etc.), which can be viewed as an encoded prior on CATE function $p\big(\tau(\cdot)\big)$, that can be linked to different Meta-Learners' properties.

2.4 Simulation studies

In this section we report and comment on results from two different semi-simulated studies, carried out to compare performance of some of the models presented above in estimating CATE. A third supplemental semi-simulated study can be found in the Appendix Section A. A semi-simulated study here consists

in simulating only the outcome surface Y_i in the dataset $\mathcal{D}_i = \{\boldsymbol{X}_i, A_i, Y_i\}$, starting from real-world $\boldsymbol{X}_i$ and A_i. In the case of observational semi-simulations, Hill (2011) introduced a practical way of recreating an observational study from a randomized one. This is essentially done by leaving out a non-random portion of the treated group, so that treatment assignment is no longer randomized. Recreating an observational study from a purely randomized one has the main advantage of ensuring control over the selection mechanism, such that common support is guaranteed to hold, in this case, at least for the treated group. This means that average treatment effects on the treated (ATT and CATT) are identifiable, while those on the controls (ATC and CATC) are not.

We provide results based on the analysis of two real world randomized controlled trials, after transforming them into observational studies. The first semi-simulated setup employs the IHDP dataset, firstly introduced by Hill (2011) and popular in both the computer science and statistics literature on CATE estimation. The second and third setups instead employ the ACTG-175 dataset, and differ in the way the outcome and CATE are generated. Both code and the datasets to reproduce the results in this section are publicly available at `https://github.com/albicaron/EstITE`. As mentioned above, we present here below results from the IHDP data simulation and one of the two setups involving the ACTG-175 data, while we leave the other ACTG-175 setup in the Appendix, Section A.

The models we test are the following: random forests and BART respectively as S-, T- and X-Learners; LASSO regression, gradient boosted trees as R-Learners, and Causal Forest, which is a particular implementation of random forests as an R-Learner (Wager and Athey, 2018; Athey and Wager, 2019); two Multitask-Learners in the form of Multioutput Gaussian Processes, one with stationary kernel (Causal Multitask Gaussian Process - CMGP) and the other with non-stationary kernel (Non-Stationary Gaussian Process - NSGP), developed by Alaa and van der Schaar (2017) and Alaa and van der Schaar (2018) respectively; finally, Bayesian Causal Forest (Hahn et al., 2020; Caron

Meta-Learner	Label	Base-Learner
S-Learners	S-RF	Random Forest
	S-BART	BART
T-Learners	T-RF	Random Forest
	T-BART	BART
X-Learners	X-RF	Random Forest
	X-BART	BART
R-Learners	R-LASSO	LASSO Regression
	R-BOOST	Gradient Boosted Trees
	CF	Causal Forest
Multitask-Learners	CMGP	Causal Multi-task Gaussian Process
	NSGP	Non-Stationary Gaussian Process
τ-Learners	BCF	Bayesian Causal Forests (BART)

Table 2.2: List of Meta-Learner models compared in this experimental section. The "Base-Learner" column indicates which statistical learning (parametric or non-parametric) model is being used within the corresponding more general Meta-Learning framework.

et al., 2022b), which is a specific implementation of τ-Learner employing BART. A summary of the tested models is provided in Table 2.2. As explained in earlier sections, some of the methods envisage a specific role for propensity score, while for the other non-propensity methods we added PS estimates as an extra covariate (Hahn et al., 2020) to provide a comparison that decouples any difference in performance attributable to the inclusion of propensity score in the model.

For each of the two datasets analyzed, in order to provide a comparison of the methods presented above, we computed $\sqrt{\text{PEHE}}$ estimates for each of the $B = 1000$ Monte Carlo simulations, and we averaged it over all the simulations. Consistently with what we discussed earlier, $\sqrt{\text{PEHE}}$ was evaluated only in the covariate space regions corresponding to the treated units (thus on CATT) $\mathcal{X}_1$, as overlap is not guaranteed to hold on the covariate space regions of the controls (i.e. on CATC). Estimates of PEHE were obtained through its sample

equivalent, namely:

$$\widehat{\text{PEHE}}_\tau = \frac{1}{n_T} \sum_{i=1}^{n_T} \left(\tau(\boldsymbol{x}_i) - \widehat{\tau}(\boldsymbol{x}_i) \right)^2 , \tag{2.14}$$

where: subscript τ indicates PEHE is being computed on an estimate for τ; n_T is the size of the treated group; $\widehat{\tau}(\boldsymbol{x}_i)$ is the CATT estimate obtained under the given method; $\tau(\boldsymbol{x}_i)$ is the ground-truth CATT, always unknown in the real world. Data are randomly split in 70% train set used to train the models, and 30% test set to evaluate the model on unseen data. $\sqrt{\text{PEHE}}$ is reported for both train and test data. Supplementary results on all the simulated experiments regarding bias and $\sqrt{\text{PEHE}}$, evaluated also on CATC (out-of-overlap) regions, are provided in the appendix.

It is worth mentioning here that, although we measure PEHE on CATT point estimates, where overlap is theoretically guaranteed, there could still be regions of limited overlap where all models rely heavily on extrapolation. In general then, PEHE alone may not paint a reliable picture of models' performance if measured in (theoretically guaranteed, but) poor overlap regions. In fully simulated experiments (such as some of the ones appearing in the next chapters) we have more direct control on, and thus can limit, poor overlap. A more indicated way is to present PEHE alongside other performance measures involving intervals estimates rather than just point estimates (as we do in some of the next chapters' experiments), e.g., uncertainty coverage and credible intervals' width.

2.4.1 IHDP data

The first semi-simulated setup makes use of the IHDP dataset (Brooks-Gunn et al., 1992), popular in the literature for CATE estimation and used for the first time in Hill (2011). It includes data taken from the Infant Health and Development Program (IHDP), a randomized controlled trial aimed at investigating the efficacy of educational and family support services, with pediatric follow-ups, in improving cognitive skills of low birth weight preterm

infants, who are known to have developmental problems regarding visual-motor and receptive language skills (McCormick, 1985; McCormick et al., 1990). The study includes observations on 985 infants whose weight at birth was less than 2 500 grams, across 8 different sites. About one third of the infants were randomly assigned to treatment ($A_i = 1$), which consisted in routine pediatric follow-up (medical and developmental), in addition to frequent home visits to inform parents about child's progress and communicate instructions about recommended activities for the child. Following Hill (2011), the outcome variable (Y_i) we use is the score in a Stanford Binet IQ test, whose values can range from a minimum of 40 to a maximum of 160, taken at the end of the intervention period (child's age equal 3). The available final sample, obtained after removing 77 observations with missing IQ test score, consists of $n = 908$ data points. The dataset features 25 pre-treatment covariates, 6 continuous and 19 binary. The data are transformed into observational by leaving out a non-random portion of the treated individuals, namely those with non-white mothers. This leaves 139 observations in the treated group and 608 in the control group, for a total of 747 observations. Notice that removing a non-random portion of the treated inevitably generates lack of common support, as we no longer have children with non-white mothers in both treatment arms. For this reason estimating ATE and/or CATE would be unwise, as outlined in 1. Thus, we can resort to treatment effects estimation on the treated group only, i.e. ATT and CATT, where overlap is guaranteed to hold.

ITE is derived as the difference between the simulated potential outcomes, which are generated as:

$$\begin{aligned} Y^{(0)} &\sim \mathcal{N}\Big(\exp\big((X+W)\beta_B\big), 1\Big) \ , \\ Y^{(1)} &\sim \mathcal{N}\big(X\beta_B - \omega_B^b, 1\big) \ , \end{aligned} \tag{2.15}$$

where W is an offset matrix of same dimension as X with every entry equal to 0.5, and β_B is a 25-dimensional vector of regression coefficients $\big(0, 0.1, 0.2, 0.3, 0.4\big)$, sampled in each replication b of the experiment with probabilities $\big(0.6, 0.1,$

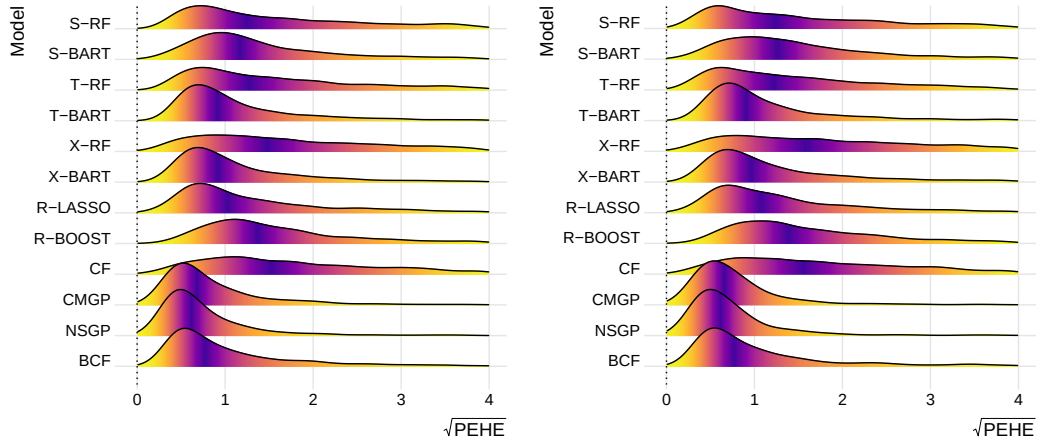


Figure 2.3: $\sqrt{\text{PEHE}}$ distribution in the train set (left) and test set (right), IHDP data.

$0.1, 0.1, 0.1\big)$, as described in the experiment "Response surface B" in Hill (2011). Following Hill (2011), coefficients β_B are re-sampled in each replication b to introduce some degree of sampling variation in the sparsity underlying the potential outcomes surfaces, i.e. different relevant covariates in each replication. For each replication $b \in \{1, ..., 1000\}$, ω_B^b is an offset chosen to guarantee that $\text{ATT} = \mathbb{E}\big[Y^{(1)} - Y^{(0)} \mid A = 1\big] = 4$.

Given the features of this specific simulated experiment, we might anticipate some of the Meta-Learners' behaviors, based on the properties that we laid out in the previous sections. First of all, we notice that, by the way POs are generated, CATT $\tau_1(\boldsymbol{x}_i) \in \mathcal{T}_1$ is bound to be a rather complex function. We expect this feature to particularly favours T-Learners and their extensions Multitask-Learners, over S-, X-, R- and τ-Learners, as they tackle CATT estimation problem by fitting two separate functions $f_0 \in \mathcal{F}_1$ and $f_1 \in \mathcal{F}_0$, which allows to capture very distinct, group-specific, degrees of complexity. Secondly, at a higher base-learner selection level, the conditional average potential outcomes generated in (2.15) are very smooth functions. This implies that base-learners enforcing a higher degree of smoothness via regularization (e.g. splines, Gaussian Processes, etc.) are well suited for the problem at hand.

Simulation results on performance are reported in Table 2.3 and Figure 2.3.

Table 2.3: Meta-Learners' results on IHDP and ACTG-175 data. $\sqrt{\text{PEHE}_\tau}$ estimates $\pm$ 95% confidence intervals for each tested model on CATT, on train and test sets respectively.

	IHDP		**ACTG-175**	
	Train	Test	Train	Test
S-RF	3.02 ± 0.23	2.96 ± 0.25	0.50 ± 0.01	0.48 ± 0.01
S-BART	1.75 ± 0.11	2.02 ± 0.15	0.43 ± 0.01	0.45 ± 0.01
T-RF	2.39 ± 0.17	2.28 ± 0.18	0.51 ± 0.01	0.49 ± 0.01
T-BART	1.35 ± 0.09	1.31 ± 0.09	0.48 ± 0.01	0.51 ± 0.01
X-RF	3.04 ± 0.21	3.15 ± 0.24	**0.34 ± 0.01**	**0.36 ± 0.01**
X-BART	1.34 ± 0.09	1.50 ± 0.12	0.44 ± 0.01	0.47 ± 0.01
R-LASSO	1.78 ± 0.13	1.82 ± 0.14	0.60 ± 0.01	0.63 ± 0.01
R-BOOST	2.04 ± 0.12	2.22 ± 0.15	0.47 ± 0.01	0.48 ± 0.01
CF	2.88 ± 0.19	2.84 ± 0.21	0.40 ± 0.01	0.39 ± 0.01
CMGP	**0.89 ± 0.05**	**0.84 ± 0.07**	0.42 ± 0.01	0.43 ± 0.01
NSGP	**0.80 ± 0.05**	**0.81 ± 0.07**	0.41 ± 0.01	0.42 ± 0.01
BCF	1.26 ± 0.09	1.22 ± 0.09	**0.36 ± 0.01**	**0.38 ± 0.01**

As anticipated by the above considerations, the best models appear to be the multitask Gaussian Processes (CMGP and NSGP) of Alaa and van der Schaar (2017) and Alaa and van der Schaar (2018). Also, T-Learners generally display better performance than their S- and X-Learner counterparts (particularly over S-RF, S-BART and X-RF, while X-BART has comparable performance to T-BART). Less anticipated is the performance of BCF (τ-Learner), which comes in second after the Gaussian Processes. This highlights BCF's ability to convey targeted Bayesian shrinkage on CATE, that, coupled with the flexibility in modelling the response function of BART, allows one to adjust to more (or less) complex CATE surfaces. In Figure 2.3, we report the empirical distribution of $\sqrt{\text{PEHE}_\tau}$ over the $B = 1000$ replications on both train and test data, for each of the models. We also notice that tree-based methods are relatively more prone to overfitting.

An important remark about the data generating process described by (2.15) is that it does not really induce strong confounding, since it is easy for a non-parametric base-learner to distinguish the two underlying polynomials

$\mathbb{E}[Y^{(A_i)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i]$. And since the two polynomials $\mathbb{E}[Y^{(A_i)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i]$ are extremely different from each other, CATE ends up being an unrealistically complex function. Besides, the fact that noise around $\mathbb{E}[Y^{(A_i)} \mid \boldsymbol{X}_i = \boldsymbol{x}_i]$ is independently simulated for the two potential outcomes produces extra noise-to-signal ratio on CATE, $\mathbb{V}(Y_i^{(1)} - Y_i^{(0)}) = \mathbb{V}(\varepsilon_{i,1}) + \mathbb{V}(\varepsilon_{i,0})$, that renders estimation challenging for every model in general. This implies that a relatively higher number of Monte Carlo replications of the experiment are needed to obtain estimates of $\sqrt{\text{PEHE}}$ with sufficiently low variance to effectively compare methods' performance (in this case $B = 1000$ appears to suffice). In the ACTG-175 simulated example illustrated in the next section, we will follow the parametrization found in Robinson (1988), Nie and Wager (2020) and Hahn et al. (2020) in the data generating process of the outcome surface, in order to induce stronger confounding (which is believed to be common in observational studies), generate a relatively simpler and more realistic CATE function, and avoid creating unnecessarily high noise around CATE.

2.4.2 ACTG-175 data

The second semi-simulated setup presented here is re-created using the ACTG-175 dataset. The data come from a randomized placebo-controlled trial aimed at comparing monotherapy versus a combination of therapies in HIV-1-infected subjects with CD4 cell counts between 200 and 500 (Hammer et al. (1996) for details). As in the case of IHDP data, an observational study is recreated by throwing away a non-random subset of patients, namely those not showing symptomatic HIV infection. The final dataset consists of 813 observations and 12 variables (3 continuous and 9 binary). The list of covariates included in the dataset are shown in Table 2.4.

Unlike the case of IHDP data, response surface Y_i is not generated via simulation of the two potential outcomes. Instead, we generate continuous outcome Y_i according to the parametrization

$$Y_i = \mu(\boldsymbol{X}_i) + \tau(\boldsymbol{X}_i)A_i + \varepsilon_i \ , \tag{2.16}$$

Table 2.4: ACTG-175 dataset variables

Variable	Description
age	Numeric
wtkg	Numeric
hemo	Binary (hemophilia = 1)
homo	Binary (homosexual = 1)
drugs	Binary (intravenous drug use = 1)
oprior	Binary (non-zidovudine antiretroviral therapy prior to initiation of study treatment = 1)
z30	Binary (zidovudine use in the 30 days prior to treatment initiation = 1)
preanti	Numeric (number of days of previously received antiretroviral therapy)
race	Binary
gender	Binary
str2	Binary: antiretroviral history (0 = naive, 1 = experienced)
karnof_hi	Binary: Karnofsky score (0 = < 100, 1 = 100)

which allows to specify CATE directly, instead of starting from the simulation of potential outcomes, and features a single error term ε_i. The prognostic score $\mu(\boldsymbol{x}_i)$ and CATE $\tau(\boldsymbol{x}_i)$ are generated as:

$$\begin{aligned}\mu(\boldsymbol{x}_i) &= 8 - 0.5hemo - |wtkg - 1| + 0.5gender - \frac{0.2}{age + 2} \\ &\quad + 0.5karnof_{hi} - 0.5z30 - 0.5race \\ \tau(\boldsymbol{x}_i) &= 1 + 0.2wtkg + 2\phi_Z(wtkg) \cdot (karnof_{hi} + 2) \ ,\end{aligned} \tag{2.17}$$

where $\phi_Z(\cdot)$ is the density function of a standard normal distribution. Noise is added by simulating normally distributed i.i.d. errors $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, with homoskedastic standard deviation equal to $\sigma = 0.2(\mu_{max} - \mu_{min})$, where μ_{max} is the sample maximum of the generated prognostic score, while μ_{min} is the sample minimum. Notice that, unlike the case of IHDP data, the error term is not simulated independently for the two groups, which avoids imposing too much noise around CATE. This translates into better signal-to-noise ration reflected in the PEHE estimates to evaluate the models, as shown both in Table 2.3 and Figure 2.4. As in the IHDP simulated example, $\sqrt{\text{PEHE}}$ is evaluated on the treated group only, given that only CATT is guaranteed to be identifiable.

In this second simulated setting, CATE is of simpler form. Hence, contrary to the IHDP setup, we expect learners that better accommodate less complex CATE functions (and thus higher confounding conveyed by the common prognostic score), such as S-, X- and τ-Learners, to perform better than T- and Multitask-Learners counterparts. In addition, the design is slightly unbalanced, with 281 individuals in the treated group and 532 in the control, a feature that might favour X-Learners. By inspecting the results reported in Table 2.3, we notice that X-RF and BCF are comparably the two best performing methods. As we have pinpointed earlier, this is thanks to their ability to detect simple heterogeneity patterns. S- and X-Learner implementation of BART are then relatively better than T-BART, while S-RF and T-RF do not exhibit any significant difference. CF (random forest R-Learner), which shares the characteristics of conveying targeted regularization with BCF, trails just behind X-RF and BCF. Finally, the two causal multitask Gaussian Processes perform reasonably well considering that the setup is not favourable to T- type of learners, as they tackle group imbalances by joint estimation of conditional average POs. Figure 2.4 depicts again the distribution of $\sqrt{\text{PEHE}}$ over the $B = 1000$ replications, for both train and test data, for all the tested models.

We employ the ACTG-175 data also in a third semi-simulated setup featuring more complex $\mu(\boldsymbol{x}_i)$ and $\tau(\boldsymbol{x}_i)$ surfaces, compared to the ones in (2.17). Description and results of this third example are provided in the Appendix Section A.2.

2.5 The effect of school meal programs on health indicators

Eventually, in this section we provide a full-length analysis of the NHANES data introduced in Section 2.1, previously analyzed by Chan et al. (2016), to demonstrate the use of CATE estimation methods and related tools in the study of heterogeneity. The dataset consists of $n = 2330$ observations and $P = 11$ covariates. The outcome variable of interest Y_i is child's BMI, while the

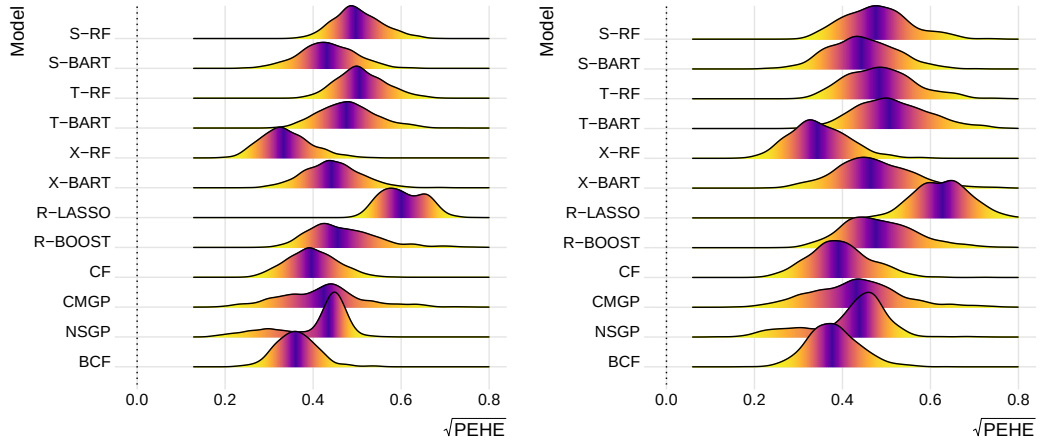


Figure 2.4: $\sqrt{\text{PEHE}}$ distribution in the train set (left) and test set (right), ACTG-175 data.

treatment A_i denotes participation in the National School Lunch or the School Breakfast programs, which are both designed to tackle poor or insufficient food access in low-income households. The full list of the variables, including the available covariates, is provided in Table A.5 in the Appendix. This specific setup suggests that the impact of participation in school meal programs might be heterogeneous across children, in that demographics such as age, gender or ethnicity might play a role in how effective participation is (e.g. younger kids might benefit more than older ones, etc.). This advocates the use of methods for CATE estimation.

By taking a rather agnostic approach to the problem, we decide to employ Bayesian Causal Forests (τ-Learner) (Hahn et al., 2020) in the analysis, for two primary reasons. Together with the causal multitask Gaussian Processes of (Alaa and van der Schaar, 2017, 2018), BCF was the most flexible method across different CATE simulations in the earlier section. In addition, as an advantage over causal multitask Gaussian Processes, BCF makes it easier to directly pick a BART prior to place on CATE and prognostic score $\mu(\cdot)$.

First of all, we employ a 1-hidden-layer neural network classifier to estimate the propensity score as a function of all the covariates, to be added as an additional covariate, as envisaged by the τ-Learner framework. We then run

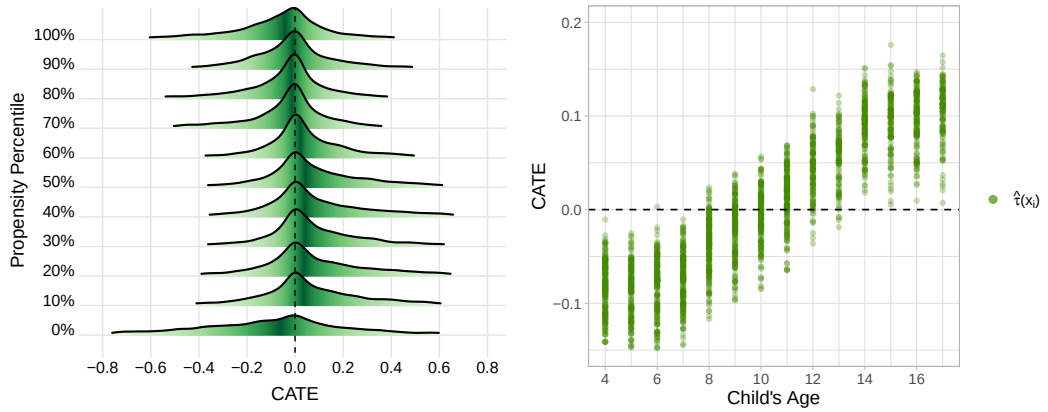


Figure 2.5: Left pane: BCF's posterior distribution estimates on CATE corresponding to the approximated propensity score percentiles (i.e. to individuals in the sample whose estimated propensity corresponds or is closest to PS percentiles). Right pane: BCF's CATE point estimates (averaged over the 5 000 post burn-in MCMC iterations) as a function of child's age.

Bayesian Causal Forest algorithm for a total of 10 000 MCMC iterations, of which the first 5 000 are discarded as burn-in. The left pane in Figure 2.5 represents the resulting CATE posterior distributions corresponding to different approximated propensity score percentiles (namely to individuals in the sample whose estimated propensity is equal or closest to the PS percentiles). This type of plot allows us to visualize uncertainty around individual CATE point estimates. CATE estimates seem to be concentrated around zero for all propensity score levels, which suggests a null average treatment effect (ATE) and very weak or absent heterogeneity patterns. Uncertainty patterns are quite consistent across different PS percentiles, apart from the minimum (i.e. 0% percentile) where CATE distribution is much more diffuse, potentially signalling a region of poor overlap.

Drawing attention to the study of moderation effects, we run a simple decision tree partition algorithm using the R package `rpart`, where average CATE point estimates $\hat{\tau}(\boldsymbol{x}_i)$ are treated as the target variable, while the covariates $\boldsymbol{X}_i \in \mathcal{X}$ are treated as predictors. The purpose of this exercise, that can done using CATE estimates obtained from any Meta-Learner, is to

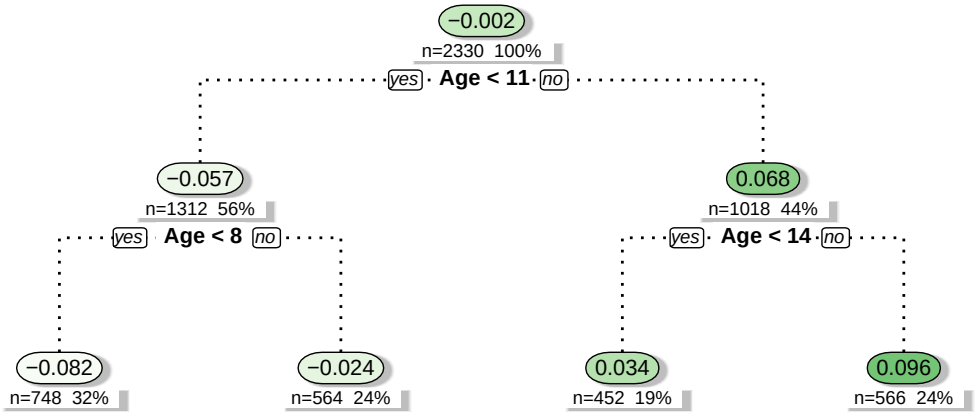


Figure 2.6: Decision tree indicating the most homogeneous subgroups in terms of treatment response, as a function of the available covariates (moderators). The nodes report CATE estimates averaged within the corresponding subgroup. The first node intuitively reports ATE estimate.

find ex-post (post estimation) the most homogeneous subgroups in terms of treatment response and the most informative moderating covariates. The results are depicted in Figure 2.6 in the form of a simple decision tree, where nodes report CATE estimates averaged within the corresponding subgroup, and provide evidence of very little, if not null, heterogeneity arising from Children's Age, given that the first two most informative splits in the tree feature this covariate, and that the estimated treatment effect is not very different across these subgroups. To better visualize this relationship, in the right pane of Figure 2.5 we plot point estimates of CATE against Children's Age, that show a weak but positive relationship. Figures 2.5 and 2.6 capture the role of Children's Age. Although it does not appear to be a major driver of propensity score (Ethnicity, Poverty Level and Participation to other Food Programs seem to be the main determinants of A — see Table A.6 in the Appendix), it is likely the main source of the, albeit small, moderation effects.

From the original analysis carried out by Chan et al. (2016) on the same NHANES dataset, it emerges that the estimated average treatment effect (ATE), on the 2007-2008 logged data, is significantly small, perhaps actually

null. As stated by the authors, this finding is likely attributable to the fact that the school meal programs are already well implemented, that is, treatment administration is already targeting the right population of individuals, with the policy implication that there is no need for re-designing it. To relate their findings to our analysis, we notice that results are very similar, in that we find no significant treatment effect across propensity score percentiles (Figure 2.5), and neither across subgroups defined by children's age (Figure 2.6). Treatment response patterns emerging from this analysis can be linked back to a settings similar to the ACTG simulation example, where causal effects are weakly heterogenous (in this case virtually constant and null), so that S-, X- and τ-Learners would be the preferred choice. Results from BCF implementation of τ-Learners in this analysis demonstrate its particular feature of being able, as described by Hahn et al. (2020), to shrink CATE estimates back to homogeneity if required, through targeted regularization.

2.6 Conclusions

In this chapter, we discussed the most recent developments on estimation of heterogeneous treatment effects in the context of observational studies. Our review of Meta-Learner frameworks and simulation studies lead to a few general observations. With regards to the properties of Meta-Learners reviewed in Section 2.3.1, there is a clear distinction between different groups of Meta-Learners according to the type of CATE functional complexity. S-Learners are appropriate when CATE displays simple heterogeneity patterns, while T-Learners when CATE is rather complex. The other Meta-Learners are instead designed to ideally adjust to different CATE complexity setups, although some of them appear to do so better than others. Multitask-Learners feature parameters sharing in the estimation of conditional average POs across "tasks" $A_i = a$, but being an extension of T-Learners they essentially assume separate models, with independent error terms, for the conditional average POs. For this reason, although they generally perform drastically better than T-Learners,

they appear to do slightly worse in settings with weak heterogeneity. The remaining batch, formed by X-, R- and τ-Learners, consist of methods that are instead specifically designed to address unbalanced designs and capture simple heterogeneity patterns, in different ways. X-Learners successfully extend T-Learners by applying propensity score re-weighting, while R- and τ-Learners are both propensity methods that apply targeted regularization/shrinkage by exploiting Robinson (1988) parametrization.

As for the results from the simulation exercises in Section 2.4, we described how BCF (τ-Learners) and causal multitask Gaussian Processes (Multitask-Learners) both performed consistently well across different simulated scenarios, with X-Learner trailing just a bit behind, and thus appear to be quite flexible and reliable methods to be chosen. BCF has also been shown to be particularly effective in addressing strong confounding (through incorporation of PS estimates).

Based on these findings, we suggest that model selection (or Meta-Learner selection) for CATE estimation in practice particularly benefits from restrictions placed on the CATE functional space capacity $\mathcal{T}$ stemming from domain knowledge, equivalently interpretable as placing a "prior" $p(\tau(\cdot))$ on possible models space. Now, while some of the methods reviewed above have to be adjusted ex-post (e.g., if we have unbalanced groups and suspect CATE is a relatively simple function, we estimate a T-Learner and then apply the X-Learner framework), some others enable us to do this ex-ante (R- and τ-Learner), regardless of whether we suspect CATE is relatively complex or not. In the next chapter indeed, we are going to discuss how we can exploit the Robinson (1988) parametrization by developing novel versions of R- and τ-Learners with additional desirable features, to simultaneously tackle the issues of targeted regularization and uncertainty-quantification and moderation effects interpretability in heterogeneous causal effects estimation.

Chapter 3

Interpretability, Regularization and Uncertainty Quantification in Causal Effects Learning

Contributions *The first part of this chapter is based on contents from a paper submitted and accepted to the Journal of Computational & Graphical Statistics (Caron et al., 2022b), with additional material explaining Bayesian tree-ensembles models more in details. The second part instead includes material based on a short workshop paper accepted to the 2nd Interpretable Machine Learning for Healthcare Workshop at ICML (Caron et al., 2022c).*

In this chapter, we focus on developing new Bayesian Non-Parametric (or Probabilistic Machine Learning) techniques for individual treatment effects estimation, that specifically build on top of R- and τ- type of Learners (Nie and Wager, 2020; Hahn et al., 2020) and Robinson's parametrization (Robinson, 1988). As mentioned in earlier chapters, Robinson's parametrization (Robinson, 1988) can generate advantages in terms of: i) **Interpretability** of moderation effects, if coupled with white-box models; ii) **Targeted regularization/shrinkage** on the treatment effects function CATE; iii) **Uncertainty quantification** on CATE, if coupled with appropriate Bayesian inference tools.

However, none of the existing reviewed methods has explicitly combined all these three components together. Thus, in this chapter we present two novel models that specifically address these three points simultaneously; one is based on Bayesian Additive Regression Trees (BART) (Chipman et al., 1998, 2010), while the other on a more interpretable variant of Neural Networks (NNs), called Neural Additive Models (NAMs) (Agarwal et al., 2021), coupled with approximate Bayesian deep learning inference techniques (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Pearce et al., 2020; Abdar et al., 2021).

We begin by introducing the popular BART model, developed by Chipman et al. (1998, 2010). We describe its non-parametric approach to fitting conditional mean functions in regression problems, and also how inference is achieved via Bayesian backfitting MCMC. BART has shown excellent performance in many predictive tasks, thanks to its ability to adapt to the underlying complexity of the conditional mean function (non-linearities, discontinuities, interaction between predictors, etc.) coupled with good Bayesian uncertainty coverage properties. For this reason it has also been successfully employed in causal inference tasks, with due modification to the particular setting, as we will discuss.

The second part of the chapter develops a shrinkage version of Bayesian Causal Forests, a recently proposed causal version of BART, that is specifically designed to estimate heterogeneous treatment effects under observational data, and study moderation effects. The shrinkage component we introduce is motivated by empirical studies where the number of pre-treatment covariates available is non-negligible, leading to different degrees of shrinkage underlying the surfaces of interest in the estimation of individual treatment effects. The extended version presented in this work, which we name Shrinkage Bayesian Causal Forest, is equipped with an additional pair of priors allowing the model to adjust the weight of each covariate through the corresponding number of splits in the tree ensemble. These priors improve on the model's computational efficiency and mixing time, adaptability to sparse settings and allow to perform

fully Bayesian variable selection in a framework for treatment effects estimation, and thus to uncover the moderating factors driving heterogeneity. In addition, the method allows for prior knowledge about the relevant confounding pre-treatment covariates and the relative magnitude of their impact on the outcome to be incorporated in the model.

We then illustrate the empirical performance of Shrinkage Bayesian Causal Forests in simulated studies, in comparison to Bayesian Causal Forest and other state-of-the-art models already encountered in Chapter 2, to demonstrate how it scales up with an increasing number of covariates and how it handles strongly confounded scenarios. We also provide an example of application using real-world data.

Finally, in the last part of the chapter, we introduce novel deep causal learning architecture, Interpretable Causal Neural Networks (ICNN), which exploit a type of R-Learner loss function, combining it with the interpretability of Neural Additive Models (NAMs) (Agarwal et al., 2021), rooted in the theory of Generalized Additive Models (GAMs), that can output Shapley value (Shapley, 1953) functions on moderating effects.

3.1 Bayesian Causal Forests

This section begins by introducing Bayesian Additive Regression Trees and Backfitting MCMC inference in them (Chipman et al., 1998, 2010). We will briefly see how BART offer a very flexible way of estimating the conditional mean function and adapt extremely well to non-linearities, lack of smoothness (discontinuities) and interactions between predictors. For these reasons, BART is not only widely used for prediction problems, but has also been successfully employed in causal inference. In the second part of this section, we will thus thoroughly review a popular causal version of BART, Bayesian Causal Forests, already encountered in the previous chapter in the context of τ-Learners, Section 2.3.1.6.

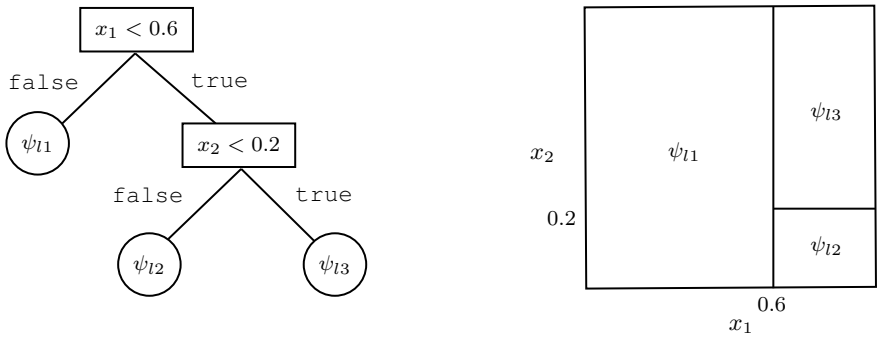


Figure 3.1: Simple tree structure, mapping inputs x_1 and x_2 to the terminal nodes $\boldsymbol{\psi} = \{\psi_{l1}, \psi_{l2}, \psi_{l3}\}$. Figure on the right represents the partition induced by the tree on the input space.

3.1.1 Regression Trees

The conventional definition of a "decision tree" is that it is a collection of recursive binary split rules of the type $\{x_j \geq c\}$ vs $\{x_j < c\}$ that maps inputs $\boldsymbol{X} = (X_1, ..., X_P) \in \mathcal{X}$ to a set of terminal nodes $\boldsymbol{\psi} \in \Psi$ located at its "leaves". More formally, a regression tree is a particular type of **Adaptive Basis-Function Model** (ABM). An ABM has the following general form to fit the conditional mean function of an outcome Y:

$$f(\boldsymbol{X}) = \mathbb{E}[Y \mid \boldsymbol{X}] = \psi_0 + \sum_{k=1}^{K} \psi_k \phi_k(\boldsymbol{X}; \varpi) \tag{3.1}$$

where $\phi_k(\cdot)$ is the k-th basis function and ϖ are its parameters. In the specific case of regression trees, the basis functions are defined by the binary split rules and the parameters $\boldsymbol{\psi} = (\psi_1 \ ... \ \psi_k)$ are the piecewise constant means in those regions; so that (3.1) becomes:

$$f(\boldsymbol{X}) = \mathbb{E}[Y \mid \boldsymbol{X}] = \sum_{k=1}^{b} \psi_k \mathbb{I}\big(\boldsymbol{x} \in \mathcal{J}_k\big) \tag{3.2}$$

where $\mathcal{J}_k$ denote the k-th region into which the input space is divided. Figure 3.1 shows an example of binary rules in a regression tree and the partition induced by its basis functions on two inputs $\boldsymbol{X} = [X_1 \ \ X_2]$. Although being a non-parametric model, simple regression trees sometimes struggle to capture

complexity of the response function. In particular, due to the piece-wise constant nature of the basis function approximation, they lack smoothness. To improve on complexity of trees, "forest" models, combining trees via e.g. *bagging* or *boosting*, such as in random forests (Breiman, 2001) and gradient boosted trees (Friedman, 2001), have been developed. Gradient boosted trees in particular share a lot of similarities with Bayesian Additive Regression Trees, considered their "Bayesian equivalent", as we will describe in the next section.

3.1.2 Bayesian Additive Regression Trees

Bayesian Additive Regression Trees (BART) are a non-parametric regression model that estimates the conditional mean of a response variable Y_i via a "sum-of-trees". Considering the regression framework in (2.1), one can use BART to flexibly represent $f(\cdot)$ as:

$$\begin{aligned} f(\boldsymbol{X}, A) &= \sum_{j=1}^{m} g_j\Big(\big[\boldsymbol{X} \;\; A\big],\; \big(T_j, M_j\big)\Big) \\ &= \sum_{j=1}^{m}\sum_{k=1}^{b} \psi_{j,k}\mathbb{I}_{\mathcal{J}_{j,k}}\big([\boldsymbol{X} \;\; A]\big)\;, \end{aligned} \tag{3.3}$$

where $\boldsymbol{X}$ and A are the covariate set and the binary treatment indicator, as in the previous chapter. We will now describe the rest of the seemingly daunting notation in (3.3) above. Starting from the first equality, m is the total number of trees in the model. The pair (T_j, M_j) defines the structure of the j-th tree, namely T_j embeds the collection of binary split rules while $M_j = \{\psi_{j,1}, ..., \psi_{j,b}\}$ the collection of b terminal nodes in that tree. $g_j(\cdot)$ is a tree-specific function mapping the predictors $(\boldsymbol{X}, A) \in \mathcal{X} \times \mathcal{A}$ to the set of terminal nodes $M_j \subseteq \mathbb{R}^b$, following the collection of binary split rules expressed by T_j. Hence, to expand further, $g_j\big([\boldsymbol{X}\; A],\, (T_j, M_j)\big) = \sum_{k=1}^{b} \psi_{j,k}\mathbb{I}_{\mathcal{J}_{j,k}}\big([\boldsymbol{X} \;\; A]\big)$ is essentially a step function where $\mathbb{I}_{\mathcal{J}_{j,k}}\big([\boldsymbol{x} \;\; a] \in \mathcal{J}_{j,k}\big) = 1$. The collection $\{\mathcal{J}_{j,k}\}_{k=1}^{b}$ denotes the sub-regions of the covariate space $\mathcal{X} \times \mathcal{A}$ defined by the partitioning rules of T_j. The intuition is that conditional mean function $f(\boldsymbol{x}, a) = \mathbb{E}\big[Y_i \mid \boldsymbol{X}_i = \boldsymbol{x}_i, A_i = a_i\big]$ fit is computed by summing up all the

terminal nodes $\psi_{j,k}$ assigned to the predictors $[\boldsymbol{X} \quad A]$ by the tree functions $g_j(\cdot)$, i.e. $\sum_{j=1}^{m} g_j(\cdot)$. The "sum-of-trees" structure is what helps BART better adapt to surface's smoothness.

3.1.2.1 BART priors

Inference in BART is achieved by placing a prior distribution on the models parameters $\theta = \big\{(T_1, M_1), ..., (T_m, M_m), \sigma\big\}$, where (T_j, M_j) is the stochastic tree structure and σ is the error term's homoskedastic standard deviation, such that $\text{Var}(\varepsilon_i) = \sigma^2$. Notice we have not included m, the number of trees in the ensemble. This is technically a parameter itself, but it is usually fixed ($m = 200$ has been shown to work nicely in a variety of prediction problems), or cross-validated, although this increases computational complexity of the algorithm. The priors chosen for θ are independent, and regularizing. Under independence between (T_j, M_j) and σ, we can rewrite the prior as

$$\begin{aligned} p\Big((T_1, M_1), ..., (T_m, M_m), \sigma\Big) &= \Bigg[\prod_{j=1}^{m} p\big((T_j, M_j)\big)\Bigg] p(\sigma) \\ &= \Bigg[\prod_{j=1}^{m} p(M_j|T_j)p(T_j)\Bigg] p(\sigma) \\ &= \Bigg[\prod_{j=1}^{m} \Big[\prod_{k=1}^{b} p(\psi_{kj}|T_j)\Big] p(T_j)\Bigg] p(\sigma) \ , \end{aligned} \tag{3.4}$$

so that its specification reduces to defining $p(T_j)$, $p(\psi_{kj}|T_j)$ and $p(\sigma)$, that depend on few hyperparameters pre-set or tuned from the data.

The prior for $p(T_j)$ is a branching process prior (Linero and Yang, 2018; Ročková and Saha, 2019), made of three subsequent components that shape the structure of the single tree T_j:

(i) The first component is concerned with determining the depth of each tree, in a stochastic way. To this end, the probability that a node at depth d_j, within tree j, is non-terminal is defined as

$$\frac{\alpha}{(1 + d_j)^{\beta}}, \qquad \text{where} \quad \alpha \in (0, 1), \ \beta \in [0, \infty) \tag{3.5}$$

and $d_j \in (0, 1, 2, 3, ...)$. Default values for the hyperparameters are $\alpha = 0.95$ and $\beta = 2$. These values ensures that most of the probability mass is assigned to small trees, with no more than $d_j = 5$ splits each, $\mathbb{P}(d_j > 5) \approx 1\%$. Small trees are a form of "weak learners" and are responsible of inducing regularization (once added all up, as we will see later) in a very similar way to gradient boosting (Friedman, 2001). BART approach however differs from frequentist-based supervised tree algorithms such as random forest and gradient boosted trees (Breiman, 2017), as regularization is conveyed through prior distribution and tree depth is not a deterministic, fixed, tuning hyperparameter, but stochastic.

(ii) Conditional on the tree depth d_j, the second component decides which subset of predictors $\boldsymbol{X}_{\text{split}} \subset \boldsymbol{X}$ will form the tree's binary splitting rules. In the default BART case, a uniform distribution is placed on the splitting variables so that each predictor x_p has P^{-1} probability of being picked (where P is the total number of predictors). We will see in later sections how this component can be modified to adaptively account for sparsity when learning $f(\cdot)$ and speed up MCMC convergence (Linero, 2018).

(iii) Finally, given the splitting variables, a uniform distribution is placed on the possible cutpoints/splitting values of each chosen x_p, such that $\mathcal{J}_p = \{x_p \in \mathcal{X}_p : x_p < c\}$ vs $\neg\mathcal{J}_p = \{x_p \in \mathcal{X}_p : x_p \geq c\}$ is the resulting binary rule. In case x_p is continuous, it is generally discretized to up to 10 000 possible cutpoints to choose from.

The prior for $\mathbf{p}(\psi_{\mathbf{kj}} \mid \mathbf{T_j})$ on the independent terminal nodes then reads

$$\psi_{kj} \mid T_j \sim \mathcal{N}(\mu_\psi, \sigma_\psi^2) \ ,$$

such that the induced prior on $f(x) = \mathbb{E}(Y \mid x)$, which is the sum of all the terminal nodes, is consequently $\mathcal{N}(m\mu_\psi, m\sigma_\psi^2)$. The hyperparameters μ_ψ and

σ_ψ^2 are set empirically by solving:

$$y_{min} = m\mu_\psi - k\sqrt{m}\sigma_\psi, \quad \text{and} \quad y_{max} = m\mu_\psi + k\sqrt{m}\sigma_\psi \ , \tag{3.6}$$

with default $k = 2$, which guarantees that $\mathbb{P}\big(f(\boldsymbol{x}) \in (y_{min}, y_{max})\big) \approx 95\%$. For convenience, the outcome Y is re-scaled using min-max normalization such that $y_{min} = -0.5$ and $y_{max} = 0.5$, which guarantees that $\psi_{ij} \sim \mathcal{N}(0, \sigma_\psi^2)$ where $\sigma_\psi = \frac{0.5}{k\sqrt{m}}$, i.e. the prior on ψ_{ij} is centered, and thus shrinks to 0.

The prior for $\mathbf{p}(\sigma)$ is a standard conjugate Inverse Chi-Square $\sigma^2 \sim \frac{\nu\lambda}{\chi^2_\nu}$, where the hyperparameters are the degrees of freedom ν and the scale λ. As in the case of $p(\psi_{kj} \mid T_j)$ hyperparameters, ν and λ are set empirically as follows: λ is such that the q-th quantile of the σ prior is located at $\widehat{\sigma}$, the standard deviation of the residuals $\hat{\varepsilon}_i$ obtained from a crude OLS regression of Y on $\boldsymbol{X}$, i.e. $\mathbb{P}(\sigma < \widehat{\sigma}) = q$, where $q = 0.90$ as default; and $\nu = 3$ by default. The combination of values $(\nu, q) = (3, 0.90)$ produces a prior which is neither overdispersed nor overconcentrated, and give generally nice results in a variety of applications. Other possible choices of q and ν are $q \in \{0.75, 0.99\}$ and $\nu = 10$ respectively. In particular, $(\nu, q) = (10, 0.75)$ gives a more disperse and conservative distribution, while $(\nu, q) = (3, 0.99)$ a more aggressive and concentrated one.

3.1.2.2 Bayesian Backfitting MCMC in BART

Default hyperparameter specification in BART follows an "Empirical Bayes" approach that avoids computationally expensive parameter tuning, considering having to restart the MCMC chain multiple times. As a consequence, the computational complexity of BART is entirely determined by its MCMC inference, that we briefly describe here. BART's MCMC algorithm takes the form of a "Metropolis-Hasting within Gibbs" sampling procedure, utilized to approximately sample from the posterior distribution

$$p(\theta \mid y) \overset{\text{def}}{=} p\Big((T_1, M_1), ..., (T_m, M_m), \sigma \mid y\Big) \ .$$

For notational convenience, we denote $T_{(-j)}$ and $M_{(-j)}$ as the set of all tree structures and all terminal nodes, except for those of the j-th tree, i.e. $(-j) = (1, ..., j-1, j+1, ..., m)$. A single Gibbs sampler MCMC iteration implies m successive draws of (T_j, M_j) from the full conditional distribution:

$$p\Big((T_j, M_j) \mid (T_{(-j)}, M_{(-j)}), \sigma, y)\Big) \ , \tag{3.7}$$

more specifically with an initial draw from the full conditional of T_j (determining tree depth, splitting variables and cutpoints) $p\big(T_j \mid (T_{(-j)}, M_{(-j)}), \sigma, y)\big)$, and then from the full conditional of $p\big(M_j \mid T_j, (T_{(-j)}, M_{(-j)}), \sigma, y)\big)$. Later, a draw from the full conditional

$$p\Big(\sigma \mid (T_1, M_1), ..., (T_j, M_j), y\Big) \tag{3.8}$$

is obtained, corresponding to a draw from an inverse-gamma. Draws from (3.7) are obtained as follows. It can be seen that (3.7) depends on $\big((T_{(-j)}, M_{(-j)}), \sigma)\big)$ only through the partial residuals $R_j = y - \sum_{k \neq j} g\big(x, (T_k, M_k)\big)$, i.e. residuals from the fit excluding the j-th tree. Draws in (3.7) are thus equivalent to draws from:

$$p\big((T_j, M_j) \mid R_j, \sigma\big) \tag{3.9}$$

which uniquely identifies tree-related parameter in the single-tree regression model described by $R_j = g_j\big(x, (T_j, M_j)\big) + \varepsilon$, whose likelihood function $p(R_j|\theta)$ is still Gaussian. We can obtain draws from this posterior by sampling subsequently from $p(T_j|R_j, \sigma)$ and then from $p(M_j|T_j, R_j, \sigma)$. While the latter has a conjugate Gaussian form, and can be efficiently sampled in block (Murray, 2020), the former can be expressed up to a normalizing constant as

$$p(T_j \mid R_j, \sigma) \propto p(T_j) p(R_j \mid T_j, \sigma) \ . \tag{3.10}$$

Draws from the above j-th tree posterior (3.10) can be obtained using a Metropolis-Hasting (Metropolis et al., 1953; Hastings, 1970) step proposed in

Chipman et al. (1998). This approach aims at generating a reversible Markov chain sequence of trees, which is $T^0, T^1, T^2, ...$, whose limiting distribution is $p(T_j|R_j, \sigma)$. The MH algorithm step (within the Gibbs sampler) works in the following way:

1. Start with an initial tree T^0 (the initial tree will be of depth $d = 0$)

2. Generate new candidate tree T^*, according to the kernel probability $q(T^i, T^*)$

3. Set $T^{i+1} = T^*$ with acceptance probability:

$$\alpha(T^i, T^*) = \min\left\{1, \frac{q(T^*, T^i)}{q(T^i, T^*)} \frac{p(T^*)p(R_j|T^*, \sigma)}{p(T^i)p(R_j|T^i, \sigma)}\right\}$$

   otherwise, set $T^{i+1} = T^i$.

The transition kernel density generating the proposal T^*, $q(T^i, T^*)$, assigns different probabilities to the following possible moves in the search space of trees T^j, to be applied to the current tree T^i (i.e. *local* moves):

1. **Grow**: with probability 0.25, pick a terminal node randomly and grow a new one by randomly assigning a splitting rule by randomly picking one of the remaining predictors and randomly choose a splitting value

2. **Prune**: with probability 0.25, pick a random pair of terminal nodes and prune them

3. **Change**: with probability 0.40, pick a random internal node and randomly reassign it to a new splitting rule (i.e. randomly pick a different predictor and randomly pick one possible splitting value)

4. **Swap**: with probability 0.10, pick a random parent-child pair of internal nodes and swap their position

The transition kernel $q(T^i, T^*)$ allows the sampler to generate trees that efficiently adapt to underlying complexity (discontinuity, interactions, etc.) in the data. However, it is also been pointed out that the default kernel $q(\cdot)$, associated with the four possible actions above, is responsible for slow-mixing

in the chain for its moves are only *local* within the search space, and bear the risk of getting trapped in local modes. Some contributions, such as Pratola (2016) and He et al. (2019) have attempted to speed mixing up by changing $q(\cdot)$. In particular the "grow-from-root" approach (He et al., 2019), where each new tree T_j in the chain is essentially grown from scratch instead of being a modified version of the previous T_{j-1} (implying then a *global* move) seems to significantly improve on mixing time, as it guarantees faster convergence. Finally, the draw from the full conditional of M_j is made of independent draws of the terminal nodes ψ_{ij} from a conjugate normal distribution.

For more details about BART prior specification and inference we refer to the seminal work of Chipman et al. (1998), Chipman et al. (2010), Linero and Yang (2018), Ročková and Saha (2019) and Rocková et al. (2020), among others.

3.1.3 BART for Causal Inference

Given their excellent predictive performance, BART models have been successfully applied also in causal inference settings, starting from the early contribution of Hill (2011), who first proposed it. In this section we will thoroughly describe a popular causal version of BART, named Bayesian Causal Forests (Hahn et al., 2020).

As already mentioned in Section 2.3.1.6 in the previous chapter, Bayesian Causal Forests make use of Robinson's parametrization in a full Bayesian fashion, by treating CATE $\tau(\boldsymbol{x}_i) \in \mathcal{T}$ as a parameter with its own prior $p(\tau(\cdot))$. In this way, through Robinson's parametrization, one can avoid imposing indirect CATE regularization on $f(\cdot) \in \mathcal{F}$ instead, as described in (2.1). In fact, as shown by Hahn et al. (2018) and Hahn et al. (2020), regularization on $f(\cdot) \in \mathcal{F}$ can generate unintended bias in the final estimation of CATE. Bayesian Causal Forests thus uses the following two-stage representation, which can be traced back to the work of Heckman (1979):

$$A_i \sim \text{Bernoulli}\big(\pi(\tilde{\boldsymbol{X}}_i)\big) \ , \ \pi(\tilde{\boldsymbol{x}}_i) = \mathbb{P}(A_i = 1 \mid \tilde{\boldsymbol{X}}_i = \tilde{\boldsymbol{x}}_i) \ , \tag{3.11}$$

$$Y_i = \mu\Big(\big[\boldsymbol{X}_i \;\; \pi(\tilde{\boldsymbol{X}}_i)\big]\Big) + \tau(\boldsymbol{W}_i)A_i + \varepsilon_i \;. \tag{3.12}$$

The first stage (3.11) deals with propensity score $\pi(\tilde{\boldsymbol{x}}_i) \in \Pi$ (Π is also known as the policies space) estimation, for which any probabilistic classifier is suitable (e.g. logistic regression, deep neural nets, etc.). Note that the regressors in the propensity score model are denoted by $\tilde{\boldsymbol{X}}_i \subseteq \boldsymbol{X}_i \in \mathcal{X}$ to indicate that a subset of covariates different to $\boldsymbol{X}_i$ can be employed. These $\tilde{\boldsymbol{X}}_i \subseteq \boldsymbol{X}_i \in \mathcal{X}$ could be the result of a combination of automated and manual variable selection procedures, where the manual component is required to guaranteed that no detrimental covariate for causal effects estimation (collider, IV, near-IV, etc.) is accidentally included by the prediction-driven automated selection. The second stage (3.12) estimates the prognostic score $\mu(\cdot)$, defined as the effect of the covariates $\boldsymbol{X}_i \in \mathcal{X}$ on the outcome Y_i in the absence of treatment $\mu(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i \mid \boldsymbol{X}_i = \boldsymbol{x}_i, A_i = 0\big]$, and CATE $\tau(\cdot)$. Notice that $\boldsymbol{W}_i \subseteq \boldsymbol{X}_i \in \mathcal{X}$ appears instead of $\boldsymbol{X}_i$ in the $\tau(\cdot)$ function, to highlight the fact that, as in the propensity score model, a different set of covariates may be used for CATE estimation. In general we use slightly different notation for the covariates in $\mu(\cdot)$, $\tau(\cdot)$ and $\pi(\cdot)$ to highlight the fact that the set of available covariates $\boldsymbol{X}_i \in \mathcal{X}$ might consist of four different types: i) **confounders**, i.e. direct and indirect common causes of A and Y; ii) **prognostic covariates**, i.e. predictors of $\mu(\cdot)$ only; iii) **moderators**, i.e. predictors of $\tau(\cdot)$ only; iv) **propensity covariates**, entering only $\pi(\cdot)$ equation. Any covariate that does not fall into one of these categories is an irrelevant/nuisance predictor.

The two-stage procedure described above belongs to a class of models known as "modularized", as opposed to full-model approaches that attempt to embed uncertainty around propensity scores in a single stage, which can lead to poor estimates due to feedback issues in the "cut-posterior" approximation of the full posterior (Zigler et al., 2013; Zigler and Dominici, 2014), which attempts to fix feedback problems. See Jacob et al. (2017) for a thorough discussion on the issue of modularized versus full/joint models.

The advantage of the parametrization in (3.12) from a Bayesian standpoint lies in the fact that separate priors can be placed on the prognostic score $\mu(\cdot)$ and on CATE $\tau(\cdot)$ directly. This approach mitigates unintended bias attributable to what the authors in Hahn et al. (2018, 2020) call *Regularization Induced Confounding* (RIC). The intuition behind RIC is that CATE posterior is strongly influenced by the regularization effects of the prior on $f(\cdot)$ in (2.1), such that posterior estimates of CATE are bound to be biased, even more so in presence of strong confounding, such as when treatment selection is suspected to be "targeted", i.e., when individuals are selected into treatment based on the prediction of an adverse potential outcome if left untreated. In order to alleviate confounding from targeted selection, the authors suggest to employ propensity score estimates obtained from the first stage $\hat{\pi}$ as an additional covariate in the estimation of $\mu(\cdot)$, in the hope that a good approximation of $\pi(\cdot) \in \Pi$ can account also for unobserved confounding, given that we observe good proxies for them (Tchetgen et al., 2020).

In practice, a BCF model assigns a default BART prior to $\mu(\cdot)$, while a prior with stronger regularization is chosen for $\tau(\cdot)$, as moderating patterns are believed to be simpler. The BART prior on $\tau(\cdot)$, compared to the default specification, consists in the use of a smaller number of trees in the ensemble (50 trees instead of 200), and a different combination of hyperparameters that govern the depth of each tree. In particular, in the context of BART priors, the probability that a node at depth $d \in \{0, 1, 2, ...\}$ in a tree is non-terminal is given by $\nu(1 + \beta)^{-d}$, where (ν, β) are the hyperparameters to set (Chipman et al., 2010). The default specification $(\nu, \beta) = (0.95, 2)$ already has a shrinkage effect that accommodates small trees. The BCF prior on $\tau(\cdot)$ instead sets $(\nu, \beta) = (0.25, 3)$, with the purpose of assigning higher probability mass to even smaller trees. This combination of hyperparameters in the CATE prior allows to detect weak heterogeneous patterns, and provides robustness in case of homogeneous treatment effects.

For the reasons illustrated above, BCF tends to outperform BART and

other tree-based methods for CATE estimation, such as Causal Forests (Wager and Athey, 2018). As we will illustrate in the following sections, our work extends the BCF framework by introducing explicit shrinkage of irrelevant predictors, which results into higher computational efficiency, and accommodates different levels of smoothness across covariates, while, at the same time, returning interpretable measures of feature importance in the estimation of $\mu(\cdot)$ and $\tau(\cdot)$, separately.

3.2 Shrinkage Bayesian Causal Forests

BART, and consequently BCF, are known to cope with sparse DGPs reasonably well, thanks to the fact that splitting variables are chosen uniformly at random in the sampling for $p(T_j|\cdot)$. However, they do not actively implement "heterogeneous" sparsity, nor feature shrinkage, which in BCF inevitably implies imposing equal importance to all the moderating covariates $\boldsymbol{X}_i$ responsible for the heterogeneity. In addition, the complexity of CATE estimation under high-dimensional covariate space inevitably depends on the smoothness and sparsity of the surfaces of interest (Alaa and van der Schaar, 2018), and thus necessitates regularization. Accounting for sparsity would then generally improve performance. At the same time, prior subject-matter knowledge on the relative importance of the covariates may be available, and can improve estimates and/or convergence if embedded in the model. In light of these considerations, in this section we propose a method that enriches the BCF model, and differ from the ones reviewed in the previous chapter in that it allows to jointly: i) account for heterogeneous smoothness and sparsity across covariates; ii) tease apart prognostic and moderating covariates through targeted variable selection; iii) incorporate prior knowledge on the relevant covariates and their relative impact on the outcome.

We start by briefly illustrating the notion of variable selection in the context of tree ensemble models such as BART. Let us define first $\boldsymbol{s} = (s_1, ..., s_P)$ as the vector of splitting probabilities of each predictor $j \in \{1, ..., P\}$, where

each s_j represents the probability, for the j-th predictor, of being chosen as a splitting variable in one of the decision nodes of a tree. The default version of BART places a uniform distribution over the splitting variables, meaning that each predictor has equal chance of being picked as a splitting variable: $s_j = P^{-1} \;\; \forall j \in \{1, ..., P\}$. As a consequence, predictors are virtually given equal importance in the final fit. A sparsity-inducing solution in this framework implies having a vector $\boldsymbol{s}$ of "stick-breaking" posterior splitting probabilities where ideally the entries corresponding to irrelevant predictors are very close to zero, while the ones corresponding to relevant predictors are significantly higher than P^{-1}. Posterior splitting probabilities in this context can be intuitively viewed as a measure of variables importance (Breiman, 2001). A complementary, decision-theoretic interpretation of sparsity-inducing solutions in this setup is given by the posterior probabilities that a predictor j appears in a decision node at least once in the ensemble. The two interpretations above (variables importance and probability of inclusion) are interchangeable and qualitatively lead to the same conclusions. In the next section we review how the sparse extension of of BART proposed by Linero (2018) can accommodate sparse solutions as described above, and how this modified version of BART can be put to use in the context of Bayesian Causal Forests.

3.2.1 Dirichlet Additive Regression Trees

Dirichlet Additive Regression Trees (Linero, 2018), or DART, constitute an effective yet practical way of inducing sparsity in BART. The proposed modification consists in placing an additional Dirichlet prior on the vector of splitting probabilities $\boldsymbol{s}$, which triggers a consequent posterior update in the backfitting MCMC algorithm. The Dirichlet prior on $\boldsymbol{s}$ reads

$$(s_1, ..., s_P) \sim \text{Dirichlet}\left(\frac{\alpha}{P}, ..., \frac{\alpha}{P}\right) \;, \tag{3.13}$$

where α is the hyperparameter governing the a priori preference for sparsity. Lower values of α correspond to sparser solutions, that is, fewer predictors

included in the model. The hyperparameter α is in turn assigned a prior distribution, in order to deal with unknown degree of sparsity. This prior is chosen to be a Beta distribution, placed over a standardized version of the α parameter, of the following form

$$\frac{\alpha}{\alpha + \rho} \sim \text{Beta}(a, b) \ , \tag{3.14}$$

where the default parameter values are $(a, b, \rho) = (0.5, 1, P)$. The combination of values $a = 0.5$ and $b = 1$ assigns higher probability to low values of α, thus giving preference to sparse solutions (the combination $(a, b) = (1, 1)$ would instead revert back to default BART splitting probabilities, i.e. uniform distribution over the splitting variables). The prior is assigned to the standardized version of α in (3.14) instead of α directly, as this allows to easily govern preference for sparsity through the parameter ρ. If one suspects that the level of sparsity is, although unknown, rather high, setting a smaller value of ρ facilitates even sparser solutions.

The modified version of DART's MCMC implies an extra step to update $\boldsymbol{s}$, according to the conjugate posterior

$$s_1, ..., s_P \mid (u_1, ..., u_P) \sim \text{Dirichlet}\left(\frac{\alpha}{P} + u_1, ..., \frac{\alpha}{P} + u_P\right) \ , \tag{3.15}$$

where the update depends on u_j, defined as the number of attempted splits on the j-th predictor in the current MCMC iteration. The phrase "attempted splits" refers to the fact that BART MCMC algorithm generates trees through a branching process undergoing a Metropolis-Hastings step, so that a proposed tree in the process might be rejected, but the chosen splitting variables are counted anyway in $\boldsymbol{u} = (u_1, ..., u_P)$ (Chipman et al., 1998, 2010; Linero and Yang, 2018). The rationale behind the update in (3.15) follows the natural Dirichlet-Multinomial conjugacy. The more frequently a variable is chosen for a splitting rule in the trees of the ensemble in a given MCMC iteration (or equivalently the higher is u_j), the higher the weight given to that variable by

the updated $\boldsymbol{s} \mid (u_1, ..., u_P)$ in the next MCMC iteration. Hence, the higher s_j, the higher the chance for the j-th predictor of being drawn as splitting variable from the multinomial distribution described by Multinomial$\big(1,\ \boldsymbol{s} \mid \boldsymbol{u}\big)$. This extra Gibbs step comes at negligible computational cost when compared to default BART typical running time.

3.2.2 Shrinkage BCF priors

Similarly to Linero (2018), symmetric Dirichlet priors can be straightforwardly embedded in the Bayesian Causal Forest framework to induce sparsity in the estimation of prognostic and moderating effects. Bearing in mind that, as described in the previous section, BCF prior consists in two different sets of independent BART priors, respectively placed on the prognostic score $\mu(\cdot)$ and CATE $\tau(\cdot)$, our proposed extension implies adding an additional Dirichlet prior over the splitting probabilities to these BART priors. Throughout the rest of the work we will consider the case where $\boldsymbol{W}_i = \boldsymbol{X}_i$, i.e. where the same set of covariates is used for the estimation of $\mu(\cdot)$ and $\tau(\cdot)$ (see eq. (3.12) for reference), but the ideas easily extend to scenarios where a different set of covariates is designed, based on domain knowledge, to be used for $\mu(\cdot)$ and $\tau(\cdot)$[1]. The additional priors are respectively

$$\begin{aligned} \boldsymbol{s}_\mu &\sim \text{Dirichlet}\left(\frac{\alpha_\mu}{P+1}, ..., \frac{\alpha_\mu}{P+1}\right) , \qquad & \frac{\alpha_\mu}{\alpha_\mu + \rho_\mu} &\sim \text{Beta}(a, b) \\ \boldsymbol{s}_\tau &\sim \text{Dirichlet}\left(\frac{\alpha_\tau}{P}, ..., \frac{\alpha_\tau}{P}\right) , \qquad & \frac{\alpha_\tau}{\alpha_\tau + \rho_\tau} &\sim \text{Beta}(a, b) , \end{aligned} \tag{3.16}$$

where the Beta's parameters are chosen to be $(a, b) = (.5, 1)$ as default. The hyperparameter ρ is set equal to $(P + 1)$ in the case of the prognostic score ($\rho_\mu = P + 1$) since, when estimating $\mu(\boldsymbol{x}_i)$, we make use of P covariates plus an estimate of the propensity score $\widehat{\pi}(\boldsymbol{x}_i)$ as an additional covariate. In the case of $\tau(\boldsymbol{x}_i)$, we set it equal to $\rho_\tau = \frac{P}{2}$ to give preference to even more aggressive

[1] In certain cases, the set of pre-treatment covariates might benefit from an initial screening by the researcher in the design of the study, and later undergo feature shrinkage in Shrinkage BCF, with the possibility of incorporating further a priori knowledge through the prior distributions, as described later in this section. As we will show in Section 3.2.3, in fact, Shrinkage BCF not only adjusts to sparse data generating processes (DGPs) per se, but allocates splitting probabilities in a more efficient way among the covariates, compared to uniformly at random splits, increasing computational efficiency.

Algorithm 1: Bayesian Backfitting MCMC in Shrinkage BCF

Input: Data (X, A, Y)

Output: MCMC samples of

$$\left\{\mu^{(b)}(\cdot), \tau^{(b)}(\cdot), (\boldsymbol{s}_\mu \mid \boldsymbol{u}_\mu)^{(b)}, (\boldsymbol{s}_\tau \mid \boldsymbol{u}_\tau)^{(b)}, \sigma^{(b)}\right\}_{b=1}^{B}$$

for $b = 1, ..., B$ **do**

  Result: Sample $\mu^{(b)}(\boldsymbol{x}), (\boldsymbol{s}_\mu \mid \boldsymbol{u}_\mu)^{(b)}$

  for $j = 1, ..., m_\mu$ **do**

    Sample tree structure $T_j\mu \sim p(T_j|R_j, \sigma) \propto p(T_j)p(R_j|T_j, \sigma)$

    Sample terminal nodes $M_j\mu \sim p(M_j|T_j, R_j, \sigma)$ (conjugate normal)

  end

  Sample

  $(\boldsymbol{s}_\mu \mid \boldsymbol{u}_\mu) \sim \text{Dirichlet}\big(\alpha_\mu/(P+1) + u_{1\mu},\ \dots\ , \alpha_\mu/(P+1) + u_{(P+1)\mu}\big)$

  Result: Sample $\tau^{(b)}(\boldsymbol{x}), (\boldsymbol{s}_\tau \mid \boldsymbol{u}_\tau)^{(b)}$

  for $j = 1, ..., m_\tau$ **do**

    Sample tree structure $T_j\tau \sim p(T_j|R_j, \sigma) \propto p(T_j)p(R_j|T_j, \sigma)$

    Sample terminal nodes $M_j\tau \sim p(M_j|T_j, R_j, \sigma)$ (conjugate normal)

  end

  Sample $(\boldsymbol{s}_\tau \mid \boldsymbol{u}_\tau) \sim \text{Dirichlet}\big(\alpha_\tau/P + u_{1\tau},\ \dots\ , \alpha_\tau/P + u_{P\tau}\big)$

  Result: Sample $\sigma^{(b)}$

  Sample $\sigma \sim p\big(\sigma|\widehat{\mu}(\boldsymbol{x}_i), \widehat{\tau}(\boldsymbol{x}_i), Y\big)$

end

shrinkage, as CATE is typically believed to display simple heterogeneity patterns and a higher degree of sparsity compared to the prognostic score.

We refer to this setup as Shrinkage Bayesian Causal Forest (Shrinkage BCF). Naturally, the two Dirichlet priors trigger two separate extra sampling steps in the Gibbs sampler, implementing draws from the conjugate posteriors:

$$\begin{aligned} \boldsymbol{s}_\mu \mid \boldsymbol{u}_\mu &\sim \text{Dirichlet}\big(\alpha_\mu/(P+1) + u_{1\mu},\ \dots\ , \alpha_\mu/(P+1) + u_{(P+1)\mu}\big) \\ \boldsymbol{s}_\tau \mid \boldsymbol{u}_\tau &\sim \text{Dirichlet}\big(\alpha_\tau/P + u_{1\tau},\ \dots\ , \alpha_\tau/P + u_{P\tau}\big)\ , \end{aligned} \tag{3.17}$$

where concentration parameter for $\boldsymbol{s}_\mu \mid \boldsymbol{u}_\mu$ is rescaled by $(P+1)$, as we are including a propensity score estimate $\hat{\pi}$ in the covariates. Shrinkage BCF's setup allows first of all to adjust to different degrees of sparsity in $\mu(\cdot)$ and $\tau(\cdot)$, and thus to induce different levels of smoothness across the covariates. Secondly, it naturally outputs feature importance measures on both the prognostic score

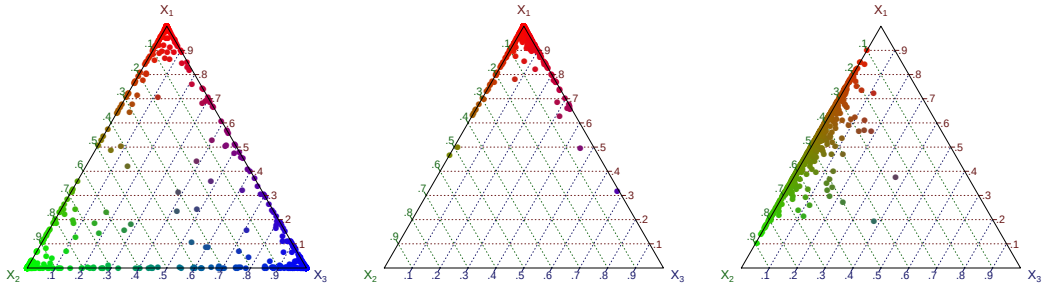


Figure 3.2: Dirichlet draws from Dirichlet$(0.1, 0.1, 0.1)$ (left), Dirichlet$(5, 0.1, 0.1)$ (centre) and Dirichlet$(5, 5, 0.1)$ in the case of $P = 3$ variables.

and CATE separately, given that separate draws of the posterior splitting probabilities are returned. The extra computational time, per MCMC iteration, is slightly higher, albeit negligible, compared to default BCF; however, Shrinkage BCF demonstrates higher computational efficiency and better mixing (Linero and Yang, 2018; He et al., 2019), thanks to the fact that it avoids splitting on irrelevant covariates and guides exploration of the search space accordingly. Thus, it necessitate far fewer MCMC iterations to converge, and improves performance under sparse DGPs. A sketch of pseudo-code illustrating the backfitting MCMC algorithm in Shrinkage BCF can be found in Box 1.

The Dirichlet priors in Shrinkage BCF can be also adjusted to convey prior information about the relevant covariates and their relative impact on the outcome. This can be achieved by introducing a set of scalar prior weights $\boldsymbol{k} = \{k_1, ..., k_P\} \in \mathbb{R}_+^P$, such that

$$\begin{aligned} \boldsymbol{s}_\mu &\sim \text{Dirichlet}\left(k_{1\mu}\frac{\alpha_\mu}{P+1}, \ldots, k_{(P+1)\mu}\frac{\alpha_\mu}{P+1}\right) , \\ \boldsymbol{s}_\tau &\sim \text{Dirichlet}\left(k_{1\tau}\frac{\alpha_\tau}{P}, \ldots, k_{P\tau}\frac{\alpha_\mu}{P}\right) . \end{aligned} \tag{3.18}$$

The weights can take on different values for each covariate and can be set separately for prognostic score and CATE. If the j-th covariate is believed to be significant in predicting $\mu(\cdot)$, then its corresponding prior weight $k_{j\mu}$ can be set higher than the others, in order to generate draws from a Dirichlet distribution

that allocate higher splitting probability to that covariate. In the simulated experiment sections later we will introduce a version of Shrinkage BCF with informative priors assigning higher a priori weight to the propensity score $\pi(\boldsymbol{x}_i)$ in $\mu\big(\boldsymbol{x}_i, \pi(\boldsymbol{x}_i)\big)$, to investigate whether this helps tackling strong confounding.

In Figure 3.2 we provide a visual example to illustrate how increasing the value of the parameters of the Dirichlet distribution leads to more dense draws in proximity of a specific covariate. For ease of visual representation, Figure 3.2 depicts a simple case of $P = 3$ predictors. The first graph on the left depicts equally sparse Dirichlet draws, similar to those obtain from the Dirichlet prior in the non-informative Sparse BCF version. The graph in the center and on the right show what happens if one or two covariates are given higher weight: the stick-breaking process allocates most probability to X_1 in the center plot, and to X_1 and X_2 in the right plot, while assigning near zero probability to X_3.

3.2.3 Experiment 1: Targeted sparsity and covariate heterogeneity

As a result of a fully Bayesian approach to feature shrinkage, Shrinkage BCF returns non-uniform posterior splitting probabilities that assign higher weight to more predictive covariates. This automatically translates into more splits along covariates with higher predictive power, compared to default BCF. To investigate whether this more strategic allocation of splitting probabilities in Shrinkage BCF leads to better performance, we test it against a default version of BCF including all the covariates and a version of BCF that already employs the subset of relevant covariates only. Think of the latter as a sort of "oracle" BCF that knows a priori the subset of relevant covariates, but may not assign different weights to them in terms of relative importance in the estimation of $\mu(\cdot)$ and $\tau(\cdot)$ respectively. To this end, we run a simple simulated example with $P = 10$ correlated covariates, of which only 5 are relevant, meaning that they exert some effect on the prognostic score or on CATE. We compare default BCF, "oracle" BCF using only the 5 relevant covariates and Shrinkage BCF using all the covariates (5 relevant and 5 nuisance). We generate the $P = 10$

covariates from a multivariate Gaussian $(X_1, ..., X_{10}) \sim \mathcal{N}(\mathbf{0}, \Sigma)$, where the entries of the covariance matrix are such that $\Sigma_{jk} = 0.6^{|j-k|} + 0.1\mathbb{I}(j \neq k)$, indicating positive correlation between predictors. Sample size is set equal to $n = 1000$. We then generate treatment assignment as $A_i \sim \text{Bern}\big(\pi(\boldsymbol{x}_i)\big)$, where the propensity score is

$$\pi(\boldsymbol{x}_i) = \mathbb{P}(A_i = 1 \mid \boldsymbol{X}_i = \boldsymbol{x}_i) = \Phi\big(-0.4 + 0.3X_{i,1} + 0.2X_{i,2}\big) \ , \tag{3.19}$$

and $\Phi(\cdot)$ is the cumulative distribution function of a standard normal distribution. The prognostic score, CATE and response Y_i are respectively generated as

$$\begin{aligned} \mu(\boldsymbol{X}_i) &= 3 + X_{i,1} + 0.8\sin(X_{i,2}) + 0.7X_{i,3}X_{i,4} - X_{i,5} \ , \\ \tau(\boldsymbol{X}_i) &= 2 + 0.8X_{i,1} - 0.3X_{i,2}^2 \ , \\ Y_i \ &= \mu(\boldsymbol{x}_i) + \tau(\boldsymbol{x}_i)A_i + \varepsilon_i \ , \quad \text{where} \quad \varepsilon_i \sim \mathcal{N}(0,1) \ . \end{aligned} \tag{3.20}$$

In this experiment only the first five predictors $\{X_1, ..., X_5\}$ are relevant. Table 3.1 shows performances of the default BCF, "oracle" BCF run on just the 5 relevant predictors (oracle BCF-5) and Shrinkage BCF (SH-BCF), averaged over $H = 500$ replications. Performance of the methods is measured through: bias, defined as $\mathbb{E}\big[(\hat{\tau}_i - \tau_i) \mid \boldsymbol{X}_i = \boldsymbol{x}_i\big]$; the quadratic loss function

$$\mathbb{E}\big[(\hat{\tau}_i - \tau_i)^2 \mid \boldsymbol{X}_i = \boldsymbol{x}_i\big] \ , \tag{3.21}$$

where $\hat{\tau}_i$ is the model-specific CATE estimate, while τ_i is the ground-truth CATE; and finally 95% frequentist coverage, defined as $\mathbb{P}\big(\hat{\tau}(\boldsymbol{x_i})_{low} \leq \tau(\boldsymbol{x_i}) \leq \hat{\tau}(\boldsymbol{x_i})_{upp}\big)$, where $\hat{\tau}(\boldsymbol{x_i})_{\{low,high\}}$ are the upper and lower bounds of 95% credible interval around $\hat{\tau}(\boldsymbol{x_i})$, returned by the MCMC. The loss function in (3.21) is also known as the *Precision in Estimating Heterogeneous Treatment Effects* (PEHE) from Hill (2011). Bias, PEHE and coverage estimates are estimated by computing,

Model	Bias	$\sqrt{\text{PEHE}}$	95% Coverage
BCF	0.037 ± 0.008	0.447 ± 0.006	**0.92 ± 0.01**
Oracle BCF-5	0.034 ± 0.008	0.440 ± 0.006	0.91 ± 0.01
SH-BCF	**0.031 ± 0.007**	**0.380 ± 0.006**	0.88 ± 0.01

Table 3.1: Sample average bias, $\sqrt{\text{PEHE}}$ and 95% coverage for default BCF, "oracle" BCF which uses only the 5 relevant predictors (Oracle BCF-5) and Shrinkage BCF (SH-BCF). Bold text represents better performance.

for each of the $H = 500$ Monte Carlo simulations, their sample equivalents

$$\widehat{\text{Bias}}_\tau = \frac{1}{n}\sum_{i=1}^{n}\Big(\hat{\tau}(\boldsymbol{x}_i) - \tau(\boldsymbol{x}_i)\Big)$$
$$\widehat{\text{PEHE}}_\tau = \frac{1}{n}\sum_{i=1}^{n}\Big(\hat{\tau}(\boldsymbol{x}_i) - \tau(\boldsymbol{x}_i)\Big)^2$$
$$\widehat{\text{Coverage}}_\tau = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\Big(\hat{\tau}(\boldsymbol{x_i})_{low} \leq \tau(\boldsymbol{x_i}) \leq \hat{\tau}(\boldsymbol{x_i})_{upp}\Big)\ ,$$

and then averaging these over all the simulations. More precisely, Table 3.1 reports bias, $\sqrt{\text{PEHE}}$ and coverage estimates together with 95% Monte Carlo confidence intervals.

Shrinkage BCF shows better performance than default BCF as well as the "oracle" BCF version in terms of bias and $\sqrt{\text{PEHE}}$, while reports just marginally lower coverage, indicating that the method allocates "stick-breaking" splitting probabilities in an efficient way and necessitates fewer MCMC iterations for convergence. The intuition as to why Shrinkage BCF performs better than "oracle" BCF, is that its priors allow not only to split more along relevant covariates instead of irrelevant ones (which explains the advantage over BCF in terms of mixing), but also to split more frequently along covariates that are more predictive of the outcome, resulting in higher computational efficiency. To illustrate this concept, consider how fit $\hat{f}(\cdot)$ is constructed in simple tree algorithms, i.e. piecewise constant, and suppose we have the following trivial linear DGP with two covariates on the same scale, $Y = 2X_1 + X_2$. Both covariates are relevant for predicting Y, but X_1 has a relatively higher impact

in magnitude. DART, and thus Shrinkage BCF, allocates more splits along the more predictive dimension X_1, while BART produces a similar level of splits along both X_1 and X_2 and hence requires a larger number of MCMC iterations to converge.

3.2.4 Experiment 2: Targeted regularization in confounded studies

The parametrization in BCF, and thus in Shrinkage BCF as well, is designed to effectively disentangle prognostic and moderating effects of the covariates and to induce different levels of sparsity when estimating these effects, in contrast to other methods for CATE estimation. The purpose of this section is to briefly illustrate with a simple example how naively introducing sparsity through a model that does not explicitly guard against RIC can have a detrimental effect on CATE estimates. To this end, we simulate, for $n = 1000$ observations, $P = 5$ correlated covariates as $(X_1, ..., X_5) \sim \mathcal{N}(\mathbf{0}, \Sigma)$, where the entries of the covariance matrix are $\Sigma_{jk} = 0.6^{|j-k|} + 0.1\mathbb{I}(j \neq k)$. The treatment allocation, prognostic score, CATE and response Y_i are then respectively generated as follows:

$$
\begin{aligned}
A_i &\sim \text{Bernoulli}\big(\pi(\boldsymbol{x}_i)\big), \quad \pi(\boldsymbol{x}_i) = \Phi\big(-0.5 + 0.4X_{i,1}\big) \ , \\
\mu(\boldsymbol{X}_i) &= 3 + X_{i,1} \ , \quad \tau(\boldsymbol{X}_i) = 0.5 + 0.5X_{i,2}^2 \ , \\
Y_i &= \mu(\boldsymbol{x}_i) + \tau(\boldsymbol{x}_i)A_i + \varepsilon_i \ , \quad \text{where} \quad \varepsilon_i \sim \mathcal{N}(0,1) \ .
\end{aligned}
$$

Notice that in this simple setup the prognostic effect is determined by the first covariate $X_{i,1}$, while the moderating effect by the second covariate $X_{i,2}$. We run CATE estimation via three different methods that make use of DART priors. The first is a S-Learner that employs DART (S-DART) to fit a single surface $f(\cdot)$ and computes CATE estimates as $\hat{\tau}(\boldsymbol{x}_i) = \hat{f}(\boldsymbol{x}_i, A_i = 1) - \hat{f}(\boldsymbol{x}_i, A_i = 0)$. The second is a T-Learner that employs DART (T-DART) to fit two separate surfaces, $f_1(\cdot)$ and $f_0(\cdot)$, for the two treatment groups and derives CATE estimates as $\hat{\tau}(\boldsymbol{x}_i) = \hat{f}_1(\boldsymbol{x}_i) - \hat{f}_0(\boldsymbol{x}_i)$. The last method is our Shrinkage BCF

Method		Variable					
		X_1	X_2	X_3	X_4	X_5	A
S-DART	$f(\cdot)$	**0.12**	**0.43**	0.00	0.00	0.00	**0.45**
T-DART	$f_0(\cdot)$	**0.99**	0.00	0.00	0.01	0.00	-
	$f_1(\cdot)$	**0.19**	**0.80**	0.00	0.01	0.00	-
SH-BCF	$\mu(\cdot)$	**0.98**	0.01	0.00	0.00	0.01	-
	$\tau(\cdot)$	0.00	**0.96**	0.00	0.03	0.01	-

Table 3.2: Posterior splitting probabilities from S-Learner DART, T-Learner DART and Shrinkage BCF over the 5 available covariates. Values in bold denote which covariates receive significant chunks of splitting probability in fitting the corresponding functions, that characterize each model.

(SH-BCF). Each of these methods is able to account for sparsity when estimating CATE. However, the interpretation of covariate importance is very different across them, due to the way the CATE estimator is derived. In particular, as indicated by the posterior splitting probabilities of each method in Table 3.2, S-DART fits a single surface $f(\cdot)$, where A is treated as an extra covariate, so it ends up assigning most of the splitting probability to A and then in turn to other relevant covariates. T-DART performs "group-specific" feature shrinkage, in that it fits separate surfaces for each of the treatment groups. Although both S-DART and T-DART turn out to select the relevant covariates for the final estimation of CATE, they are unable, by construction, to distinguish between prognostic and moderating ones. Shrinkage BCF instead, thanks to its parametrization, is capable of doing so, disentangling the two effects.

In the upcoming experiments section, we will show that Shrinkage BCF outperforms default BCF and other state-of-the-art methods in estimating CATE under two more challenging simulated exercises. Furthermore, in the supplementary material we present results from few additional simulated experiments.

3.2.5 Simulated and Real-World Application

In this section, we report results from two simulated studies carried out to demonstrate the performance of Shrinkage BCF and its informative prior version under sparse DGPs. The first simulated study is intended to evaluate Shrinkage BCF performance compared to other popular state-of-the-art methods for CATE estimation, and to show how it scales up with an increasing number of nuisance covariates. In addition, we will also illustrate how the method returns interpretable feature importance measures, as posterior splitting probabilities on $\mu(\cdot)$ and $\tau(\cdot)$. The second simulated setup instead mimics a strongly confounded study, and is designed to show how versions of Shrinkage BCF deal with targeted selection scenarios. In the supplementary material, we present further results from four additional simulated exercises, designed to: i) study what happens with perfectly known propensity scores in confounded settings; ii) investigate computational advantage of DART priors; iii) test Shrinkage BCF's reliability under increasingly larger P; iv) consider different types of sparse DGPs.

3.2.6 Comparison to other methods

The first setup consists of two parallel simulated studies, where only the total number of predictors ($P = 25$ and $P = 50$) is modified. The purpose underlying this setup is to illustrate how Shrinkage BCF relative performance scales up when nuisance predictors are added and the level of sparsity increases.

For both simulated exercises, sample size is set equal to $n = 1000$. In order to introduce correlation between the covariates, they are generated as correlated uniforms from a Gaussian Copula $C_{\Theta}^{\text{Gauss}}(u) = \Phi_{\Theta}\big(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_P)\big)$, where Θ is a covariance matrix such that $\Theta_{jk} = 0.3^{|j-k|} + 0.1\mathbb{I}(j \neq k)$. A 40% fraction of the covariates is generated as continuous, drawn from a standard normal distribution $\mathcal{N}(0, 1)$, while the remaining 60% as binary, drawn from a binomial $\text{Bin}(n, 0.3)$. Propensity score is generated as:

$$\pi(\boldsymbol{x}_i) = \mathbb{P}(A_i = 1 \mid \boldsymbol{X}_i = \boldsymbol{x}_i) = \Phi\left(-0.5 + 0.2X_{i,1} + 0.1X_{i,2} + 0.4X_{i,21} + \frac{\eta_i}{10}\right),$$

Family	Label	Description
Linear Models	S-OLS	Linear regression as S-Learner
	T-OLS	Linear regression as T-Learner
	R-LASSO	LASSO regression as R-Learner
Naive Non-Parametrics	kNN	k-Nearest Neighbors as T-Learner
Tree-Based Methods	S-BART	BART as S-Learner
	T-BART	BART as T-Learner
	CF	Causal Forest
	S-DART	DART as S-Learner
	T-DART	DART as T-Learner
	BCF	Bayesian Causal Forest
	SH-BCF	Shrinkage Bayesian Causal Forest
Gaussian Processes	CMGP	Causal Multi-task Gaussian Process
	NSGP	Non-Stationary Gaussian Process

Table 3.3: List of models tested on the simulated experiment in Section 3.2.6.

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal, and η_i is a noise component drawn from a Uniform$(0, 1)$. The binary treatment indicator is drawn as $A_i \sim \text{Bernoulli}\big(\pi(\boldsymbol{x}_i)\big)$. Prognostic score and CATE functions are simulated as follows:

$$\begin{aligned} \mu(\boldsymbol{x}_i) &= 3 + 1.5\sin(\pi X_{i,1}) + 0.5(X_{i,2} - 0.5)^2 + 1.5(2 - |X_{i,3}|) + \\ &\quad + 1.5X_{i,4}(X_{i,21} + 1) \\ \tau(\boldsymbol{x}_i) &= 0.1 + |X_{i,1} - 1|(X_{i,21} + 2) \ . \end{aligned} \tag{3.22}$$

Notice that only 5 predictors among $P \in \{25, 50\}$, namely $\{X_1, X_2, X_3, X_4, X_{21}\}$, are relevant to the estimation of the prognostic score and CATE. Eventually, the response variable Y_i is generated as usual:

$$Y_i = \mu(\boldsymbol{x}_i) + \tau(\boldsymbol{x}_i)A_i + \varepsilon_i \ , \quad \text{where} \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \ .$$

The error term standard deviation is set equal to $\sigma = \frac{\hat{\sigma}_\mu}{2}$, where $\hat{\sigma}_\mu$ is the sample standard deviation of the simulated prognostic score $\mu(\boldsymbol{x}_i)$ in (3.22).

Performance of each method is evaluated through $\sqrt{\text{PEHE}}$ estimates,

	$P = 25$		$P = 50$	
	Train	Test	Train	Test
S-OLS	1.91 ± 0.00	1.91 ± 0.01	1.91 ± 0.00	1.91 ± 0.01
T-OLS	1.41 ± 0.01	1.47 ± 0.01	1.68 ± 0.01	1.78 ± 0.01
R-LASSO	1.17 ± 0.01	1.19 ± 0.01	1.20 ± 0.01	1.22 ± 0.01
kNN	1.62 ± 0.01	1.66 ± 0.01	1.72 ± 0.01	1.76 ± 0.01
S-BART	0.77 ± 0.01	0.79 ± 0.01	0.85 ± 0.01	0.86 ± 0.01
T-BART	1.11 ± 0.01	1.11 ± 0.01	1.28 ± 0.01	1.29 ± 0.01
CF	1.05 ± 0.01	1.05 ± 0.01	1.23 ± 0.01	1.23 ± 0.01
S-DART	0.59 ± 0.01	0.60 ± 0.01	0.59 ± 0.01	0.60 ± 0.01
T-DART	0.88 ± 0.01	0.89 ± 0.01	0.90 ± 0.01	0.90 ± 0.01
BCF	0.79 ± 0.01	0.82 ± 0.01	0.86 ± 0.01	0.88 ± 0.01
SH-BCF	**0.54 ± 0.01**	**0.56 ± 0.01**	**0.55 ± 0.01**	**0.55 ± 0.01**
CMGP	0.59 ± 0.01	0.61 ± 0.01	0.85 ± 0.03	0.77 ± 0.02
NSGP	0.60 ± 0.01	0.62 ± 0.01	0.74 ± 0.03	0.75 ± 0.03

Table 3.4: Train and test set $\sqrt{\text{PEHE}}$ estimates, together with 95% confidence interval, in the case of $P = 25$ covariates and $P = 50$ covariates scenarios.

averaged over $H = 1000$ replications, reported together with 95% Monte Carlo confidence intervals. Data are randomly split in 70% train set, used to train the models, and 30% test set to evaluate the model on unseen data; $\sqrt{\text{PEHE}}$ estimates are reported both for train and test data.

The models evaluated on the simulated data are summarized in Table 3.3. The first set of models includes a S-Learner and a T-Learner least squares regressions (S-OLS and T-OLS), and a R-Learner (Nie and Wager, 2020) LASSO regression (R-LASSO). The second set consists just in a naive k-nearest neighbors (kNN) as a T-Learner. The third set includes the following popular tree ensembles methods: Causal Forest (CF) (Wager and Athey, 2018); a S-Learner and a T-Learner versions of BART (S-BART and T-BART) and DART (S-DART and T-DART); Bayesian Causal Forest (BCF) (Hahn et al., 2020); and finally our method, Shrinkage Bayesian Causal Forest (SH-BCF). The last set includes two causal multitask versions of Gaussian Processes, with stationary (CMGP) and non-stationary (NSGP) kernels respectively, both implementing sparsity-inducing Automatic Relevance Determination over the

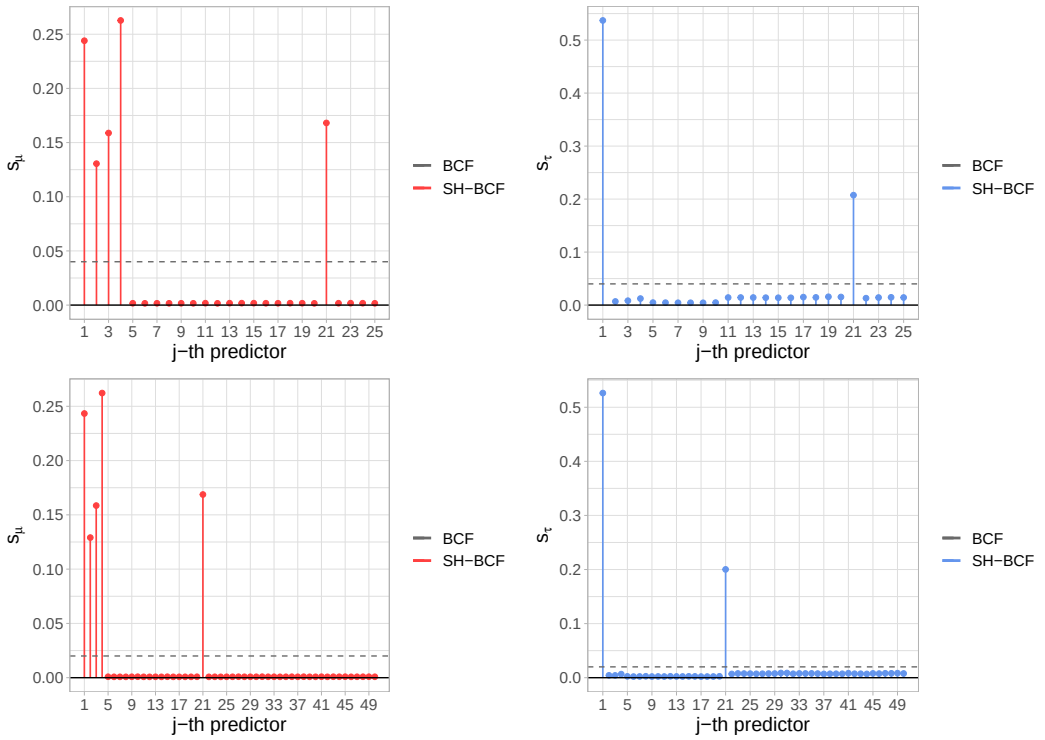


Figure 3.3: Shrinkage BCF posterior splitting probabilities for each single covariates, indexed on the x-axis, for $\mu(\cdot)$ (on the left) and $\tau(\cdot)$ (on the right), in the scenarios with $P = 25$ predictors (first row) and $P = 50$ predictors (second row). Spikes indicate higher probability assigned by Shrinkage BCF to the relevant predictors. The horizontal dashed lines denote default BCF uniform splitting probabilities.

covariates (Alaa and van der Schaar, 2017, 2018).

Performance of each method, for the two simulated scenarios with $P = 25$ and $P = 50$ covariates respectively, is shown in Table 3.4. Results demonstrate the high adaptability and scalability of Shrinkage BCF, as the method displays the lowest estimated error in both simulated scenarios, and its performance is not undermined when extra nuisance covariates are added, while the other methods generally deteriorate.

Figure 3.3 shows how Shrinkage BCF correctly picks the relevant covariates behind both prognostic and moderating effects, in contrast to default BCF which assigns equal probability of being chosen as a splitting variable to each predictor. Notice also that results do not essentially vary between the $P = 25$

and the $P = 50$ scenarios (respectively first and second row graphs in Figure 3.3), as Shrinkage BCF virtually selects the same relevant predictors.

3.2.7 Strongly confounded simulated study

We presents here results from a second simulated study, aimed at showing how Shrinkage BCF addresses scenarios characterized by strong confounding. In particular, the setup is designed around the concept of targeted selection, a common type of selection bias in observational studies, expressively tackled by the BCF framework, that implies a direct relationship between $\mu(\cdot)$ and $\pi(\cdot)$. We run the simulated experiment in the usual way, by firstly estimating the unknown propensity score; then we also re-run the same experiment assuming that propensity score is known (results in the supplementary material), to gain insights by netting out effects due to propensity model misspecification.

We simulate $n = 500$ observations from $P = 15$ correlated covariates (the first 5 continuous and the remaining 10 binary), generated as correlated uniforms from the Gaussian Copula $C_{\Theta}^{\text{Gauss}}(u) = \Phi_{\Theta}\big(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_P)\big)$, where the covariance matrix is such that $\Theta_{jk} = 0.6^{|j-k|} + 0.1\mathbb{I}(j \neq k)$. The relevant quantities are simulated as follows:

$$
\begin{aligned}
\mu(\boldsymbol{x}_i) &= 5\Big(2 + 0.5\sin(\pi X_{i,1}) - 0.25X_{i,2}^2 + 0.75X_{i,3}X_{i,9}\Big) \ , \\
\tau(\boldsymbol{x}_i) &= 1 + 2|X_{i,4}| + 1X_{i,10} \ , \\
\pi(\boldsymbol{x}_i) &= 0.9 \ \Lambda\left(1.2 + 0.2\mu(\boldsymbol{x}_i)\right) \ , \\
A_i &\sim \text{Bernoulli}\big(\pi(\boldsymbol{x}_i)\big) \ , \\
Y_i &= \mu(\boldsymbol{x}_i) + \tau(\boldsymbol{x}_i)A_i + \varepsilon_i \ , \quad \text{where} \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \ ,
\end{aligned}
\tag{3.23}
$$

where $\Lambda(\cdot)$ is the logistic cumulative distribution function. The error's standard deviation is set equal to half the sample standard deviation of the generated $\tau(\cdot)$, $\sigma^2 = \frac{\hat{\sigma}_{\tau}}{2}$. Targeted selection is introduced by generating the propensity score $\pi(\boldsymbol{x}_i)$ as a function of the prognostic score $\mu(\boldsymbol{x}_i)$ (Hahn et al., 2020). The BCF models tested on this simulated setup are: i) Default BCF; ii) agnostic prior Shrinkage BCF; iii) agnostic prior Shrinkage BCF, without propensity

Model	Bias	$\sqrt{\text{PEHE}}$	95% Coverage	$(s_\pi \mid u_\pi)$
i) BCF	-0.06 ± 0.01	0.49 ± 0.01	0.94 ± 0.00	9.09%
ii) SH-BCF	-0.05 ± 0.01	0.38 ± 0.01	0.96 ± 0.00	0.29%
iii) SH-BCF (no PS)	-0.05 ± 0.01	0.38 ± 0.01	0.96 ± 0.00	-
iv) I-BCF ($k_{PS} = 50$)	-0.05 ± 0.01	0.39 ± 0.01	0.96 ± 0.00	9.76%
v) I-BCF ($k_{PS} = 100$)	-0.05 ± 0.01	0.40 ± 0.01	0.96 ± 0.01	17.48%

Table 3.5: Bias, $\sqrt{\text{PEHE}}$, 95% Coverage and posterior splitting probability on $\hat{\pi}(x_i)$ — $(s_\pi \mid u_\pi)$ — for: i) default BCF; ii) Shrinkage BCF; iii) Shrinkage BCF without $\hat{\pi}(x_i)$; iv) informative prior BCF with $k_{PS} = 50$; v) informative prior BCF with $k_{PS} = 100$.

score estimate as an additional covariate; iv) Shrinkage BCF with informative prior on $\mu(\cdot)$ only, where prior weight given to propensity score is $k_{PS} = 50$; v) Shrinkage BCF with the same prior as iv), but $k_{PS} = 100$. We test a variety of BCF versions to examine how they tackle confounding deriving from targeted selection. In particular, with iv) and v), we investigate whether nudging more splits on the propensity score covariate induces better handling of confounding and better CATE estimates. With ii) and iii) we study whether it is sensible to have propensity score as an extra covariate, once we have accounted for sparsity, in settings such as the one described in (3.23), where propensity $\pi(\cdot)$ and prognostic score $\mu(\cdot)$ are functions of the same set of covariates — more specifically $\pi(\cdot)$ is a function of $\mu(\cdot)$.

We first compare the usual performance metrics (bias, $\sqrt{\text{PEHE}}$, 95% coverage), averaged over $H = 500$ replications, which are gathered in Table 3.5, together with the average posterior splitting probability assigned to propensity score $(s_\pi \mid u_\pi)$ by each model, where applicable. As for the posterior splitting probability $(s_\pi \mid u_\pi)$, we notice that in ii) this is nearly zero, thus not really different than not having $\pi(\cdot)$ at all, as in iii). This means that estimates of $\pi(\cdot)$ do not virtually contribute a lot to the fit. Also, in i) and iv), the probability is more or less the same, meaning that, in this example, setting $k_{PS} = 50$ implies assigning similar $(s_\pi \mid u_\pi)$ as default BCF, but allowing sparsity across the other covariates. In addition to the information in Table 3.5, for a better visual

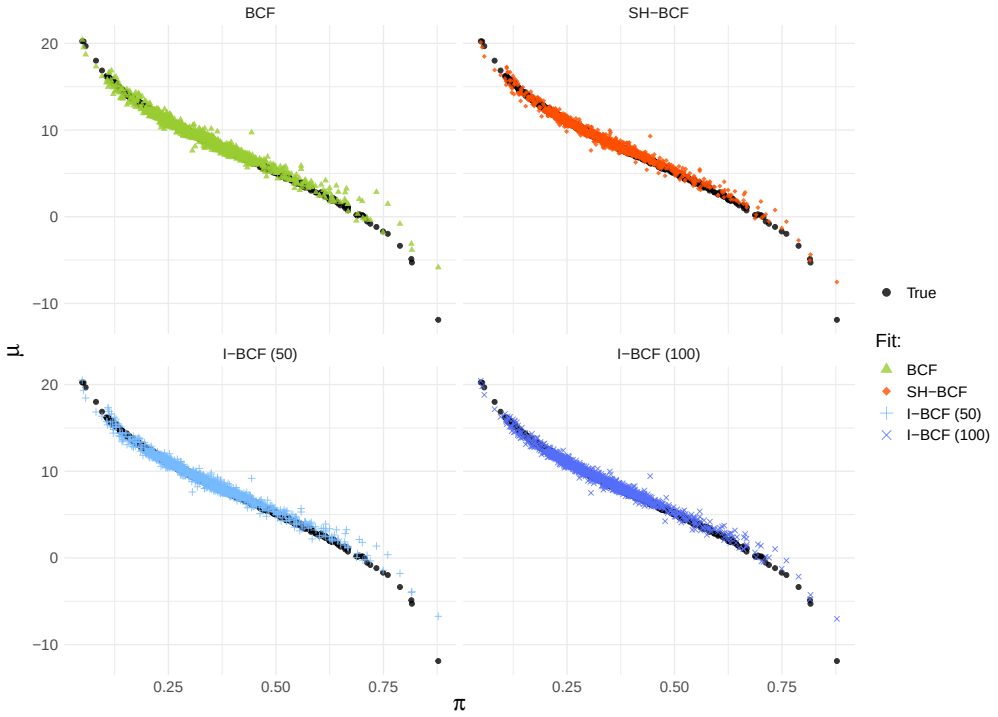


Figure 3.4: Posterior fit of $\pi(\cdot)$ and $\mu(\cdot)$ relationship, for default BCF, Shrinkage BCF (with $\pi(\cdot)$) and the two versions of informative prior BCF ($k_{PS} = 50$ and $k_{PS} = 100$). All the specifications effectively capture the underlying relationship.

inspection, we plot the posterior fit of the $\pi(\cdot)$ and $\mu(\cdot)$ relationship for each specification of BCF[2].

The results corroborate those of the previous sections, as all the Shrinkage BCF versions ii)-v) outperform default BCF i), thanks to their ability to adapt to sparsity (Table 3.5). In order to net out effects that are due to propensity model misspecification, we re-run the same example in (3.23) for $H = 250$, this time assuming that PS is known, thus plugging in the true values in $\mu(\boldsymbol{x}_i, \pi)$. Results can be found in the supplementary material.

The picture emerging from this exercise is the following. Methods ii)-v) all have comparable performances in the realistic scenario where PS is to be estimated (see Table 3.5); moreover, Figure 3.4 show that, in this case, they

[2]We avoid plotting the fit for iii) Shrinkage BCF without $\pi(\cdot)$, since it yields very similar results to ii) Shrinkage BCF with $\pi(\cdot)$ — In Table 3.5, ii) allocates nearly 0% splits to $\pi(\cdot)$, as in iii).

all effectively capture the relationship between $\pi(\cdot)$ and $\mu(\cdot)$. Hence, adjusting prior weights to nudge more splits on the estimated PS — methods iv) and v) — does not seem to improve performance. In the more abstract scenario where PS is assumed to be known (whose results are gathered in supplementary material), and thus the relationship between $\pi(\cdot)$ and $\mu(\cdot)$ can be directly estimated, versions i) and iii) perform poorly. The first because it does not induce sparsity, while iii) does not include $\pi(\cdot)$ as extra covariate. Versions ii), iv) and v) instead perform comparatively better as they virtually assign all the splitting probability to $\pi(\cdot)$, leaving the other covariates out of the model. This is unsurprising in a setup where $\pi(\cdot)$ is known, as its relationship with $\mu(\cdot)$ is straightforwardly captured. Even under this abstract scenario, specifications iv) and v), which assign higher weight to $\pi(\cdot)$, do not show improvements on performance, as also the agnostic prior version ii) effectively allocates the entire splitting probability to the $\pi(\cdot)$ covariate.

Results from the example where PS is perfectly known are in line with the findings of Hahn et al. (2020) and shed light on why adding $\pi(\cdot)$ as an extra covariate is always helpful in tackling targeted selection. Naturally, the success of this practice in addressing strong confounding heavily depends on the quality of the approximation of $\pi(\cdot)$, that is, the quality of the propensity model that estimates $\hat{\pi}(\cdot)$, and whether this is capable to account for unobserved confounders provided that there are enough good proxies among the observed covariates in practice.

3.3 Effects of early intervention on cognitive abilities in low weight infants

In this section, we illustrate the use of Shrinkage BCF by revisiting the study in Brooks-Gunn et al. (1992), which analyzes data from the Infant Health and Development Program (IHDP), found also in the previous chapter (although there we employed just the treatment assignment and the covariates set). As briefly mentioned in the last chapter, the IHDP was a randomized controlled trial

aimed at investigating the efficacy of educational and family support services, with pediatric follow-ups, in improving cognitive skills of low birth weight preterm infants, who are known to have developmental problems regarding visual-motor and receptive language skills (McCormick, 1985; McCormick et al., 1990). The study includes observations on 985 infants whose weight at birth was less than 2 500 grams, across 8 different sites. About one third of the infants were randomly assigned to treatment ($A_i = 1$), which consisted in routine pediatric follow-up (medical and developmental), in addition to frequent home visits to inform parents about child's progress and communicate instructions about recommended activities for the child. Following Hill (2011), the outcome variable (Y_i) we use is the score in a Stanford Binet IQ test, whose values can range from a minimum of 40 to a maximum of 160, taken at the end of the intervention period (child's age equal 3). The available final sample, obtained after removing 77 observations with missing IQ test score, consists of $n = 908$ data points, while the number of pre-treatment covariates amounts to $P = 31$. A full list of the variables included in the analysis, together with a short description, can be found in the supplementary material.

Firstly, we estimate propensity score using a 1-hidden layer neural network classifier. Then we run Shrinkage BCF with default agnostic prior for 15 000 MCMC iterations in total, but we discard the first 10 000 as burn-in. As output, we obtain the full posterior distribution on CATE estimates and splitting probabilities relative to each covariate. The left-hand pane graph of Figure 3.5 shows the estimated CATE posterior distribution for the individuals in the sample whose estimated propensity corresponds, or is closest, to the i-th percentile of the estimated propensity distribution, where i is 0, 10, 20, ..., 100. The represented stratified CATE posterior distribution relative to these propensity values conveys information about the uncertainty around the estimates and depicts an overall positive and rather heterogeneous treatment effects. The estimated average treatment effect is equal to ATE = 9.33 and standard deviation of CATE estimates, averaged over the post burn-in draws, is

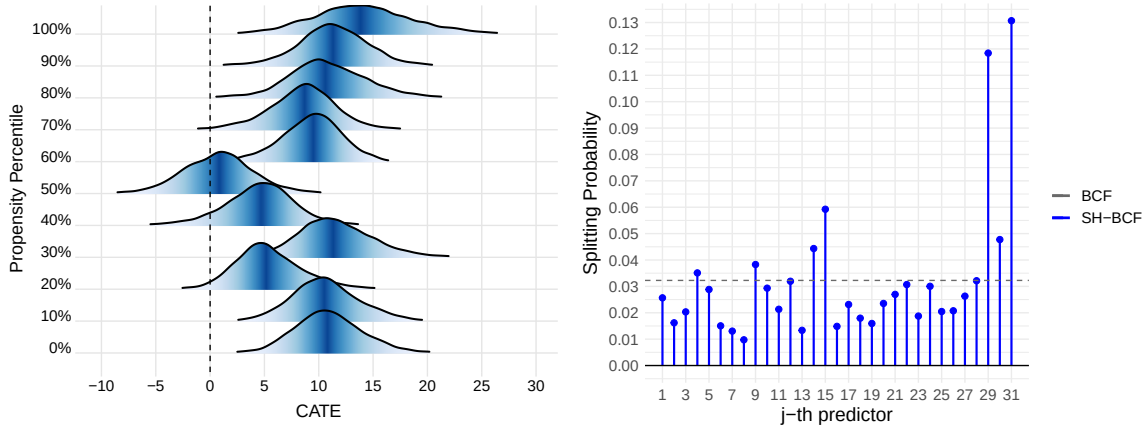


Figure 3.5: Left panel: Posterior distributions for the CATE estimates, obtained using Shrinkage BCF, corresponding to the approximated propensity percentiles (i.e. for individuals in the sample whose estimated propensity corresponds or is closest to the percentiles). Fill colour is darker around the median. Right panel: Shrinkage BCF's posterior splitting probabilities on $\tau(\cdot)$, averaged over the post burn-in MCMC draws.

equal to 3.25, which is another sign of underlying heterogeneity patterns in the treatment response. The analysis would thus benefit from further investigation about the heterogeneity of treatment effects, with the aim of distinguishing the impact within subgroups of individuals characterized by similar features (i.e. covariates values). Evidence on what the relevant drivers of heterogeneity behind treatment effect are is given by the posterior splitting probabilities on $\tau(\cdot)$ (again averaged over the post burn-in draws), reported in the right-hand pane graph of Figure 3.5, where few covariates end up being assigned relatively higher weights compared to the others. The two covariates that primarily stand out are the binary indicator on whether the mother's ethnicity is white (29^{th} predictor) and the ordinal variable indicating mother's level of education (31^{st} predictor).

We proceed with a sensitivity analysis of treatment effect subgroups by following the suggestion of Hahn et al. (2020); that is, we fit a decision tree partition algorithm using the `R` package `rpart`, by regressing mean CATE estimates obtained from Shrinkage BCF $\hat{\tau}(\boldsymbol{x}_i)$ (averaged over the MCMC

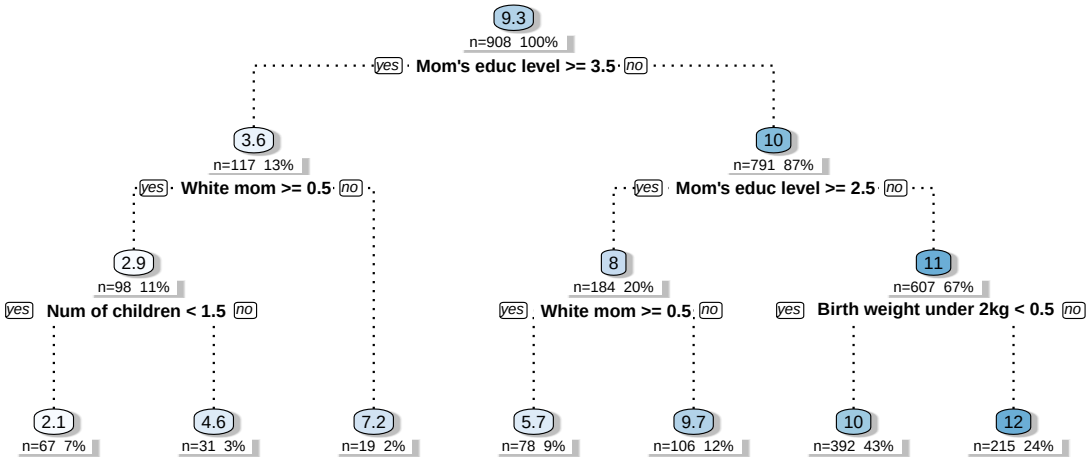


Figure 3.6: Decision tree identifying the most homogeneous subgroups in terms of treatment response, based on splitting rules involving the available covariates. The nodes report CATE estimates averaged within the corresponding subgroup.

post burn-in draws) on the available covariates $X_i \in \mathcal{X}$. The purpose of this sensitivity analysis exercise is to identify the most homogeneous subgroups, namely the subgroups leading to an optimal partition, in terms of their estimated mean CATE, as a function of the covariates, and to examine how much the emerging partition agrees with the results on posterior splitting probabilities in Figure 3.5.

Results are depicted in Figure 3.6 in the form of a decision tree, pruned at four levels. Zero splits trivially return ATE estimate (first node in Figure 3.6), while shallower nodes show CATE estimates averaged within the subgroup defined by the corresponding split rule. The first split is on the variable "Mother's level of education", specifically on whether the mother has attended college or not. The second level features a split on whether mother's ethnicity is white in one branch, and a split on whether mother has finished high school in the other. These are exactly the same covariates selected by the posterior splitting probabilities. The last set of splits is again on mother's ethnicity, number of children the mother has given birth to and whether child's birth

weight is less than 2kg. Within these subgroups, CATE estimates range from a minimum of +2.1 to a maximum of +12.

Both CATE's posterior splitting probabilities as well as subgroup analysis particularly point to covariates related to mother's education and ethnicity, in addition to birth weight (in the subgroup analysis only). Results concerning heterogeneity stemming from mother's ethnicity and child's birth weight are consistent with those in the original (Brooks-Gunn et al., 1992) and follow-up studies Brooks-Gunn et al. (1994); McCarton et al. (1997), where the treatment effect is found to be lower for white mothers and for children with lower weight. The advantage of carrying out subgroup analysis through models such as Shrinkage BCF lies in the fact that subgroup identification can be done ex-post using CATE estimates, without the need of manually identifying the groups or partitioning the original sample ex-ante.

This illustrative example showed how Shrinkage BCF detects covariates which are responsible for the heterogeneity behind treatment impact in an example of real-world analysis, and how simple a posteriori partitioning of CATE estimates allows the derivation of optimal splitting rules to identify the most homogeneous subgroups in terms of treatment response. The analysis demonstrated that the estimation of individual (or subgroup) effects is a key factor for the correct evaluation and design of treatment administration policies.

3.4 Interpretable Deep Causal Learning for Moderation Effects

We have seen in previous sections how Shrinkage BCF is a fully Bayesian model that jointly tackles the three crucial components of interpretability, targeted regularization and uncertainty quantification for individual treatment/causal effects estimation, and thus highly personalized policy-making. However, in Chapter 2 we have also discussed that properties of BCF and Shrinkage BCF, as τ-Learners, are also shared by R- type of Learners (Nie and Wager, 2020), since they use the same parametrization. In this second part of the chapter

we specifically seek to improve further on the interpretability component of moderating effects, by presenting a simple yet novel model for CATE learning based on an interpretable version of neural networks, rooted in the theory of Generalized Additive Models (GAMs).

3.4.1 Brief Overview of Feedforward Neural Networks

We begin by briefly outlining the main concepts in deep learning (Murphy, 2012; Goodfellow et al., 2016). Neural networks are a specific type of Adaptive Basis-Function Models (ABMs), introduced in Section 3.1.1, that model the conditional mean function of $Y \in \mathcal{Y}$ as a "concatenation" (or combination of) of both linear and non-linear functions of the inputs $\boldsymbol{X} \in \mathcal{X}$, that is:

$$f(\boldsymbol{X}) = \mathbb{E}[Y \mid \boldsymbol{X}] = \big(g_1 \circ ... \circ g_l\big)(\boldsymbol{X}) = g_l\big(... g_1(\boldsymbol{X})\big) \ . \tag{3.24}$$

The most basic type of neural net is also often called **feedforward neural networks**, as functions $\{g_j(\cdot)\}_{j=1}^{l}$ are applied sequentially at each "layer" to the previously obtained intermediate output $g_{j-1}(\cdot)$, starting from the "input layer", and do not feature feedback connections (as in e.g. Recurrent Neural Nets used for sequence data). The intermediate layers between the initial input layer $\boldsymbol{h}^{(0)} = \boldsymbol{X}$ and the final output layer $\boldsymbol{h}^{(l)} = f(\boldsymbol{X})$, identified by the output of the functions $\boldsymbol{h}^{(1)} = g_1(\boldsymbol{X})$ up to $\boldsymbol{h}^{(l-1)} = g_{l-1}(g_{l-2}(...))$, are referred to as **hidden layers**. The number of hidden layers, or length of chain of functions $\{g_j(\cdot)\}_{j=1}^{l}$, represents the depths of the neural network, while the dimensionality of a hidden layer $\boldsymbol{h}^{(j)}$ is the width.

The functions $\{g_j(\cdot)\}_{j=1}^{l}$ are usually broken down into a sequence of linear layers and non-linear **activation** functions, where at each stage previous layer's output is multiplied by a layer-specific **weight matrix** W_j and added a bias term $\boldsymbol{b}_j$, i.e. $W_j\boldsymbol{h}^{(j-1)} + \boldsymbol{b}_j$, then this is applied a non-linear transformation $g_j(\cdot)$ (activation function). Popular types of non-linear activation functions are the Sigmoid, Tanh, ReLU, Softplus, etc. The NN structure can be then

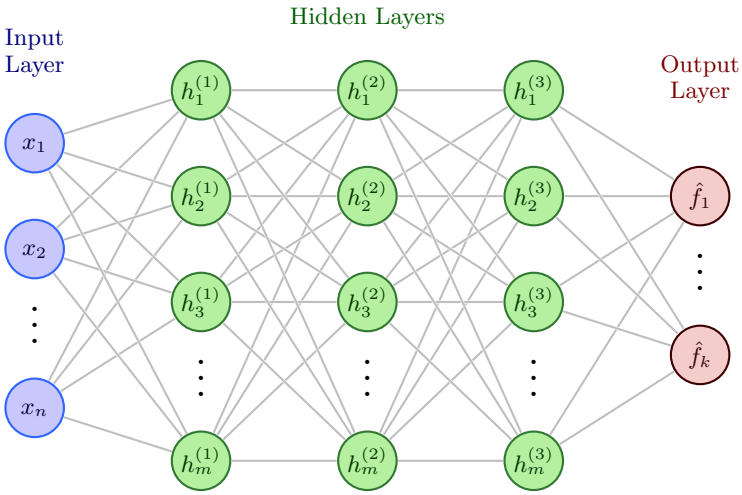


Figure 3.7: Intuitive graphical representation of a simple deep learning structure, where inputs are passed through 3 hidden layers of m width (m nodes), and mapped to a multi-dimensional vector outcome $\{\hat{Y}_j\}_{j=1}^{k}$ (regression or classification task).

represented as the following chain :

$$\begin{aligned}
\boldsymbol{h}^{(1)} &= g_1(W_1\boldsymbol{X} + \boldsymbol{b}_1) \\
&\vdots \\
\boldsymbol{h}^{(l-1)} &= g_{l-1}(W_{l-1}\boldsymbol{h}^{(l-2)} + \boldsymbol{b}_{l-1}) \\
f(\boldsymbol{X}) &= \mathbb{E}[Y \mid \boldsymbol{X}] = g_l(\boldsymbol{h}^{(l-1)}) \ .
\end{aligned}$$

Generally speaking, the depth, width and activation functions in a feedforward neural network control the complexity of the resulting class of non-linear functions $\mathcal{F}$ with which we are trying to approximate the target $f(\cdot)$. Thus, adjusting for different degrees of complexity requires ad-hoc adjustments to the neural network structure. Feedforward neural networks with a linear output layer and at least a hidden layer paired with a non-linear activation function have been proven to be universal approximators, i.e. can approximate any Borel measurable function $f \in \mathcal{B}(\boldsymbol{X})$, given the right amount of hidden nodes (Hornik et al., 1990). The parameter space in a neural network is composed by the collection of all the layer-specific weights and biases $\theta \stackrel{\text{def}}{=} \left\{\{W_j\}_{j=1}^{l}, \{\boldsymbol{b}_j\}_{j=1}^{l}\right\}$.

Figure 3.7 depicts a simple neural net structure as a graphical model.

3.4.1.1 Training of Neural Networks

In reasonably large (i.e. deep and wide) neural nets, parameters $\theta \overset{\text{def}}{=} \big\{\{W_j\}_{j=1}^{l}, \{\boldsymbol{b}_j\}_{j=1}^{l}\big\}$ are typically learned via **Stochastic Gradient Descent** (or generalizations of it such as RMSprop, ADAM, etc.), an iterative method that uses approximate gradient information to minimize a loss function $\mathcal{L}(\theta) = \mathbb{E}[\ell(\theta)]$. In standard Gradient Descent, the loss function is approximated by its sample average as $\hat{\mathcal{L}}(\theta) = \frac{1}{n_T}\sum_{i=1}^{n_T} \ell_i(\theta)$, where $\{1, ..., n_T\}$ is the training sample, and iterative updates to the parameters are computed as follows:

$$\text{Gradient Descent: } \theta_t \leftarrow \theta_{t-1} - \eta\big[\sum_{i=1}^{n_T} \nabla\ell_i(\theta)\big] \ , \quad \text{for } t \in \{1, ..., T\} \tag{3.25}$$

where $\nabla\ell(\cdot)$ is the gradient of ℓ, η is the learning rate and T are the number of iterations the algorithm is run for. The evaluation of the sum-gradient element in (3.25) can be very expensive if one needs to compute the gradient for many data points in the training set. In SGD, intense computations are avoided by updating the parameters using gradient information on only a sub-sample of observations. In particular, in SGD this is done by replacing sum-gradient with single observation gradient, so that the "online" update becomes $\theta_j \leftarrow \theta_{j-1} - \eta\nabla\ell_i(\theta)$, speeding up the computations significantly.

Gradient information, to be used in SGD, is acquired through a method known as **back-propagation**. Back-propagation computes the partial derivatives $\partial\ell/\partial W$ and $\partial\ell/\partial \boldsymbol{b}$ by leveraging the chain rule of differentiation to propagate the error backwards through the nodes in the network. For example, the differential loss with respect to $w_{i,j}^{(k)}$, i.e. the weight relative to layer k, output node i and input node j, can be computed as $\partial\ell/\partial w_{i,j}^{(k)} = \partial\ell/\partial h_j^{(k)} \times \partial h_j^{(k)}/\partial w_{i,j}^{(k)}$.

3.4.2 Interpretable Causal Neural Networks

We first raise the question on how we can straightforwardly use flexible deep learning models, described in earlier paragraphs, in the context of CATE estimation, in particular following Robinson (1988)'s parametrization, as we

proved this leads to several improvements, among which better interpretability. A very simple yet effective way of doing this, is to construct deep architecture made of two separable neural net blocks that respectively learn the prognostic function $\mu(\boldsymbol{x}_i)$ and the CATE function $\tau(\boldsymbol{x}_i)$, but are "reconnected" at the end of the pipeline to minimize a single loss function $\mathcal{L}_y(\cdot)$, unlike T-Learners which would instead minimize separate loss functions on $f_1(\cdot) \in \mathcal{F}_1$ and $f_0(\cdot) \in \mathcal{F}_0$. The target loss function to minimize is generally defined as follows:

$$\text{TCNN:} \quad \min_{\mu(\cdot),\tau(\cdot)} \mathcal{L}_y\big(\mu(\boldsymbol{x}) + \tau(\boldsymbol{x})a,\ y\big), \quad \mu(\cdot) \in \mathcal{M} \stackrel{\text{def}}{=} \mathcal{F}_0,\ \tau(\cdot) \in \mathcal{T} \tag{3.26}$$

where $\mathcal{L}_y(\cdot)$ can be any standard loss function (e.g., MSE, negative log-likelihood,...), inclusive of penalization (ℓ-1, ℓ-2,...). Note that $\mathcal{M} \stackrel{\text{def}}{=} \mathcal{F}_0$ as prognostic score $\mu(\cdot)$ coincide with $f_0(\cdot)$ in a T-Learner: $\mu(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i) = \mathbb{E}\big[Y_i \mid \boldsymbol{X}_i = \boldsymbol{x}_i, A_i = 0\big]$. Through its separable block structure, the model allows the design of different NN architectures for learning $\mu(\cdot) \in \mathcal{F}_0$ and $\tau(\cdot) \in \mathcal{T}$, which can be though of as "hardcoded" priors, while preserving sample efficiency (i.e., avoiding sample splitting as in T-Learners), and to produce uncertainty measures around CATE $\tau(\cdot) \in \mathcal{T}$ directly like τ-Learners if coupled with (approximate) Bayesian methods (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Pearce et al., 2020; Abdar et al., 2021). Thus, if $\tau(\cdot)$ is believed to display simple moderating patterns as a function of $\boldsymbol{X}_i$, a shallower NN structure with fewer nodes, and more aggressive regularization (e.g., higher regularization rate or dropout probabilities), can be specified, while retaining higher level of complexity in the $\mu(\cdot)$ block. We generally refer to this model as Targeted Causal Neural Network (TCNN) for simplicity from now onwards. Figure 3.8 provide a simple visual representation. While in this work we have been focussing on binary intervention variables A_i for simplicity, TCNN can be easily extended to multi-category A_i by adding extra blocks to the structure in Figure 3.8.

In addition to the separable structure, and in order to guarantee higher level of interpretability on prognostic and moderating factors, we also propose

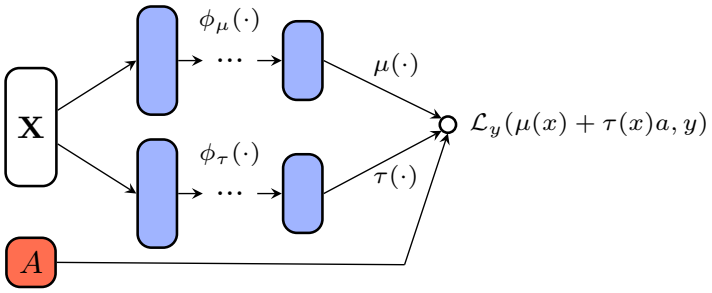


Figure 3.8: Intuitive TCNN structure. The deep architecture is modelled through a sample efficient, tailored loss function based on Robinson's parametrization.

using a recently developed neural network version of Generalized Additive Models (GAMs), named Neural Additive Models (NAMs) (Agarwal et al., 2021), as the two $\mu(\cdot)$ and $\tau(\cdot)$ NN building blocks of TCNN.

Neural Additive Models impose restrictions on the neural network structure to guarantee more interpretable output as a trade-off. In particular, they model the response function in $Y_i = g(\boldsymbol{x}_i) + \varepsilon_i$ as a Generalized Additive Model (GAM) (Hastie, 2017), a type of adaptive basis-function model that assumes additive separability of $g(\cdot)$ for each input $\boldsymbol{x}_i$:

$$Y_i = g(\boldsymbol{x}_i) + \varepsilon_i = \beta_0 + \sum_{j=1}^{P} f_j(\boldsymbol{x}_i) + \varepsilon_i \ , \quad \mathbb{E}(\varepsilon_i) = 0 \ \ \forall i \in \{1, ...n\},$$

where P is the total number of inputs $X \in \mathcal{X}$ in the model, so that $f_j(\cdot) \in \mathcal{F}_j$ is the j-th input-specific function. GAMs type of model have been shown, under common smoothness assumptions on each true $\{f_j^*(\cdot)\}_{j=1}^P$ and other mild conditions, to achieve optimal rates of convergence of the order of $\mathcal{O}(n^{-2r})$, where $r = \frac{\gamma}{2\gamma+P}$, for γ-smooth function $g(\cdot)$ (Stone, 1985). The input-specific functions $f_j(\cdot) \in \mathcal{F}_j$ also have an interpretation as Shapley values (Shapley, 1953; Agarwal et al., 2021), as they represent the single predictive contribution of input j to the total $g(\cdot)$ ("score functions"). The main difference in NAMs from other types of GAMs is in the training techniques, as NAMs are trained by minimizing e.g. a squared loss function through the usual combination of the back-propagation procedure and gradient descent algorithms.

Hence, we proceed by using NAMs as the building blocks to learn the $\mu(\cdot)$ and $\tau(\cdot)$ functions and refer to this particular version of TCNN as Interpretable Causal Neural Network (ICNN). Contrary to normal NNs, which fully "connect" inputs to every nodes in the first hidden layer, NAMs "connect" each single input to its own NN structure and thus outputs input-specific *score functions*, that fully describe the predicted relationship between each input and the outcome. The structure of the loss function (3.26) in ICNN thus becomes additive also in the P covariate-specific $\mu_j(\cdot)$ and $\tau_j(\cdot)$ functions:

$$\text{ICNN: } \min_{\mu(\cdot),\tau(\cdot)} \mathcal{L}_y\Big(\sum_{j=1}^{P} \mu_j(x_j) + \sum_{j=1}^{P} \tau_j(x_j)a,\, y\Big)\,, \quad \mu_j(\cdot) \in \mathcal{F}_{0,j},\ \tau_j(\cdot) \in \mathcal{T}_j$$

where the single $\mu_j(x_j)$ score function represents the Shapley value in terms of prognostic effect of covariate x_j, while $\tau_j(x_j)$ its Shapley value in terms of moderating effect. Hence, the NAM architecture in ICNN allows us to estimate the impact of each covariates as a prognostic and moderating factor and quantify the uncertainty around them as well. Under ICNN, the outcome function thus becomes twice additively separable:

$$Y_i = \sum_{j=1}^{P} \mu_j(x_{i,j}) + \sum_{j=1}^{P} \tau_j(x_{i,j})A_i + \varepsilon_i\,, \quad \mathbb{E}(\varepsilon_i) = 0 \quad \forall i \tag{3.27}$$

where $i \in \{1, ..., n\}$ and $j \in \{1, ..., P\}$. Naturally, the downside of NAMs is that they might miss out on interaction terms among the covariates as a trade-off losswith respect to interpretability gains. These could possibly be constructed and added manually as additional inputs, although this would increase the computational complexity of the model. To obtain better coverage properties in terms of uncertainty quantification in both TCNN and ICNN, we implement the Monte Carlo dropout technique (Srivastava et al., 2014; Gal and Ghahramani, 2016) in both $\mu(\cdot)$ and $\tau(\cdot)$ blocks to perform post-training resampling from the posterior predictive distribution, in am approximate Bayesian inference fashion.

Monte Carlo dropout , roughly speaking, consists in re-sampling a pre-trained neural network with dropout layers (Srivastava et al., 2014). Dropout is a neural nets regularization technique that aims at improving out-of-sample generalization and reducing complexity in over-parametrized neural nets by stochastically "dropping" nodes within each hidden layer $i \in \{1, ..., L\}$ with probability p_{drop}, during each training step. Probabilistic dropout re-sampling, if performed after training the neural network, generates an approximate sample $\{1, ..., T\}$ from the posterior predictive distribution $q(\cdot \mid \cdot)$ defined by the NN's parameters (Gal and Ghahramani, 2016), i.e. weights $\boldsymbol{\omega} = \{W\}_{i=1}^{L}$ and biases $\boldsymbol{b} = \{b\}_{i=1}^{L}$ where $i \in \{1, ..., L\}$, that is:

$$\big\{(\boldsymbol{W}_1, \boldsymbol{b}_1), ..., (\boldsymbol{W}_T, \boldsymbol{b}_T)\big\} \overset{\text{approx}}{\sim} q(\boldsymbol{y}^* \mid \boldsymbol{x}^*) = \int p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \boldsymbol{W}, \boldsymbol{b}) \, \mathrm{d}p(\boldsymbol{W}, \boldsymbol{b})$$

where $(\boldsymbol{W}_t, \boldsymbol{b}_t)$ are grouped over layers $\boldsymbol{W}_t = (W_t^1, ..., W_t^L)$, $\boldsymbol{b}_t = (b_t^1, ..., b_t^L)$, while $\boldsymbol{y}^*$ and $\boldsymbol{x}^*$ are test points.

MC dropout can then produce approximate Bayesian credible intervals around prognostic $\mu(\cdot)$ and CATE estimates $\tau(\cdot)$ in a very straightforward way, and, in ICNN specifically, credible intervals around each inputs' score functions $\mu_j(\cdot)$ and $\tau_j(\cdot)$ for $j \in \{1, ..., P\}$, as we will demonstrate in the experimental section below.

3.4.3 Simple Simulated Experiments

We hereby present results from a simple simulated experiment on CATE estimation, to compare TCNN and ICNN performance against other methods. In addition, we demonstrate how ICNN with MC dropout in particular can be employed to produce highly interpretable score function measures, fully describing the estimated moderating effects of the covariates $\boldsymbol{x}_i$ in $\tau(\cdot)$, and uncertainty around them. For performance comparison we rely on the root *Precision in Estimating Heterogeneous Treatment Effects* (PEHE) metric already, encountered in previous sections. The list of models we compare include: S-Learner version of NNs (S-NN); T-Learner version of NNs (T-NN); Causal Forest (Wager

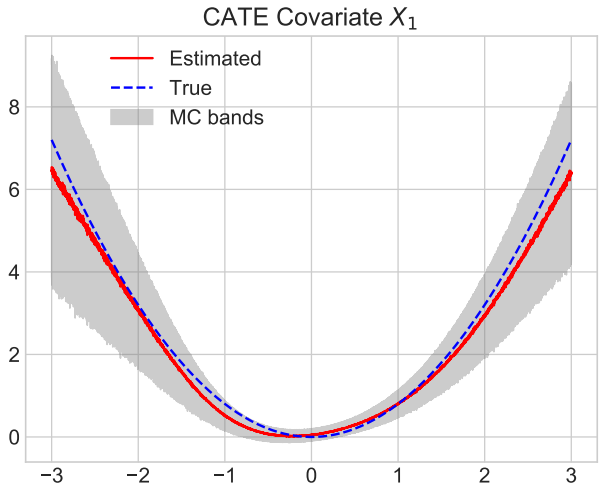


Figure 3.9: Score function output from ICNN model relative to covariate X_1, depicting its moderating effect on CATE, plus MC dropout generated credible intervals.

and Athey, 2018), a particular type of R-Learner (R-CF); a "unique-block", fully connected NN that uses Robinson's parametrization minimizing the loss function in (3.26) (R-NN); a "unique-block" NAM, again minimizing the loss function in (3.26) (R-NAM); our TCNN with fully connected NN blocks; and ICNN. S-NN, T-NN and R-NN all feature two [50, 50] hidden layers. R-NAM features two [20, 20] hidden layers for each input. TCNN features two [50, 50] hidden layers in the $\mu(\cdot)$ block, and one [20] hidden layer in the $\tau(\cdot)$ block. ICNN features two [20, 20] hidden layers in the $\mu(\cdot)$ block, and one [50] hidden layer in the $\tau(\cdot)$ block, for each input.

We simulate $n = 2000$ data points on $P = 10$ correlated covariates, with binary A_i and continuous Y_i. The experiment was run for $B = 100$ replications and results on 70%-30% train-test sets average $\sqrt{\text{PEHE}_\tau}$, plus 95% Monte Carlo errors, can be found in Table 3.6. The full description of the data generating process utilized for this simulated experiment can be found in the appendix. NN models minimizing the Robinson loss function in (3.26) perform considerably better than S- and T-Learner baselines on this particular example, especially TCNN and ICNN that present the additional advantage of conveying

Model	Train $\sqrt{\text{PEHE}_{\tau}}$	Test $\sqrt{\text{PEHE}_{\tau}}$
S-NN	1.046 ± 0.007	1.076 ± 0.007
T-NN	1.021 ± 0.002	1.074 ± 0.002
R-CF	1.467 ± 0.002	1.494 ± 0.002
R-NN	0.706 ± 0.003	0.712 ± 0.003
R-NAM	0.787 ± 0.002	0.787 ± 0.002
TCNN	**0.361 ± 0.001**	**0.362 ± 0.001**
ICNN	**0.328 ± 0.001**	**0.331 ± 0.001**

Table 3.6: Performance on simulated experiment, measured as 70%-30% train-test set $\sqrt{\text{PEHE}_{\tau}}$. Bold indicates better performance.

targeted regularization. Considering the ICNN model only, we can then access the score functions on the $\tau(\cdot)$ NAM block that describe the moderating effects of the covariates $\boldsymbol{x}_i$. In particular in Figure 3.9 we plot the score function of the first covariate $X_{i,1}$ on CATE $\tau(\cdot)$, plus the approximate Bayesian credible intervals generated through MC dropout resampling (Gal and Ghahramani, 2016). In this specific simulated example, CATE function is generated as $\tau(\boldsymbol{x}_i) = 3 + 0.8X_{i,1}^2$. So only $X_{i,1}$, out of all $P = 10$ covariates, drives the simple heterogeneity patterns in treatment response across individuals, in a quadratic form. As Figure 3.9 shows, ICNN is able in this example to learn a score function that very closely approximates the underlying true relationship $0.8X_{i,1}^2$, and quantifies uncertainty around it. Naturally, in a different simulated setup with strong interaction terms among the covariates, performance of ICNN would inevitably deteriorate compared to the other versions of NN and models considered here. Thus, performance and interpretability in this type of scenario would certainly constitute a trade-off.

3.4.4 Real-World Example: the ACTG-175 data

Finally, we briefly demonstrate the use of ICNN on a real-world example. Although the focus of the work so far has been on observational type of studies, we will analyze data from a randomized experiment to show that the methods introduced naturally extend to this setting as well, with the non-

negligible additional benefit that both *unconfoundedness* and *common support* assumptions hold by construction (i.e., no "causal" arrow going from $\boldsymbol{X} \rightarrow A$). The data we use are taken from the ACTG-175 study, a randomized controlled trial comparing standard mono-therapy against a combination of therapies in the treatment of HIV-1-infected patients with CD4 cell counts between 200 and 500. Details of the design can be found in the original contribution by Hammer et al. (1996). The dataset features $n = 2139$ observations and $P = 12$ covariates $\boldsymbol{X}$ (which are listed in the appendix section below), a binary treatment A (mono-therapy VS multi-therapy) and a continuous outcome Y (difference in CD4 cell counts between baseline and after 20 ± 5 weeks after undertaking the treatment — this is done in order to take into account any individual unobserved time pattern in the CD4 cell count).

The aim is to investigate the moderation effects of the covariates in terms of heterogeneity of treatment across patients. In order to do so, we run ICNN and obtain the estimated score functions, together with approximated Bayesian MC dropout bands, for each covariate X_j, and we report these in Figure 3.10. The results generally suggest a good degree of treatement heterogeneity, with most of the covariates playing a significant moderating role.

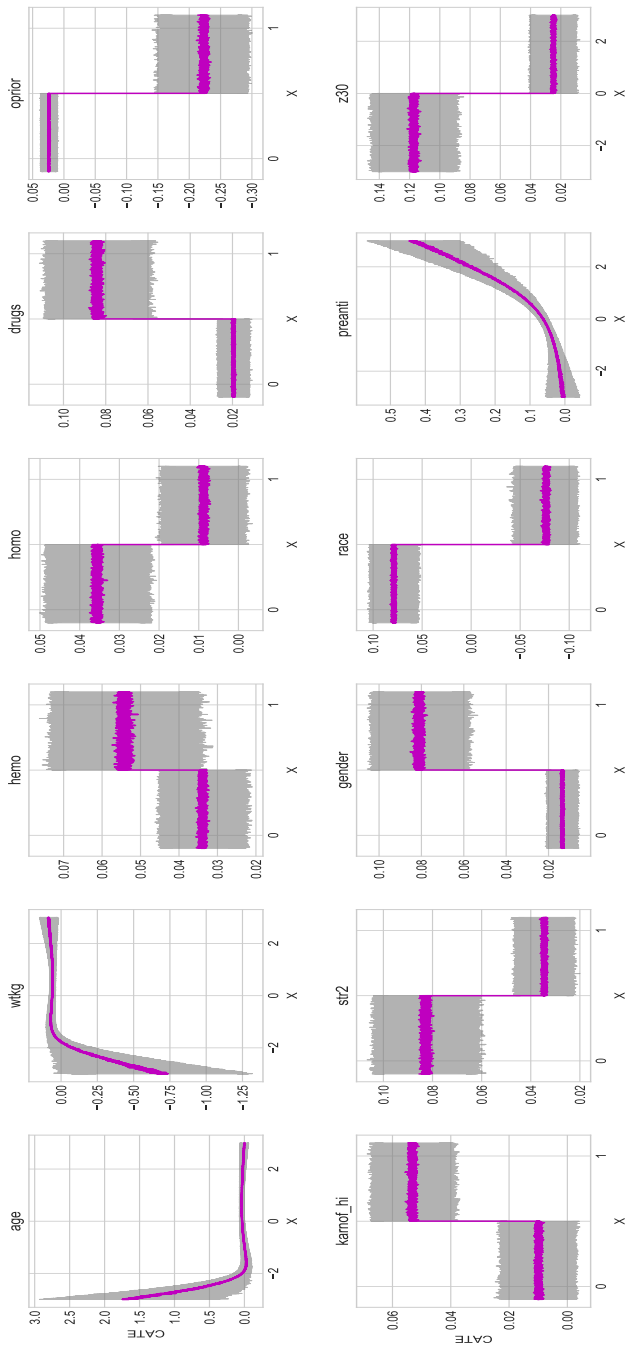


Figure 3.10: Estimated score functions, with associated MC dropout bands, describing moderation effects of each covariate x_j on CATE, i.e. $\tau_j(x_j)$, $\forall j \in \{1, ..., P\}$. All the $P = 12$ covariates in the ACTG-175 data included in the moderation analysis are described in Table 2.4.

Chapter 4

Scalable Bayesian Causal Learning for Multi-Action and Multi-Outcomes Settings

Contributions *The contents in this chapter are relative to a paper submitted and accepted to the Transaction on Machine Learning Research journal. Reference: Caron et al. (2022d). In addition, we include a discussion on the links with the tangential topics of Reinforcement Learning.*

So far in this work we have considered settings where the action or treatment space is limited to a discrete binary space $\mathcal{A} = \{0, 1\}$ and the outcome is continuous and unique $Y_i \in \mathcal{Y} = \mathbb{R}$, while the covariate space $\boldsymbol{X}_i \in \mathcal{X}$ is high-dimensional. However, studies with multiple discrete actions $\mathcal{A} = \{0, 1\}^d$ and multiple outcomes are quite common in applied research, as policy decisions are rarely based on a single outcome, but rather on a profile of different outcomes that might exhibit positive or negative correlation. As an example, in the medical domain, prescription of a specific treatment, such as anti-coagulants to prevent the risk of blood clots forming, affects both the risk of myocardial infarction (primary outcome), as well as the risk of bleeding (unwanted, correlated, side effect).

In this chapter we introduce a scalable Bayesian causal inference method based on multi-task Deep Kernel Learning (DKL), a computational surrogate of Gaussian Process regression that scales better with dimensions, to tackle such scenarios characterized by high dimensionality along multiple "axes", i.e. actions, outcomes and covariates. We start by briefly formalizing the problem by highlighting also connection with Model-Based (Offline) Reinforcement Learning, as several applications in this topic typically present multi-arm discrete action spaces. We then proceed by reviewing more in details the multi-task Gaussian Process paradigm in causal learning (Alaa and van der Schaar, 2017, 2018), already encountered in Chapter 2, and how this can in theory be easily extended to multi-actions and multi-outcomes scenarios. Finally, we discuss how multi-task GP struggle in high-dimensions along multiple axis, and offer a scalable, yet Bayesian, solution represented by multi-task Deep Kernel Learning models.

4.1 Problem Framework

Suppose we have access to an observational dataset $\mathcal{D}_i = \{\boldsymbol{X}_i, \boldsymbol{A}_i, \boldsymbol{Y}_i\} \sim p(\cdot)$, with $i \in \{1, ..., n\}$, where $\boldsymbol{X}_i \in \mathcal{X}$ is a set of covariates, $\boldsymbol{A}_i \in \mathcal{A} = \{0, 1\}^d$ a set of discrete, mutually exclusive, actions, and $\boldsymbol{Y}_i = \{Y_{i,j}\}_{j=1}^M \in \mathbb{R}^M$ a set of M different outcomes. For ease of notation, we will consider the "condensed" version of the vector of categorical actions A_i instead of the one-hot encoded one $\boldsymbol{A}_i \in \mathcal{A} = \{0, 1\}^d$. Also, we begin by considering continuous type of outcomes for simplicity, but we extend the methods also to discrete outcomes (Milios et al., 2018), as in the experimental Section 4.4.2. As always, the goal is to identify and estimate the effects of intervening on A_i, by setting it equal to some value a, on the M outcomes $\boldsymbol{Y}_i$. Using the *do*-calculus notation of Pearl (2009a) for simplicity in working with multiple actions and outcomes, more in details we are interested in Bayesian inference on the interventional multivariate distribution $p(\boldsymbol{Y} \mid do(A = a))$. We assume that the SCM is fully

described by the following set of equations:

$$\begin{aligned} \boldsymbol{X}_i &= f_X(\varepsilon_{i,X}), \\ A_i &= f_A\big(pa(A_i), \varepsilon_{i,A}\big) = f_A(\boldsymbol{X}_i, \varepsilon_{i,A}) \\ \boldsymbol{Y}_i &= \boldsymbol{f}_Y\big(pa(\boldsymbol{Y}_i), \varepsilon_{i,Y}\big) = \boldsymbol{f}_Y(\boldsymbol{X}_i, A_i, \varepsilon_{i,Y}) \ , \end{aligned} \tag{4.1}$$

where $pa(A_i)$ denotes parent variables (or direct causes) of A_i; $\varepsilon_{i,j}$ are error terms with a distribution $p(\varepsilon_{i,j})$. We make the two standard assumptions for identification of the causal effect $A \rightarrow \boldsymbol{Y}$ in this scenario, that is *unconfoundedness*, that must hold for all the M outcomes $\{Y_{i,j}\}_{j=1}^{M}$, and *overlap*, $0 < p(A_i = a|\boldsymbol{X}_i = x) < 1$, $\forall a \in \mathcal{A}$. Remember that violation of overlap for portions of $\mathcal{X}$ undermines generalization and extrapolation of the causal model's prediction in those regions. Under these two assumptions, the multivariate interventional distribution $p(\boldsymbol{Y} \mid do(A = a), \boldsymbol{X} = \boldsymbol{x})$ can be recovered via *backdoor adjustment* as $p(\boldsymbol{Y} \mid do(A = a), \boldsymbol{X} = \boldsymbol{x}) = p(\boldsymbol{Y} \mid A = a, \boldsymbol{X} = \boldsymbol{x})$ (Pearl, 2009a).

4.1.1 Connections to Reinforcement Learning

Generally speaking, the problem of causal effects learning is closely related to the Reinforcement Learning literature (Sutton and Barto, 2018). We make this connection only at this point of the work because Reinforcement Learning problems typically feature multi-action scenarios. Loosely speaking, the main idea in Reinforcement Learning is that an agent faces a dynamic programming problem, where it sequentially interact with an environment at each time $t \in \{1, ..., T\}$, whose information is summarized in a state $S_t \in \mathcal{S}$, and decides on a variety of actions to be taken at each state $A_t \in \mathcal{A}$. The combination of state-action at time t sends the agent to a new state S_{t+1} through a trajectory function $f_s : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$, and generate also a reward $R_{t+1} \in \mathcal{R}$ through a reward function $f_r : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{R}$. Generally the dynamics of this system is assumed to be Markovian, and can be thus described by a **Markov Decision Process** (MDP). An MDP is a tuple consisting of $\langle \mathcal{S}, \mathcal{A}, p_s, f_r, \gamma \rangle$, where:

1) $\mathcal{S}$ is the state set
2) $\mathcal{A}$ is the action set
3) $p_s(s' \mid S_t = s, A_t = a)$ is the stochastic transition probability
4) $f_r : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{R}$ is the reward function generating $r_t = f_r(s, a) = \mathbb{E}[R_t \mid S_t = s, A_t = a]$
5) $\gamma \in [0, 1]$ is a discount factor

The ultimate goal of the agent is to find an optimal **policy**, defined as a probabilistic mapping $\pi : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ between the current state and the action space $\pi(A_t = a \mid S_t = s)$, that maximizes the expected **returns**, i.e. the expected value of the sum of all discounted rewards $G_t = \sum_{k=1}^{T-t-1} \gamma^k R_{t+k+1}$. A policy $\pi^* \in \Pi$ is said to be optimal if it maximizes the **state value function** $\mathcal{V}_\pi(s)$ or equivalently the **state-action value function** $q_\pi(s, a)$, defined as:

$$\texttt{State:} \quad \mathcal{V}_\pi(s) = \mathbb{E}_{a\sim\pi}[G_t \mid S_t = s]$$
$$\texttt{State-action:} \quad q_\pi(s, a) = \mathbb{E}_{a\sim\pi}[G_t \mid S_t = s, A_t = a]$$

where $\mathcal{V}_\pi(s) = \mathbb{E}_{\mathcal{A}}[q_\pi(s, a)]$, obtained by marginalizing on a. Thus, the optimal policy satisfies $\pi^* = \arg\max_{\pi\in\Pi} \mathcal{V}_\pi(s)$, or equivalently $\pi^* = \arg\max_{\pi\in\Pi} q_\pi(s, a)$. In typical RL settings, the agent usually learns π^* "online", in the sense of active learning, with data collection operated by the agent happening on the fly. In the more narrow, and more closely related, sub-topic of Offline Reinforcement Learning (which includes e.g. the case of Dynamic Treatment Regimes, where rewards are sparse) instead, one has access to some historical (observational) data $\mathcal{D}_t = \{\boldsymbol{X}_t \overset{def}{=} \boldsymbol{S}_t, A_t, R_t\}_{t=1}^T$, where $\boldsymbol{X}_t \overset{def}{=} \boldsymbol{S}_t \in \mathcal{X}$ denotes the environment/state covariates, while $A_t \in \mathcal{A}$ and $R_t \in \mathbb{R}$ are the usual action and reward/outcome spaces. These historical data relate to a logged "behaviour policy" implemented previously.

It is easily noticeable that RL inherently involves elements of causality and counterfactual reasoning about actions to be played, and reward they consequently generate. It can be straightforwardly shown how the settings described above links back to estimation of causal effects and CATE in particular, with

the non-negligible difference that we specifically work under the assumption that data $\mathcal{D}_i$ are i.i.d., which is often referred to as **contextual bandits** setting in the RL literature, while general RL works in dynamic/sequential sampling settings (data are i.i.d. only conditional on a "time slice"). If we use Potential Outcome $\boldsymbol{Y}^{(a)}$ to denote the collection of trajectory's counterfactual outcomes corresponding to a played action $A_i = a$, i.e. $\{Y_1^{(a)}, ..., Y_T^{(a)}\}$, under the doubleton $\mathcal{A} = \{0, 1\}$, one can define an equivalent state value function generated by a policy $\pi \in \Pi$ and the state/covariate set $\boldsymbol{X} \in \mathcal{X}$ as

$$\mathcal{V}_{\pi\in\Pi}(\boldsymbol{X}) = \mathbb{E}\big[\boldsymbol{Y}^{(1)}\,\pi(A = 1|\boldsymbol{X}) + \boldsymbol{Y}^{(0)}\,\pi(A = 0|\boldsymbol{X})\big]\ ,$$

which is in turn equivalent to writing $\mathcal{V}_{\pi\in\Pi}(\boldsymbol{X}) = \mathbb{E}\big[\tau(\boldsymbol{x})\,\pi(A = 1|\boldsymbol{X})\big]$ where $\tau(\boldsymbol{x}) = \mathbb{E}[Y^{(1)} - Y^{(0)}|\boldsymbol{X}]$ is the CATE. In RL, one can generally distinguish between two types of policy tasks: i) **Off-Policy Evaluation** (OPE), where the goal is to evaluate a given policy $\pi_e \in \Pi$ in terms of the value it generates; ii) **Off-Policy Learning**, where the goal is to derive the optimal policy $\pi^* = \arg\max_{\pi\in\Pi} \mathcal{V}_{\pi\in\Pi}(\boldsymbol{X})$, defined as above. The term "Off-Policy" here does not refer to whether the task is an active type of learning, but rather to the fact that, in an online setting, data are not being collected by the agent under a certain given policy regime which generates the actions and the new states according to $A_t \sim \pi_\beta(\cdot|\cdot)$. In policy learning we generally seek policy estimators that give guarantees in terms of **policy regret**, defined as

$$\mathcal{R}(\hat{\pi}) = \mathcal{V}^*(\pi) - \mathbb{E}_{\mathcal{P}}\big[\mathcal{V}(\hat{\pi})\big] = \mathbb{E}_{\mathcal{P}}\big[\mathcal{V}^*(\pi) - \mathcal{V}(\hat{\pi})\big] \geq 0\ , \tag{4.2}$$

for the OPL task, where $\mathcal{V}^*(\pi_p) = \sup_{\pi_p\in\Pi} \mathcal{V}(\pi_p)$. A very quick overview of some of the relevant contributions in the related field of OPE and OPL include the relatively early work of Qian and Murphy (2011), where theoretical guarantees for direct regression-adjustment methods are provided. Dudík et al. (2011); Zhang et al. (2012) first propose a classical doubly-robust approach to the problem, while Kallus (2018) improves on the way the weights are

derived. Zhao et al. (2012); Zhou et al. (2017) instead view OPL as a particular case or classification problem and focus on developing a direct (non plug-in) method aimed at directly maximizing a convex surrogate of the non-convex classification loss. Eventually, Kitagawa and Tetenov (2018), Athey and Wager (2021) and Kallus (2021) address the problem of OPL under constrained policy class $\Pi_0 \subseteq \Pi$.

4.2 Counterfactual Learning with Multitask Gaussian Processes

For ease of exposition, consider the simple case with a single continuous outcome $Y_i \in \mathbb{R}$, with $i \in \{1, ..., n\}$. We tackle the problem of estimating $p(Y|do(A = a))$ via non-linear regression-adjustment (Johansson et al., 2016; Shalit et al., 2017; Künzel et al., 2017; Nie and Wager, 2020; Caron et al., 2022a). As in earlier chapters, we assume the additive noise structural model:

$$Y_i = f_Y(\boldsymbol{X}_i, A_i) + \varepsilon_{i,Y} \ , \quad \mathbb{E}(\varepsilon_{i,Y}) = 0 \, . \tag{4.3}$$

As seen in Chapter 2, there are different ways in which one can derive an estimator for $p\left(Y \mid do(A = a)\right)$ and its moments, e.g. $\mathbb{E}[Y \mid do(A = a)]$. Alaa and van der Schaar (2017, 2018) first proposed the use of multitask learning via Gaussian Process regression, in the specific context of conditional average treatment effects estimation, which is defined, assuming binary $A_i \in \{0, 1\}$, as the quantity $\tau(\boldsymbol{x}_i) = \mathbb{E}[Y_i|do(A_i = 1), \boldsymbol{x}_i] - \mathbb{E}[Y_i|do(A_i = 0), \boldsymbol{x}_i]$. The idea behind causal multitask GPs is to view the $D = |\mathcal{A}|$ interventional quantities Y_a, where D is the number of discrete action arms, as the output from a vector-valued function $\boldsymbol{f}_Y(\cdot) : \mathcal{X} \mapsto \mathbb{R}^D$ (plus noise), modelled with a GP prior:

$$\boldsymbol{f}_Y(\cdot) \sim \mathcal{GP}\Big(\boldsymbol{m}(\cdot), K(\cdot, \cdot)\Big) \ , \tag{4.4}$$

with mean $\boldsymbol{m}(\boldsymbol{x}_i) \in \mathbb{R}^D$ and covariance/kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbb{R}^D \times \mathbb{R}^D$, given two P-dimensional input points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$ for units i and j. Given the

likelihood function as a multivariate Gaussian $p(\boldsymbol{y}_i|\boldsymbol{f}_Y, \boldsymbol{x}_i, \Sigma) \triangleq \mathcal{N}(\boldsymbol{f}_Y(\boldsymbol{x}_i), \Sigma)$, where $\Sigma \in \mathbb{R}^D \times \mathbb{R}^D$ is the error covariance diagonal matrix with $\{\sigma_a^2\}_{a=1}^D$ on the diagonal and $\boldsymbol{y}_i \in \mathbb{R}^D$ an output point, the posterior predictive distribution for a train set covariate realization $\boldsymbol{x}_i \in \mathcal{X}$, train set outcome realization $\boldsymbol{y}_i \in \mathbb{R}$ and a test set covariate realization $\boldsymbol{x}_j^* \in \mathcal{X}$ is obtained as, assuming zero prior mean $\boldsymbol{m}(\cdot) = \boldsymbol{0}$ for simplicity:

$$
\begin{gathered}
p\big(\boldsymbol{f}_Y(\boldsymbol{x}_j^*) \mid (\boldsymbol{x}_i, \boldsymbol{y}_i), \boldsymbol{f}_Y, \phi\big) \triangleq \mathcal{N}\big(\boldsymbol{f}_Y^*(\boldsymbol{x}_j^*), K^*(\boldsymbol{x}_j^*, \boldsymbol{x}_j^*)\big) \ , \\
\boldsymbol{f}_Y^*(\boldsymbol{x}_j^*) = K(\boldsymbol{x}_j^*, \boldsymbol{x}_i) H \boldsymbol{y} \ , \quad K^*(\boldsymbol{x}_j^*, \boldsymbol{x}_j^*) = K(\boldsymbol{x}_j^*, \boldsymbol{x}_j^*) - K(\boldsymbol{x}_j^*, \boldsymbol{x}_i) H K^\top(\boldsymbol{x}_j^*, \boldsymbol{x}_i) \ , \\
\text{where} \quad H = \Big[K(\boldsymbol{x}_i, \boldsymbol{x}_i) + \Sigma\Big]^{-1} \ ,
\end{gathered}
\tag{4.5}
$$

and where ϕ denotes the model parameters and $\boldsymbol{f}_Y^*(\boldsymbol{x}_j^*)$ the function evaluated at a test point $\boldsymbol{x}_j^*$. Under zero prior mean $\boldsymbol{m}(\cdot) = \boldsymbol{0}$, the multitask GP in (4.4) is fully parametrized by its kernel function $K(\cdot, \cdot)$. The structure of the kernel function in a multitask GP is what induces task-relatedness when fitting the multi-valued surface $\boldsymbol{f}_Y(\cdot)$.

4.2.1 The multitask kernel

The simplest specification for the multitask kernel matrix is given by the *separable kernels* structure, which assumes single entries in $K(\cdot, \cdot)$ to be of the form $k_{a,a'}(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)\, k_A(a, a') = k(\boldsymbol{x}_i, \boldsymbol{x}_j)\, b_{a,a'}$, with action $a \in \mathcal{A} = \{1, ..., D\}$. Here, $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ represents a base kernel (e.g. linear, squared exponential, Matérn, etc.) while $b_{a,a'} = k_A(a, a')$ is the generic entry of the $D \times D$ **coregionalization matrix** B, which contains the parameters governing task-relatedness over the actions A. In the trivial case where $b_{a,a'} = 0$ we have that tasks a and a' are uncorrelated, i.e. actions a and a' are unrelated in the way they affect the outcome Y.

A slightly more general framework, which we adopt in this work, is given by the *sum of separable kernels* structure (Alvarez et al., 2012). This assumes that the single entry of $K(\cdot, \cdot)$ reads $k_{a,a'}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{q=1}^Q B_q\, k_q(\boldsymbol{x}_i, \boldsymbol{x}_j)$, i.e. the sum of Q different coregionalization matrices B_q with associated base kernel $k_q(\cdot, \cdot)$. In

compact matrix notation this translates into $K(X, X') = \sum_{q=1}^{Q} B_q \otimes K_q(X, X')$ for two different input matrices $X, X' \in \mathcal{X}$, where $\otimes$ is the Kronecker product. Imposing a *sum of separable kernels* structure is equivalent to assuming that the collection of action-specific functions $\{f_{Y_a}(\cdot)\}_{a=1}^{D}$ generates from $Q \leq D$ common underlying independent latent GP functions $\{u_q(\cdot)\}_{q=1}^{Q}$, parametrized by their base kernel $k_q(\cdot, \cdot)$, that is $\text{cov}(u_q(\boldsymbol{x}_i), u_{q'}(\boldsymbol{x}_j)) = k_q(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (Alvarez et al., 2012).

In terms of the form of the coregionalization matrices B_q, with $q \in \{1, ..., Q\}$, we will follow the **linear model of coregionalization** (LMC), which assumes that each B_q is equal to $B_q = L_q L_q'$, with single entries $b_{q;\,(a,a')} = \sum_{r=1}^{R_q} \alpha_{a,q}^{r} \alpha_{a',q}^{r}$. R_q represents the number of GP samples obtained from the same latent GP function q, $u_q(\cdot)$. Thus, adopting the LMC for causal learning is equivalent to assuming that correlation in the $\{f_{Y_a}(\cdot)\}_{a=1}^{D}$ action-specific functions, modelled through the multitask kernel $K(\cdot, \cdot)$, arises from Q different samples of R_q GP functions with the same kernel $k_q(\cdot, \cdot)$, drawn from $Q \leq D$ different independent latent GP processes $\{u_q(\cdot)\}_{q=1}^{Q}$. To express this more compactly, a causal multitask GP model under the LMC reads $\boldsymbol{f}_Y(\cdot) \sim \mathcal{GP}\big(\boldsymbol{m}(\cdot), K(\cdot, \cdot)\big)$, with single entries of $K(\cdot, \cdot)$ being

$$
\begin{aligned}
k_{a,a'}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{q=1}^{Q} B_q \, k_q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{q=1}^{Q} A_q A_q' \, k_q(\boldsymbol{x}_i, \boldsymbol{x}_j) \,, \quad \text{implying} \\
\text{cov}\big(f_a(\boldsymbol{x}_i), f_{a'}(\boldsymbol{x}_j)\big) = \sum_{q,q'}^{Q} \sum_{r,r'}^{R_q} \alpha_{a,q}^{r} \alpha_{a',q'}^{r'} \, \text{cov}\big(u_q^{r}(\boldsymbol{x}_i), u_{q'}^{r'}(\boldsymbol{x}_j)\big) \,.
\end{aligned}
\tag{4.6}
$$

In our specific case, as in Alaa and van der Schaar (2018), we employ a special case of LMC, named **intrinsic coregionalization model** (ICM) (Bonilla et al., 2008), where the underlying latent GP function is unique ($Q = 1$), so that $k_{a,a'}(\boldsymbol{x}_i, \boldsymbol{x}_j) = B \, \tilde{K}(\boldsymbol{x}_i, \boldsymbol{x}_i')$, with unique base kernel $\tilde{K}(\cdot, \cdot)$. The ICM specification attempts to avoid severe parameter proliferation in high-dimensional settings with multiple correlated actions $D = |\mathcal{A}|$, while still being capable of capturing task-relatedness through the relatively simple

structure of B. However, beside the issue of parameter proliferation when $\mathcal{A}$ features multiple discrete actions, exact GP regression is also known to scale poorly with sample size and cardinality of input space $|\mathcal{X}|$, and direct likelihood maximization methods face issues in over-parametrized models, although some solutions, such as variational methods (Titsias, 2009; Hensman et al., 2013), might be adopted for better scalability.

4.2.2 Why multitask counterfactual learning?

We know that asymptotically under no sample selection bias between action/treatment groups as in a randomized experiments, the best approach to estimate the causal quantities $\boldsymbol{f}_Y(\boldsymbol{x}_i)$ would be a "T-Learner", which implies splitting the sample and fitting separate models for each arm $A = a$, $\hat{f}(\mathbf{x})_a$. However, in finite samples flawed by selection bias, this is very often not the best strategy as we have seen in earlier chapters. By simply extending results on optimal min-max rates of convergence in Alaa and van der Schaar (2018), one can show that with increasing action and covariates spaces at the same time the problem of estimating causal effects becomes increasingly harder. For these reasons, a multitask GP prior over the actions is well suited to tackle selection bias and particularly estimation in regions with poor overlap, i.e. regions in $\mathcal{X}$ where we mainly observe data points with specific action $A_i = a$ and very few others. In addition to this, as shown by Alaa and van der Schaar (2018), a multitask GP prior can achieve the optimal minimax rate of Corollary 3.1 in its posterior contraction rates.

Hence, splitting the sample into n_a subgroups and fitting independent models can be very sample inefficient in these settings. Multitask GPs can aid extrapolation in such cases of strong sample selection bias, by learning the correlated functions $\{f_{Y_a}(\cdot)\}_{a=1}^{D}$ jointly as $\boldsymbol{f}_Y(\cdot)$. The first row plots in Figure 4.1 provide a very simple one-covariate example of how multitask learning addresses the issue of extrapolation and prediction in poor overlap regions. Fitting the two surfaces $f_{Y1}(x_i)$ and $f_{Y0}(x_i)$ (dashed lines) through separate GP regressions results in a bad fit out of overlap regions (top-left plot) in this

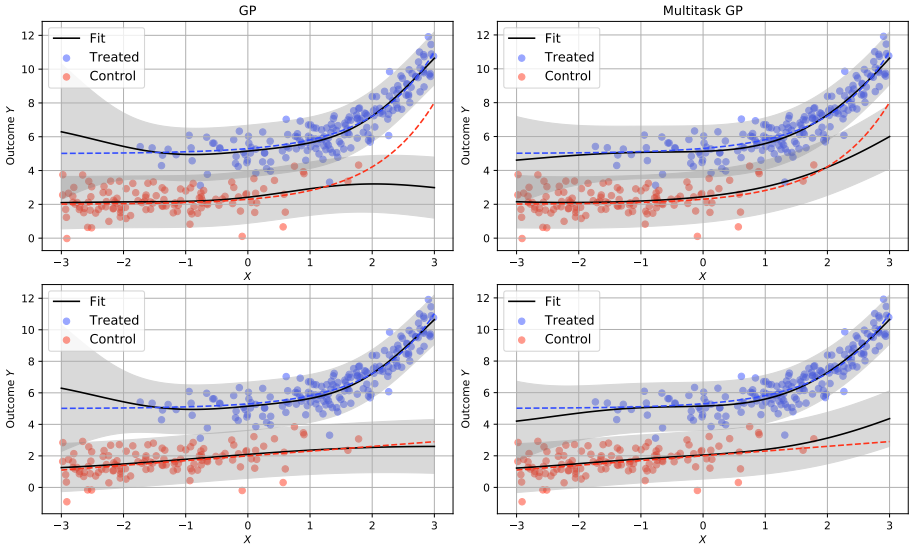


Figure 4.1: Simple one covariate example, with $\mathcal{A} = \{0, 1\}$. Overlap is guaranteed to hold over the whole support $\mathcal{X}$ in the data generating process, i.e. every unit has non-zero probability of being assigned to either $A_i = 1$ or $A_i = 0$, but $p(A_i = 1|X_i)$ is generated as an increasing function of X_i (selection bias). In the top row simulation, the two underlying counterfactual surfaces $f_{Y_d}(x_i)$ (dashed lines) are generated with very similar patterns, thus GP (left panel) is unable to borrow information from the other arm in poor overlap regions contrary to multitask GP (right panel). In the bottom row simulation instead we generate less similar surfaces, so borrowing of information through multitask GP does not lead to any improvement.

specific case. Multitask coregionalized GP attempts to fix this problem by embedding the assumption that the two surfaces share similar patterns via joint estimation of $\boldsymbol{f}_{Y_a}(x_i)$ and their task-relatedness parameters, increasing sample efficiency (right panel). When the two surfaces share minor patterns instead, such as in the second row plots of Figure 4.1, the sample efficiency gains are less significant; and in some more extreme cases where the surfaces do not share any similar pattern at all, assuming a multitask GP prior might also introduce bias. The issue of partial overlap might be less severe in scenarios with larger sample size, but not always; however, in settings with strong sample selection bias, or settings with multiple discrete actions or action spaces that grow with the sample size, the issue remains relevant. This is because the sampling mechanism

is inherently flawed, so that even with infinite samples, poor overlap regions do not fade away, independently of the modelling assumptions one makes.

Modelling assumptions on how to estimate $\{f_{Y_a}(\cdot)\}_{a=1}^{D}$ hence ultimately depends on domain expert knowledge about the setting (Hahn et al., 2020). If one possesses prior knowledge that the causal effects might be fairly homogeneous across units with different covariate realizations $\boldsymbol{X}_i = \boldsymbol{x}_i$, so that the CATE function is likely to display simple patterns (first row, Figure 4.1), then using a multitask approach would actually help incorporate this assumption in the model. Conversely, if one believes instead that CATE is likely to be a rather complex function itself (second row, Figure 4.1), estimating $\{f_{Y_a}(\cdot)\}_{a=1}^{D}$ independently would be a better choice.

4.2.3 Multiple Output Designs

Reverting back to setups with multiple correlated outcomes, we introduce a simple extension to the class of counterfactual GPs presented above that involves an extra multitask learning layer over the M outcomes $\boldsymbol{Y}_i$, in addition to the one over A_i. The way we formulate this is through an additional coregionalization matrix, in the same fashion as we did for the actions case. This extended version featuring *stacked coregionalization*, has a GP prior of the following form:

$$\begin{aligned} \boldsymbol{Y}_i &= \boldsymbol{f}_{Y_a}(\boldsymbol{x}_i) + \boldsymbol{\varepsilon}_i \ , \quad \mathbb{E}(\boldsymbol{\varepsilon}_i) = \mathbf{0} \\ \boldsymbol{f}_{Y_a}(\cdot) &\sim \mathcal{GP}\big(\mathbf{0}, K_Y(\cdot,\cdot)\big) \ , \quad K_Y(\cdot,\cdot) = B_Y \otimes B_A \otimes \tilde{K}(\cdot,\cdot) \ , \end{aligned} \tag{4.7}$$

where B_Y is the $M \times M$ coregionalization matrix over the outcomes, B_A the $D \times D$ coregionalization matrix over the actions and $\tilde{K}(\cdot,\cdot)$ is the base kernel. The vector-valued function $\boldsymbol{f}_{Y_a}(\cdot)$ in this case includes all the single-valued functionals $\{f_{a,m}(\cdot)\}_{a,m}^{D,M}$. The extra multitask learning layer over the outcomes $\boldsymbol{Y}_i$ is aimed at increasing sample efficiency by borrowing information among correlated outcomes, as opposed to fitting M separate counterfactual GPs with a single coregionalization layer over A, but it is also conceptually

sound, as the quantity of interest is indeed the joint interventional distribution $p(\boldsymbol{Y}|do(A=a), \boldsymbol{X}=\boldsymbol{x})$, which accounts for and explicitly models correlation between the outcomes, rather than the collection of marginal distributions $\{p(Y_m|do(A=a))\}_{m=1}^{M}$, which leaves correlation unspecified. As we will address in the later section, although the extra layer defined by B_Y allows for higher sample efficiency, it also poses some issues due to parameter proliferation and stability of the optimization problem in high dimensions.

4.3 Counterfactual Multitask Deep Kernel Learning

Gaussian Processes regressions are known to scale poorly with large samples. Their typical computational cost amounts to $\mathcal{O}(n^3)$ for training points and $\mathcal{O}(n^2)$ for test points. Closely related, coregionalized GPs suffer from poor scalability with sample size, but also from over-parametrization and instability in the optimization procedure as the number of inputs P and the number of discrete actions D increase. Deep Kernel Learning (DKL) was firstly introduced by Wilson et al. (2016) with the aim of combining the scalability of Deep Learning methods and the nonparametric Bayesian uncertainty quantification of GPs in tackling prediction tasks in high-dimensional settings. Given a base kernel $k(\cdot,\cdot)$ (e.g. linear, squared exponential, etc.), a DKL structure learns a functional $f_{Y_a}(\cdot)$ by passing the P inputs $\boldsymbol{X}_i \in \mathcal{X}$ through a deep architecture (a fully-connected feedforward neural network in our case), which maps them to a lower dimensional representation space via non-linear activation functions. The base kernel $k(\cdot,\cdot)$ is then applied in this lower dimensional representation space, $k(h^{(l)}, h^{(l)'})$, where $h^{(l)}$ is a neural network's hidden layer, constituting a final Gaussian Process layer (or an infinite basis functions representation layer). The resulting mathematical object can be described as a kernel being applied to a concatenation of linear and non-linear functions of the inputs, namely $\tilde{K}(\boldsymbol{x}, \boldsymbol{x}') = K(g_1 \circ ... \cdots \circ g_l(\boldsymbol{x}), g_1 \circ ... \circ g_l(\boldsymbol{x}'))$ (Bohn et al., 2019). Thus, the DKL architecture is end-to-end, fully-connected and learnt jointly:

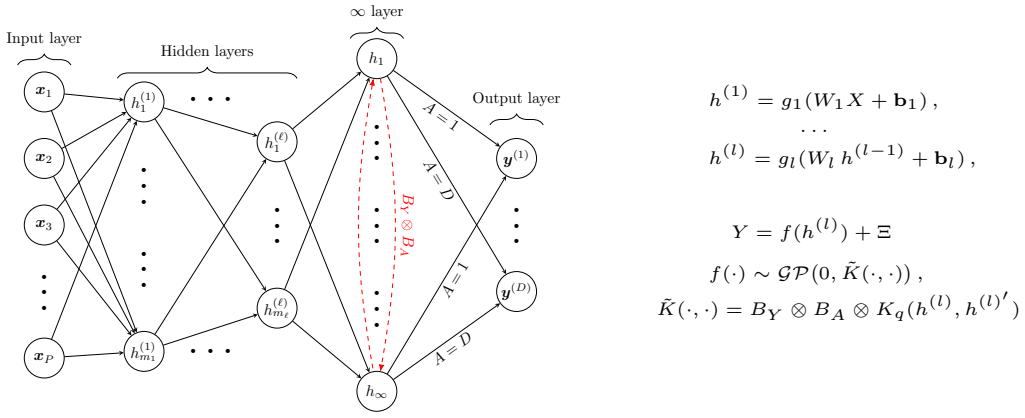


Figure 4.2: Counterfactual multitask DKL architecture. The P raw inputs are passed through a deep learning structure with ℓ hidden layers. Multioutput separable kernels (inducing coregionalization over actions A and outcomes $\boldsymbol{Y}$) are then applied to the last Gaussian Process hidden layer, before the M action-specific output layer. Parameters are estimated jointly by minimizing the negative log likelihood.

the P inputs are passed on to ℓ hidden neural nets layers where the last hidden layer before the GP layer typically maps them to a lower dimensional representation space (with e.g. two hidden units). This is what generates benefits in terms of scalability compared to a classic GP, as the base kernel $k(\cdot,\cdot)$ is applied to a lower dimensional representation space, rather than the higher dimensional inputs space directly. Another intrinsic advantage of DKL is that the deep architecture preceeding the GP layer can itself learn arbitrarily complex function, so the choice of a specific GP kernel becomes less cumbersome. For example, Wilson et al. (2016) show that DKL is more robust in recovering step functions, due to weaker smoothness assumptions compared to standard GP kernels.

DKL naturally presents some limitations concerning the more burdensome parameter tuning (e.g., hidden layers and units selection) and the fact that they more easily tend to overfit when overly-parametrized (we refer to Ober et al. (2021) for a more detailed discussion of the issue). The kernel $k(\cdot,\cdot)$ in the last GP layer of a DKL architecture can easily incorporate the separable kernel structure for multitask learning, in the same fashion as the class of causal GPs

presented earlier. Thus, we propose a multitask modelling framework to induce correlation across the action-specific functions $\{f_{Y_a}(\cdot)\}_{a=1}^{D}$, under the name of Counterfactual DKL (CounterDKL), where a similar Intrinsic Coregionalization Model (ICM) in the same spirit of (4.6) is placed on the last hidden layer of the neural network $h^{(l)} = g_l(W_l\, h^{(l-1)} + \boldsymbol{b}_l)$, where $(W_l, \boldsymbol{b}_l)$ are the last layer's weights and bias, such that

$$\begin{aligned} \boldsymbol{f}_{Y_a}(h^{(l)}) &\sim \mathcal{GP}\Big(0, K(\cdot,\cdot)\Big)\ , \quad \text{where} \\ K(h^{(l)}, h^{(l)'}) = K_A(a,a') \otimes \tilde{K}(h^{(l)}, h^{(l)'}) &= B \otimes \tilde{K}(h^{(l)}, h^{(l)'}) \end{aligned} \tag{4.8}$$

In this case, the Kronecker product of the coregionalization matrix occurs in the last hidden layer, and features lower dimensional representations instead of the potentially large number of raw inputs. Similarly, we can induce coregionalization over the M outcomes by adding another level of coregionalization, with the kernel reading $K(h^{(l)}, h^{(l)'}) = K_Y(y,y') \otimes K_A(a,a') \otimes K_q(h^{(l)}, h^{(l)'}) = B_Y \otimes B_A \otimes K_q\big(h^{(l)}, h^{(l)'}\big)$. Figure 4.2 graphically depicts a counterfactual multitask Deep Learning architecture, with fully-connected hidden layers, a final (infinite) GP layer and the M action-specific outcomes. The parameter space in CounterDKL comprises: i) the set of deep neural network's weight matrices $\{W_i\}_{i=1}^{l}$ and biases $\{\boldsymbol{b}_i\}_{i=1}^{l}$; ii) the base kernel $\tilde{K}$ hyperparameters Φ, e.g. in the case of squared exponential kernel Φ includes lengthscales and variances parameters $\Phi = \{\boldsymbol{\ell}, \sigma^2\}$; iii) the coregionalization matrix B entries. Hence, the parameter space is the collection $\Theta = (\{W_i\}_{i=1}^{l}, \{\boldsymbol{b}_i\}_{i=1}^{l}, \Phi, B)$. These parameters are estimated jointly via maximization of the log-marginal likelihood. More details are provided in the appendix and in Wilson et al. (2016); Gardner et al. (2018). In the next section, we will investigate properties of counterfactual multitask GPs and DKL on a variety of experiments.

4.4 Experiments

We evaluate the performance of counterfactual GPs and counterfactual DKL on a data generating process with three different but related tasks, and on

a real-world example combining experimental and observational data. The GPs implementations in the simulated examples all make use of the KISS-GP approximation to compute the base kernel covariance matrix (Wilson and Nickisch, 2015). The idea behind KISS-GP kernel approximation is that it is a combination of grid structure methods (e.g. Kronecker structure $K = K_1 \otimes ... \otimes K_P$, that guarantee a more easily solvable linear system) and inducing points methods for sparse GPs (Quiñonero-Candela and Rasmussen, 2005) (where only a chosen subset of $m < n$ points is used for parameter training) that guarantees better scalability. In KISS-GP, kernel $k(x, x')$ is approximated as $k(x, x') \approx \boldsymbol{w}_x k(U, U')\boldsymbol{w}_{x'}$, where U are m latent inducing points and $\boldsymbol{w}_x, \boldsymbol{w}_{x'}$ are sparse interpolation vectors[1]. KISS-GP allows one to go from $\mathcal{O}(n^3)$ typical training computational cost of GPs to $\mathcal{O}(n + m)$.

4.4.1 Fully Simulated Example

We consider a simulated setting with $D = 4$ possible actions $\mathcal{A} = \{0, 1, 2, 3\}$ and $M = 2$ correlated outcomes $\boldsymbol{Y} = (Y_1, Y_2) \in \mathbb{R}^2$. Actions and outcomes are generated according to a propensity score/policy $\pi_b(\boldsymbol{x}_i) = p(A_i = a|\boldsymbol{X}_i = \boldsymbol{x}_i)$ and an outcome function $\boldsymbol{f}_Y(\boldsymbol{x}_i)$, both dependent on the covariates $X_i \in \mathcal{X}$. The DGP is fully described in the supplementary materials. The models we compare are the following: i) separate standard GP regressions, employed to fit $f_{Y_\text{d}}(\cdot)$ distinctly for each outcome and for each action (**GP**, as a T-Learner); ii) counterfactual multitask GP regression (Alaa and van der Schaar, 2017, 2018), with coregionalization over A_i only, meaning that we fit two separate models for each outcome, but a unique multi-valued function model for A_i (**CounterGP**); iii) counterfactual multioutput GP regression, a unique model with coregionalization both over A_i and Y_i (**MOGP**); iv) separate DKL regressions with 3 hidden layers of $[50, 50, 2]$ units, the equivalent of i) but with deep kernel implementation (**DKL**); v) counterfactual multitask DKL regression with 3 hidden layers of $[50, 50, 2]$ units, the DKL equivalent of ii)

[1] All experiments were run on a Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz, 8Gb RAM CPU.

(**CounterDKL**); vi) counterfactual multioutput DKL, the DKL equivalent of iii) (**MODKL**). In particular, we consider two slightly different versions of this setup. In the first version we fix the number of covariates to $P = 10$ (only 7 of them being relevant for the estimation) and study the behaviour of the estimators with increasing sample size $n \in \{500, 1000, 1500, 2000, 2500\}$. In the second version we fix sample size to $n = 1500$ and study the behaviour of the estimators with increasing number of covariates $P \in \{10, 15, 20, 25\}$. Performance of the models is evaluated on the following three related tasks:

- **CATE**: The first is the prediction of CATE: $\mathbb{E}(Y_i|do(A_i = a), \boldsymbol{X}_i = \boldsymbol{x}_i)$. This is carried out using a 80% training set, and evaluated via root PEHE (equivalent to RMSE) on a 20% left-out test set.

- **OPE**: The second is Off-Policy Evaluation, which is concerned with quantifying how good a given alternative policy π_e is, compared to the action allocation policy that generated the data (behavior policy, π_β). This is done by estimating the *policy value* $\mathcal{V}(\pi_e) = \mathbb{E}_{\mathcal{X},\mathcal{A},\mathcal{Y}}\big[\sum_i \pi_e(a_i|x_i)\big(\boldsymbol{Y}_i|do(A_i = a_i)\big)\big]$ originating from π_e. In our case we pick the alternative policy to be a uniformly-at-random action allocation, i.e. $\pi_e \sim \text{Multinom}(.25, .25, .25, .25)$. Performance is evaluated through RMSE on the entire sample.

- **OPL**: The last is Off-Policy Learning, which is concerned with finding the optimal policy $\pi^* : \mathcal{X} \to \mathcal{A}$, that is the policy that generates the highest *policy value*: $\pi_p^* \in \arg\max_{\pi_p \in \Pi} \mathcal{V}(\pi_p)$. This last task is evaluated through an Accuracy metric on the whole sample, which we label Optimal Allocation Rate (OAR), indicating the percentage of units correctly assigned to their specific optimal action $\pi^*(x_i) = a$, i.e. the action that generates the best outcome for them.

Since we are dealing with $M = 2$ outcomes, we produce performance measurements on PEHE/RMSE and optimal allocation rate for both outcomes and then average them, assuming both outcomes are given equal policy importance and live on the same scale. For both versions of the setup, namely increasing

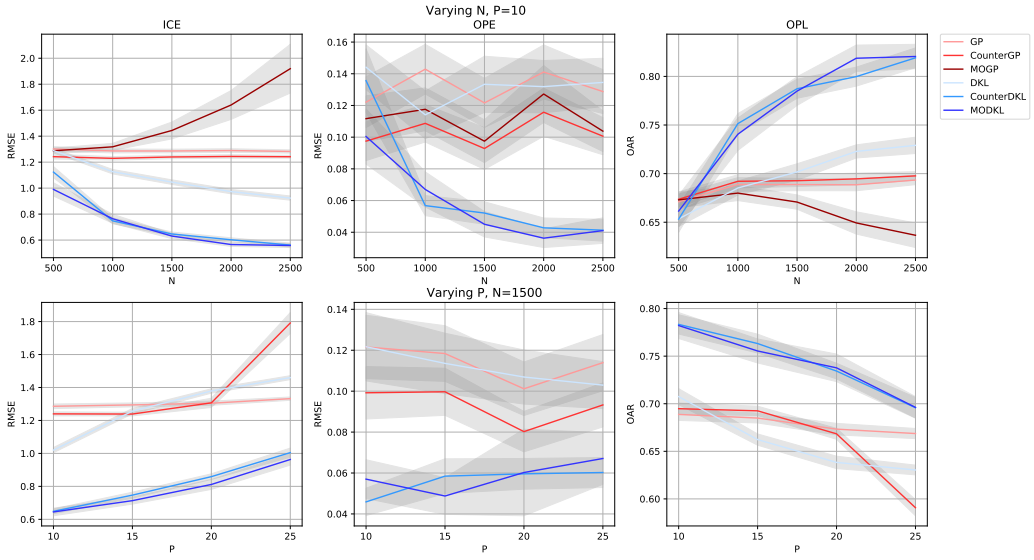


Figure 4.3: Results on performance of the methods compared, in terms of RMSE or Optimal Allocation Rate (OAR), averaged across $B = 100$ replications for each $n \in \{500, 1000, 1500, 2000, 2500\}$ (first row) and each $P \in \{10, 15, 20, 25\}$ (second row). First column: RMSE evaluated on the individual causal effect (ICE) estimation task (on the test set). Second column: RMSE evaluated on the OPE task. Third column: OAR on the OPL task, defined as percentage of units correctly allocated to the best action among the D ones.

n and increasing P, we replicate the experiment $B = 100$ times to obtain Monte Carlo averages and 95% confidence intervals for the metrics. Results are depicted in Figure 4.3. RMSE performance in all models for increasing n and P behaves accordingly what we discussed earlier. CounterDKL and MODKL perform consistently better than the GP models, as they scale better with an increasing sample size n and increasing number of predictors P. Particularly MOGP's performance deteriorates for issues related to stability of the marginal likelihood maximization and over-parametrization, as we had to omit it from the study of increasing predictors due to failed convergence for $P > 10$. The advantages over standard DKL regression instead are entirely attributable to sample efficiency gains from multitask coregionalization in CounterDKL and MODKL, both in the increasing n and increasing P studies. In the case of increasing P, we emphasize that as predictor space grows larger the causal

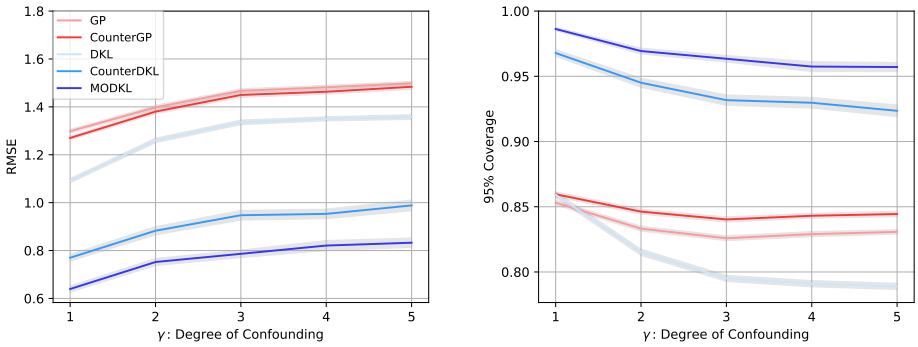


Figure 4.4: Models' performance in terms of RMSE (left plot) and 95% Coverage (right plot), in estimating Individual Causal Effects (ICE) on a 20% left-out test set, given an increasing level of confounding, represented by the γ parameter: higher values of γ corresponds to higher probability of being assigned to one of the two action $A_i = \{3, 4\}$, thus generating more arms imbalance.

DGP becomes relatively sparser (only 7 predictors out of P remain relevant for the estimation), especially in the case of $P = \{20, 25\}$. So in these two cases the batch of DKL models would perhaps achieve better performance from increasing the number of hidden units or hidden layers and adding regularization (dropout, $\ell 1$ or $\ell 2$ regularizers) in the deep architecture part.

In addition, we run a slightly different version of the CATE experiment above, to further investigate properties of the models in terms of uncertainty quantification, that we measure through the 95% coverage of each CATE estimates (then averaged over actions and outcomes). This has also been defined earlier in Chapter 3 as

$$\text{Coverage}_{95\%} = \frac{1}{n}\sum_{i=i}^{n} \mathbb{I}\Big(\hat{f}_a(\boldsymbol{x}_i)_{low} \leq f_a(\boldsymbol{x}_i) \leq \hat{f}_a(\boldsymbol{x}_i)_{upp}\Big),$$

where $\hat{f}_a(\boldsymbol{x}_i)_{low}$ and $\hat{f}_a(\boldsymbol{x}_i)_{upp}$ are the lower and upper bands of the 95% credible interval output by the model on $\hat{f}_a(\boldsymbol{x}_i)$, while $f_a(\boldsymbol{x}_i)$ is the true individual counterfactual outcome corresponding to action $A_i = a$. Given fixed $n = 2000$ and $P = 20$ and a similar data generating process as before, we introduce the

parameter γ, which governs the level of confounding, or the degree of groups imbalance in terms of action allocation. Particularly, for increasing values of γ, we assign higher probability of choosing one of the two action $A_i = \{3, 4\}$. This generates action arms imbalance as it leaves gradually less units in arms $\{1, 2\}$. Results are gathered in Figure 4.4, where MODKL display higher performance both in terms of error and uncertainty quantification.

4.4.2 Real-World Example: Job Training Programs and Unemployment

We demonstrate the efficiency of CounterDKL also on a second experiment taken from Shalit et al. (2017), involving a popular real-world study on a job training program, dating back to LaLonde (1986). The distinctive feature of this dataset is that it combines a randomized and an observational subgroups, where the aim is to estimate the effects of participation on a job training program on earnings and employment. The randomized experiment features 297 treated and 425 control units. The observational subsample is instead made of 2490 control units only. The binary treatment $A_i \in \{0, 1\}$ denotes participation to the job training program. The original outcome Y_i is earnings after the program, which is censored continuous ($Y_i = 0$ for unemployed units). However, following Shalit et al. (2017), we construct a binary indicator $Y_i \in \{0, 1\}$ denoting employment status at the end of the job training program as outcome. This gives us the opportunity to demonstrate the use of the methods presented earlier also on binary/categorical type of outcomes. To this end we use the classification method for GPs proposed in Milios et al. (2018), where class labels are interpreted as coefficients of a degenerate Dirichlet distribution, which makes the GP classification task efficiently faster and more scalable. The 7 covariates $\boldsymbol{X}_i \in \mathcal{X}$ in the study are the following: age, years of schooling, african american ethnicity, hispanic ethnicity, marital status, high school diploma. Given the presence of a randomized subsample, we can exploit it to compute unbiased estimates of the quantities of interest and treat them as ground truth. The two quantities of interest in this case are: i) the Average Treatment Effect

Model	Train MAE	Test MAE	Train $\mathcal{R}_{\text{pol}}$	Test $\mathcal{R}_{\text{pol}}$	Runtime (s)
GP	0.033 ± 0.006	0.036 ± 0.008	0.22 ± 0.02	0.27 ± 0.02	171.3 ± 16.1
CounterGP	0.033 ± 0.006	0.035 ± 0.007	0.24 ± 0.01	0.27 ± 0.02	248.6 ± 6.4
PCA + GP	0.073 ± 0.002	0.074 ± 0.003	0.22 ± 0.01	0.27 ± 0.02	66.3 ± 2.4
PCA + CounterGP	0.074 ± 0.001	0.074 ± 0.001	0.23 ± 0.01	0.26 ± 0.02	126.1 ± 3.9
AutoEnc + GP	0.075 ± 0.004	0.075 ± 0.003	0.21 ± 0.03	0.27 ± 0.02	76.0 ± 3.0
AutoEnc + CounterGP	0.076 ± 0.003	0.076 ± 0.003	0.24 ± 0.02	0.30 ± 0.03	138.7 ± 9.2
DKL	0.029 ± 0.011	0.042 ± 0.015	**0.20 ± 0.01**	**0.21 ± 0.02**	44.8 ± 3.3
CounterDKL	**0.011 ± 0.003**	**0.015 ± 0.005**	**0.22 ± 0.01**	**0.25 ± 0.02**	122.7 ± 7.4

Table 4.1: Train and test set performance on the Jobs data experiment in terms of Mean Absolute Error (MAE) in estimating ATT, Policy Risk ($\mathcal{R}_{\text{pol}}$) and overall runtime (s), with 10-fold cross-validated 95% intervals. Bold indicates better performance.

on the Treated group (ATT), defined as $\text{ATT} = T^{-1}\sum_{i=1}^{T_e} y_i - C^{-1}\sum_{i=1}^{C} y_i$, where T and C are the number of treated and control units in the experimental data; ii) the Policy Risk (Shalit et al., 2017), defined as the average error in allocating the treatment according to the ICE estimates policy rule — namely $\pi(\boldsymbol{x}_i) = 1$ if $\text{ICE} = \mathbb{E}(Y_i|do(A_i = 1), \boldsymbol{x}_i) - \mathbb{E}(Y_i|do(A_i = 0), \boldsymbol{x}_i) > 0$ — or $\mathcal{R}_{\text{pol}} = 1 - \big[\mathbb{E}\big(Y|do(A_i = 1), \pi(\boldsymbol{x}_i) = 1\big)p(\pi(\boldsymbol{x}_i) = 1) + \mathbb{E}\big(Y|do(A_i = 0), \pi(\boldsymbol{x}_i) = 0\big)p(\pi(\boldsymbol{x}_i) = 0)\big]$. Notice that we cannot measure performance on ICE directly as this is always unobservable in real-world scenarios; also, we restrict analysis of average causal/treatment effects on the treated group since we are sure that overlap holds there, as all the treated units were part of the randomized experiment subgroup, while the observational subgroup is made only of control units. More details about this experiment can be found in the Appendix C of supplementary materials and in Shalit et al. (2017).

We compare the following models: i) GP and CounterGP, as in Alaa and van der Schaar (2017); ii) vanilla PCA plus either GP or CounterGP; iii) vanilla deep AutoEncoder plus either GP or CounterGP; iv) DKL and CounterDKL (ours). Results on performance are gathered in Table 4.1, in terms of 70%-30% train and test set Mean Absolute Error (MAE) on ATT, Policy Risk $\mathcal{R}_{\text{pol}}$ and average runtime, accompanied by 10-fold cross-validated 95% error intervals.

Model	Train $\sqrt{\text{PEHE}_\tau}$	Test $\sqrt{\text{PEHE}_\tau}$
RF	1.85 ± 0.13	2.39 ± 0.17
X-RF	3.29 ± 0.23	3.37 ± 0.24
CF	3.10 ± 0.21	3.07 ± 0.20
BART	0.97 ± 0.04	1.44 ± 0.09
X-BART	2.07 ± 0.14	2.13 ± 0.14
BCF	0.87 ± 0.05	1.34 ± 0.10
CounterGP	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.70 \pm 0.04}$
CounterDKL	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.67 \pm 0.04}$

Table 4.2: $\sqrt{\text{PEHE}_\tau}$ on CATE estimates, plus 95% Monte Carlo intervals, of compared models on the semi-simulated IHDP setup, evaluated on 80%-20% train-test sets.

In this example multitasking is induced only over the binary treatment, as we deal with just a single outcome of interest. As the results depict, by operating jointly via a unique loss function, CounterDKL is significantly more efficient than naively applying dimensionality reduction and fitting a multitask GP on a lower dimensional space as two separate steps. It also displays gains over CounterGP, thanks to its deep component that guarantees better computational time (in terms of runtime) and scalability, and is able learn arbitrarily complex functions while imposing weaker smoothness assumptions than standard GP kernels, even on a low-dimensional covariate space example such as the one presented here (7 covariates).

4.4.3 The Infant Health Development Program data

Finally we compare CounterDKL with few other methods for CATE estimation, encountered earlier in Chapter 2, on the simulated experiment utilizing the Infant Health Development Program (IHDP) data, originally found in Hill (2011). As explained also in Chapter 2 experimental sections, this consists in a semi-simulated setup, in that it makes use of real-world data from the IHDP study, a randomized clinical trial aimed at improving the health status of premature infants with low weight at birth through pediatric follow-ups and parent support groups, and recreates an observational type of study by

removing a non-random portion of treated units, namely those with "non-white mothers". This leaves a total of 139 observations in the treated group and 608 in the control. In addition, the semi-simulated setup uses the real-world binary treatment $A_i \in \{0, 1\}$ and the 25 available covariates $\boldsymbol{X}_i \in \mathcal{X}$, but simulates the two continuous potential outcomes $(Y_1, Y_0) \in \mathbb{R}^2$, as described in the non-linear "Response Surface B" setting in Hill (2011).

As anticipated above, the estimand of interest in this case is CATE again. The models we compare include: i) vanilla Random Forest (**RF**), as a T-Learner; ii) X-Learner version of Random Forests (**X-RF**) as in (Künzel et al., 2017); iii) Causal Forest (**CF**), or Random Forests as an R-Learner, developed by Wager and Athey (2018); iv) vanilla Bayesian Additive Regression Trees (**BART**), as a T-Learner; v) X-Learner version of BART (**X-BART**); vi) Bayesian Causal Forests (**BCF**) by Hahn et al. (2020); vii) Counterfactual GP (**CounterGP**) as in Alaa and van der Schaar (2018); viii) our Counterfactual DKL (**CounterDKL**) with [100, 100, 2] hidden layers. Results in Table 4.2 report $\sqrt{\text{PEHE}_\tau}$ estimates relative to 1000 replication of the experiment on 80%-20% train-test split as in Alaa and van der Schaar (2017).

Chapter 5

Conclusions

This last chapter summarizes the ideas presented in the previous ones and discusses some open problems in the causal inference literature, in addition to issues in the more general machine learning domain that can be addressed using tools from causality.

We start by very briefly summarizing the content of the different chapters. In Chapter 1 we introduced the fundamental notions behind causal inference and causal learning, together with the mathematical notation used in different causal frameworks. We highlighted how the frameworks are interchangeable and lead to the same causal identification results on *Average* and *Conditional Average Treatment Effects* (i.e., ATE and CATE respectively). In Chapter 2 we reviewed the most recent and popular regression adjustment models, based on modern ML techniques, for CATE estimation, by developing a new unifying taxonomy and laying out their empirical finite-sample properties. In Chapter 3 we proposed two new methods, Shrinkage Bayesian Causal Forest and Interpretable Causal Neural Networks, specifically aimed at addressing the interrelated issues of interpretability, uncertainty coverage and targeted regularization on prognostic and moderating effects, in the estimation of CATE. Finally, in Chapter 4 we developed a Bayesian model based on multitask Gaussian Processes and multitask Deep Kernel Learning to efficiently tackle scenarios with high-dimensionality over multiple axis, i.e., multiple actions, outcomes and high-dimensional covariate space.

5.1 Further challenges in Causal Learning

As thoroughly argued in the introductory Chapter 1, causal inference/learning is a hard problem, particularly with observation data. The assumptions outlined in previous chapters, such as unconfoundedness and SUTVA (i.e., no interference), might not often hold in practice. This requires due adjustments and use of slightly different methods, that nonetheless imply a different set of causal assumptions. We discuss here below general cases where the uncofoundedness assumption (which is untestable) is violated, and also cases where the *i.i.d.* sampling assumption is violated and how this affects causal assumptions as well.

5.1.1 The unconfoundedness assumption

Unconfoundedness, i.e. the absence of unobserved common causes of A and Y, often fails to hold in some types of studies, particularly in disciplines that deal with complex systems such as socio-economic sciences. In these cases, attempting to retrieve a point estimate of the causal effects of interest would result in a degree of bias that depend on the entity and strength of the unobserved confounders. In order to better illustrate this concept, we briefly derive bias resulting from hidden confounders in the context of a simple linear model. Suppose the structural causal model, represented by the DAG in Figure 5.1a), is the following:

$$\begin{aligned} A &= f_A(X, U, \varepsilon_A) = \alpha_1 X + \alpha_2 U + \varepsilon_A & \varepsilon_A &\sim \mathcal{N}(0, \sigma^2_{\varepsilon_A}) \\ Y &= f_Y(X, A, U, \varepsilon_Y) = \beta_1 X + \gamma A + \beta_2 U + \varepsilon_Y & \varepsilon_Y &\sim \mathcal{N}(0, \sigma^2_{\varepsilon_Y}) \end{aligned} \tag{5.1}$$

where X is an observed confounder, while U is a hidden one, and trivially ATE = CATE = γ. If we proceed towards identification by conditioning only on the observed confounder X, we get that the conditional mean response of group A is

$$\mu_A = \mathbb{E}_X[\mu_A(x)] = \mathbb{E}_X\big[\mathbb{E}[Y \mid X, A = a]\big] =$$

(a) U - Confounded (b) L - Instrument

Figure 5.1: a) DAG with unobserved confounder U; b) DAG with F representing a factor or latent variable that, if conditioned on, restores unconfoundedness.

$$
\begin{aligned}
&= \mathbb{E}_X[\beta_1 X + \gamma A + \beta_2 U + \varepsilon_Y \mid X, A = a] = \\
&= \mathbb{E}_X\big[\beta_1 X + \gamma a + \beta_2 \mathbb{E}[U \mid X, A = a]\big] = \\
&= \mathbb{E}_X\Big[\beta_1 X + \gamma a + \beta_2 \left(\frac{a - \alpha_1 X}{\alpha_2}\right)\Big] = \\
&= \left(\gamma + \frac{\beta_2}{\alpha_2}\right) a - \left(\beta_1 - \frac{\alpha_1}{\alpha_2}\right) \mathbb{E}_X[X] \ ,
\end{aligned}
$$

so that when we attempt to retrieve ATE/CATE estimates $\gamma_X = \mathbb{E}_X[\tau(x)] = \mathbb{E}_X[\mu_1 - \mu_0]$, we run into confounding bias that proportional to ratio of the coefficients relative to U specified in the SCM equations above:

$$
\text{Bias}(\gamma_X) = \gamma_X - \gamma = \mathbb{E}_X[\tau(x)] - \mathbb{E}_{X,U}[\tau(x,u)] = \gamma + \frac{\beta_2}{\alpha_2} - \gamma = \frac{\beta_2}{\alpha_2} \ . \quad (5.2)
$$

There are different ways one can attempt to tackle settings where unconfoundedness assumption is not credible. Generally these alternative approaches come at the cost of some extra assumptions. For instance, one possibility to achieve full identifiability back is through the use of Instrumental Variables (IV) (Angrist et al., 1996; Pearl, 2009a; Imbens and Rubin, 2015), as depicted by the causal DAG in Figure 5.1b). An Instrumental Variable L is essentially an auxiliary variables that does not depend on the unobserved confounder U and affect the outcome Y only through A. The contour of L in Figure 5.1b) is dashed, to indicate that L can also be learned as a "de-confounder" latent variable, e.g. if $\boldsymbol{A}$ is multiple actions and have sufficient variability. For example, Wang and Blei (2019) have considered using Bayesian nonparametric latent variable

models to learn the latent representation L and tackle IV estimation. The problem with the IV strategy is that good instruments requires extra identifiability assumptions (e.g., **validity**, **exclusion restriction** and **exogeneity**) and these are not always achievable/available.

Another possible way of addressing non-identifiable causal effects due to unobserved confounding is via *partial identification* methods (Manski, 1990; Balke and Pearl, 1997; Pearl, 2009a). The main intuition behind these type of methods is that they rely on the identification and estimation of "bounds" around causal effects, rather than on point estimates identification, which implies constructing an interval $\hat{\tau}_{\text{low}}(\boldsymbol{x}) \leq \tau_{\text{true}} \leq \hat{\tau}_{\text{upp}}$.

5.1.2 The *i.i.d.* assumption

Another class of problems in causal inference relates instead to settings where the *i.i.d.* assumption in the sampling of data fails to hold. In particular, this can refer to dynamic settings, where data are *i.i.d.* conditional on a "time index", or loosely speaking within a specific time slice. The literature on causal effects estimation over time is abundant and spans different topics that are in fact very similar between each other, such as Dynamic Treatment Regimes or sub-topics in the Reinforcement Learning domain, discussed in the previous chapter (Murphy, 2003; Robins, 2004; Schulte et al., 2014; Zhao et al., 2015; Sutton and Barto, 2018).

Another way *i.i.d.* assumption can break apart is under some form of interference between the units of analysis. This is reasonably common in e.g. social science studies, and as a consequence violates the SUTVA assumption as well. In some specific works, we could assume some type of known (or partially known) networked interference, where essentially units are *i.i.d.* conditional on their "neighbors" in the network (Hudgens and Halloran, 2008; Tchetgen and VanderWeele, 2010; Forastiere et al., 2021; Ma and Tresp, 2021). In these cases, manipulative interventions exert the usual **direct effect** on the units' outcome, but also a **spillover effects** on their neighbors (e.g., think about policies regarding vaccine rollout).

5.2 Causality for Machine Learning

To conclude, we highlight that, although this work has intensively revolved around how flexible probabilistic machine learning techniques can used for estimating causal effects and designing optimal policies ("machine learning for causality"), there is an increasing amount of contributions whose interest lies instead in investigating how causal reasoning methods can be exploited in pure machine learning tasks (Schölkopf et al., 2021) ("causality for machine learning").

An example of ambitious application of causal concepts in ML regards the general problem of transfer learning and domain adaptation. The main idea behind domain adaptation is to learn a model that is "transportable" to other unseen data, where we usually do not have a different training set to train the model anew. Thus, ideally we would want a model that is invariant of biases and spurious correlation that are exclusively characterizing the available training data (Peters et al., 2016a; Arjovsky et al., 2020), or in other words invariant to the "environment". A typical intuitive textbook example refers to image classification. Suppose we train a model according to the empirical risk minimization principle on images of cows on a grass. The model will typically struggle to generalize and correctly classify images of cows on a beach, since it will inevitably inherit environment dependence. We would ideally want then to build a classifier that learns specific causal non-spurious associations/mechanisms, and results in better generalization, rather than one that only performs very well in-environment because of spurious correlations overfitting. There is fairly recent stream of contributions that concentrates on this issue of learning causally transportable (deep) representation models (Peters et al., 2016a; Arjovsky et al., 2020; Schölkopf et al., 2021).

Another example of the use of causal reasoning tools in ML pertains to interpretability/explainability in Black-Box models (Verma et al., 2020). The main idea here is to use counterfactual experiments to understand why a ML model has generated or predicted a certain output, by for example resorting to

"ablation" studies on nodes in a deep convolutional neural network to understand which particular traits that specific node is capturing in an image. Finally, causality has also been successfully employed for AI/ML fairness and safety. Contributions such as Kusner et al. (2017); Chiappa (2019) have developed the concept of "Counterfactual Fairness" in ML methods, with the aim of taking into account sensitive inputs/attributes in the training phase of a ML model. The idea behind "Counterfactual Fairness" is quite simple: given a sensitive attribute such as ethnicity or gender, we ideally would like to train a safe ML model that does not predict different outcomes solely based on these attributes, *ceteris paribus*, which can potentially lead to discriminatory decision-making.

Appendix A

Supplementary Material for Chapter 2

A.1 Supplementary Results on simulated examples

This appendix section contains additional results on the simulated exercises encountered in Section 2.4.1 and 2.4.2, Chapter 2, respectively. In particular we report tables with Bias_τ estimates results on CATT (in-ovelap regions), and Bias_τ and $\sqrt{\text{PEHE}}_\tau$ estimates on CATC (out-of-overlap regions), for the various meta-learning models compared. CATC results are based on non completely identifiable settings, and are shared for giving a hint of the generalizability of the methods in non-overlap regions. As discussed in the core section of 2 and 3, the Bayesian approach on CATE estimation offers a ready-made tool for the detection of non-overlap regions (Hill and Su, 2013), but this is reflected in "exploding" credible intervals in those regions, as one is attempting to extrapolate/generalized to effectively a no-data region (i.e., shift in distribution).

As a general takehome message, it emerges from these additional results that Bayesian methods based on multitask and τ learner frameworks are either the best or never trail a lot behind.

Table A.1: IHDP and ACTG-175 simulated exercises of Section 2.4. $\mathbf{Bias}_{\boldsymbol{\tau}}$ estimates ± 95% confidence intervals for each tested model on in-overlap **CATT**, on train and test sets respectively.

	IHDP		**ACTG-175**	
	Train	Test	Train	Test
S-RF	-0.05 ± 0.04	-0.07 ± 0.05	0.01 ± 0.01	0.00 ± 0.01
S-BART	-0.10 ± 0.03	-0.16 ± 0.04	-0.03 ± 0.01	-0.03 ± 0.01
T-RF	-0.02 ± 0.03	-0.01 ± 0.03	0.00 ± 0.01	-0.01 ± 0.01
T-BART	-0.04 ± 0.02	-0.03 ± 0.03	0.01 ± 0.01	0.00 ± 0.01
X-RF	-0.08 ± 0.03	-0.10 ± 0.05	-0.04 ± 0.01	-0.04 ± 0.01
X-BART	0.06 ± 0.02	0.04 ± 0.02	-0.06 ± 0.01	-0.06 ± 0.01
R-LASSO	0.18 ± 0.03	0.14 ± 0.03	0.01 ± 0.01	0.01 ± 0.01
R-BOOST	0.22 ± 0.02	0.23 ± 0.04	0.10 ± 0.01	0.12 ± 0.01
CF	0.06 ± 0.03	0.05 ± 0.05	-0.04 ± 0.01	-0.04 ± 0.01
CMGP	**0.00 ± 0.01**	**0.00 ± 0.02**	-0.04 ± 0.01	-0.05 ± 0.01
NSGP	**0.00 ± 0.01**	**0.00 ± 0.02**	-0.04 ± 0.01	-0.05 ± 0.01
BCF	0.04 ± 0.02	0.03 ± 0.03	**0.00 ± 0.01**	**0.00 ± 0.01**

Table A.2: IHDP and ACTG-175 simulated exercises of Section 2.4. $\mathbf{Bias}_{\boldsymbol{\tau}}$ estimates ± 95% confidence intervals for each tested model, on out-of-overlap **CATC**, on train and test sets respectively.

	IHDP		**ACTG-175**	
	Train	Test	Train	Test
S-RF	-0.05 ± 0.05	-0.05 ± 0.05	-0.09 ± 0.01	-0.08 ± 0.01
S-BART	-0.14 ± 0.03	-0.13 ± 0.03	-0.04 ± 0.01	-0.04 ± 0.01
T-RF	-0.03 ± 0.01	-0.04 ± 0.02	-0.09 ± 0.01	-0.08 ± 0.01
T-BART	**0.01 ± 0.01**	**0.01 ± 0.01**	-0.07 ± 0.01	-0.07 ± 0.01
X-RF	0.14 ± 0.03	0.14 ± 0.04	-0.04 ± 0.01	-0.04 ± 0.01
X-BART	-0.13 ± 0.03	-0.13 ± 0.03	-0.02 ± 0.01	-0.02 ± 0.01
R-LASSO	0.06 ± 0.02	0.06 ± 0.02	0.02 ± 0.01	0.02 ± 0.01
R-BOOST	0.41 ± 0.03	0.40 ± 0.04	0.18 ± 0.01	0.16 ± 0.01
CF	0.24 ± 0.04	0.25 ± 0.04	**-0.01 ± 0.01**	**-0.01 ± 0.01**
CMGP	0.11 ± 0.07	0.10 ± 0.07	-0.04 ± 0.01	-0.04 ± 0.01
NSGP	0.11 ± 0.02	0.10 ± 0.03	-0.05 ± 0.01	-0.05 ± 0.01
BCF	0.03 ± 0.01	0.04 ± 0.02	**0.01 ± 0.01**	**0.01 ± 0.01**

Table A.3: IHDP and ACTG-175 simulated exercises of Section 2.4. $\sqrt{\mathbf{PEHE}}_{\boldsymbol{\tau}}$ estimates ± 95% confidence intervals for each tested model, on out-of-overlap **CATC**, on train and test sets respectively.

	IHDP		**ACTG-175**	
	Train	Test	Train	Test
S-RF	2.95 ± 0.25	3.16 ± 0.25	0.59 ± 0.01	0.51 ± 0.01
S-BART	2.13 ± 0.14	2.20 ± 0.14	0.45 ± 0.01	0.46 ± 0.01
T-RF	1.70 ± 0.12	2.39 ± 0.17	0.60 ± 0.01	0.51 ± 0.01
T-BART	0.82 ± 0.02	1.43 ± 0.09	0.55 ± 0.01	0.55 ± 0.01
X-RF	3.35 ± 0.23	3.39 ± 0.23	**0.36 ± 0.01**	**0.36 ± 0.01**
X-BART	2.21 ± 0.15	2.25 ± 0.15	0.44 ± 0.01	0.44 ± 0.01
R-LASSO	2.02 ± 0.14	2.07 ± 0.15	0.63 ± 0.01	0.63 ± 0.01
R-BOOST	2.34 ± 0.15	2.52 ± 0.16	0.52 ± 0.01	0.51 ± 0.01
CF	3.14 ± 0.21	3.07 ± 0.20	0.40 ± 0.01	0.40 ± 0.01
CMGP	0.83 ± 0.09	1.10 ± 0.11	0.45 ± 0.01	0.45 ± 0.01
NSGP	**0.66 ± 0.03**	**0.97 ± 0.09**	0.43 ± 0.01	0.43 ± 0.01
BCF	0.73 ± 0.02	1.35 ± 0.10	0.39 ± 0.01	0.39 ± 0.01

A.2 ACTG-175 data: a third simulated exercise

In this second short appendix section we present results obtained from a third semi-simulated exercise involving the ACTG-175 dataset. The structure of the utilized ACTG-175 data is exactly the same as the one found in the example in Section 2.4.2 (same number of covariates, sample size, etc.). The only difference lies in how the prognostic score and CATE functions are simulated. For this third setup we chose slightly more complex surfaces compared to the ones in the other ACTG-175 simulation, to provide an additional example on the performance of the reviewed methods under a more challenging data generating process (closer to the one seen in the IHDP data example). More specifically,

the two $\mu(\boldsymbol{x}_i)$ and $\tau(\boldsymbol{x}_i)$ surfaces are generated as

$$\begin{aligned}\mu(\boldsymbol{x}_i) &= 6 + 0.3wtkg^2 - \sin(age) \cdot (gender + 1) + 0.6hemo \cdot race - 0.2z30 \ , \\ \tau(\boldsymbol{x}_i) &= 1 + 1.5\sin(wtkg) \cdot (karnof_{hi} + 1) + 0.4age^2 \ .\end{aligned} \tag{A.1}$$

Surfaces in (A.1) feature more complex functions and more interaction terms. As in the other ACTG-175 data setup, Gaussian noise was added by simulating $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, with standard deviation equal to $\sigma = 0.2(\mu_{max} - \mu_{min})$, where μ_{max} is the sample maximum of the generated prognostic score, while μ_{min} is the sample minimum value.

Table A.4: Third simulated setup (ACTG-175 data). $\sqrt{\textbf{PEHE}_{\boldsymbol{\tau}}}$ estimates ± 95% confidence intervals for each tested model on in-overlap **CATT**, on the train and test sets respectively.

	ACTG-175: 3^{rd} simulation	
	Train	Test
S-RF	0.97 ± 0.01	1.03 ± 0.01
S-BART	0.91 ± 0.01	0.95 ± 0.01
T-RF	0.82 ± 0.01	0.88 ± 0.01
T-BART	0.84 ± 0.01	0.93 ± 0.01
X-RF	0.76 ± 0.01	0.81 ± 0.01
X-BART	0.81 ± 0.01	0.88 ± 0.01
R-LASSO	1.13 ± 0.01	1.18 ± 0.01
R-BOOST	0.87 ± 0.01	0.92 ± 0.01
CF	0.87 ± 0.01	0.87 ± 0.01
CMGP	0.61 ± 0.01	0.72 ± 0.01
NSGP	**0.59 ± 0.01**	**0.70 ± 0.01**
BCF	0.77 ± 0.01	0.87 ± 0.01

Results in terms of performance of the tested models are reported in Table A.4. A ranking similar to the one encountered in the IHDP data example emerges, with the Gaussian Process Multitask-Learners being the best methods, followed up by X-RF and BCF. This is explained by the fact that CATE here is the result of a complex function, which tends to favour methods that

fit separate surfaces (T-Learners, Multitask-Learners, etc.), just like in the IHDP example. Notice in fact that also in this case T-Learners show better performance than their S-Learner counterparts (T-RF vs S-RF, T-BART vs S-BART).

A.3 NHANES variables list

Table A.5 contains the full list of variables included in the NHANES dataset analyzed in Section 2.5.

Table A.5: NHANES variables

Variable	**Description**
BMI	Numeric. Body mass index (outcome variable)
school_meal	Binary (treatment indicator)
age	Numeric (child's age)
childSex	Binary (male = 1)
afam	Binary (African American = 1)
hisam	Binary (Hispanic = 1)
povlev_200	Binary (family above 200% federal poverty lvl = 1)
sup_nutr	Binary (supplementary nutrition program = 1)
stamp_prog	Binary (food stamp program = 1)
food_sec	Binary (food security in household = 1)
ins	Binary (any insurance = 1)
refsex	Binary (adult respondent gender is male = 1)
refage	Numeric (adult respondent's age)

Table A.6: Logit regression model of A as a function of the covariates $\boldsymbol{X}$. Coefficients display log odds ratio. Stars indicate level of significance. Ethnicity (African America, Hispanic), Poverty Level and participation to other food programs (Food Stamp) appear to have the greatest and most significant impact on selection into treatment. Child's Age (the main moderator) is significant but of smaller magnitude.

	Dependent variable:
	$A_i = 1$
Child's Age	0.052^{***}
	(0.013)
Ref Age	0.001
	(0.005)
Child's Sex	-0.010
	(0.098)
African	1.047^{***}
	(0.123)
Hispanic	1.086^{***}
	(0.123)
Poverty Lvl	-1.407^{***}
	(0.110)
Suppl Nutr	0.244^{*}
	(0.140)
Food Stamp	1.117^{***}
	(0.131)
Food Security	0.345^{***}
	(0.122)
Insurance	-0.021
	(0.143)
Ref Sex	0.023
	(0.102)
Constant	-0.669^{**}
	(0.275)
Observations	2,330
Log Likelihood	$-1,260.824$
Akaike Inf. Crit.	2,545.647
Note:	*p<0.1; **p<0.05; ***p<0.01

Appendix B

Supplementary Material for Chapter 3

B.1 Shrinkage Bayesian Causal Forests

B.1.1 Perfectly known propensity scores

Table B.1 displays results obtained from Section 3.2.7 simulated exercise, where PS is assumed to be known and thus not estimated. Results are averaged over $H = 250$ simulations.

Model	Bias	$\sqrt{\text{PEHE}}$	95% Coverage	$(s_\pi \mid u_\pi)$
i) BCF	-0.03 ± 0.01	0.38 ± 0.02	0.95 ± 0.00	9.1%
ii) SH-BCF	-0.02 ± 0.01	0.31 ± 0.02	0.97 ± 0.00	95.5%
iii) SH-BCF (no PS)	-0.06 ± 0.01	0.39 ± 0.02	0.96 ± 0.01	-
iv) I-BCF ($k_{PS} = 50$)	-0.02 ± 0.01	0.31 ± 0.02	0.97 ± 0.00	96.9%
v) I-BCF ($k_{PS} = 100$)	-0.02 ± 0.01	0.31 ± 0.02	0.97 ± 0.00	96.9%

Table B.1: Bias, $\sqrt{\text{PEHE}}$, 95% Coverage and posterior splitting probability on the true $\pi(x_i)$ — $(s_\pi \mid u_\pi)$ — for: i) default BCF; ii) Shrinkage BCF; iii) Shrinkage BCF without the true $\pi(x_i)$; iv) informative prior BCF with $k_{PS} = 50$; v) informative prior BCF with $k_{PS} = 100$.

B.1.2 Computational advantage of DART

In this small experiment, we briefly illustrate some of the computational advantages of DART's backfitting MCMC versus default BART. To this end,

we compare on a purely predictive task three different specifications: i) default BART run for 6 000 MCMC draws (of which 4 000 burn-in); ii) long-chain BART run for 60, 000 MCMC draws (40 000 burn-in); iii) DART run for 6 000 MCMC draws (4 000 burn-in). The task is to predict Y_i given $P = 50$ predictors $\boldsymbol{X}_i$, of which only 5 are relevant, with $n = 500$. The predictors $\boldsymbol{X}_i$ are simulated from a Gaussian Copula where elements of the correlation matrix are $\Theta_{jk} = 0.3^{|j-k|} + 0.1\mathbb{I}(j \neq k)$. Half of the predictors are continuous and half binary. The outcome Y_i reads instead:

$$Y_i = 5 + 5\sin(\pi X_{i,1}) + 2.5(X_{i,2} - 0.5)^2 + 1.5|X_{i,3}| + 2X_{i,4}(X_{i,20} + 1) + \varepsilon_i\,,$$

where $\varepsilon_i \sim \mathcal{N}(0, 1)$.

	BART	long BART	DART
RMSE	2.11 ± 0.03	1.99 ± 0.03	1.74 ± 0.03
X_1	21.85 ± 0.20	21.77 ± 0.18	66.43 ± 1.30
X_2	18.03 ± 0.14	18.06 ± 0.13	43.94 ± 0.72
X_3	6.85 ± 0.10	6.81 ± 0.09	10.49 ± 0.28
X_4	11.18 ± 0.10	11.09 ± 0.09	24.04 ± 0.44
X_{20}	6.63 ± 0.07	6.51 ± 0.05	39.44 ± 1.14

Table B.2: Test set RMSE and average number of splits on the five relevant predictors, plus/minus 95% Monte Carlo standard error for: i) default BART; ii) long-chain BART; iii) DART.

The purpose of this exercise is to investigate whether the relative performance (measured with RMSE) of DART erodes with respect to BART run for dramatically long chain. Results displaying averaged RMSE for test set (we considered 70%-30% train-test split), in addition to the average number of splits on the 5 relevant predictors for each model are depicted in Table B.2. Results are averaged over $H = 500$ Monte Carlo replications. We can see that running BART for way longer chains results in improved performance over short-chain BART. This is due to the fact that BART's MCMC concentrates very slowly, while DART allows for much faster posterior concentration.

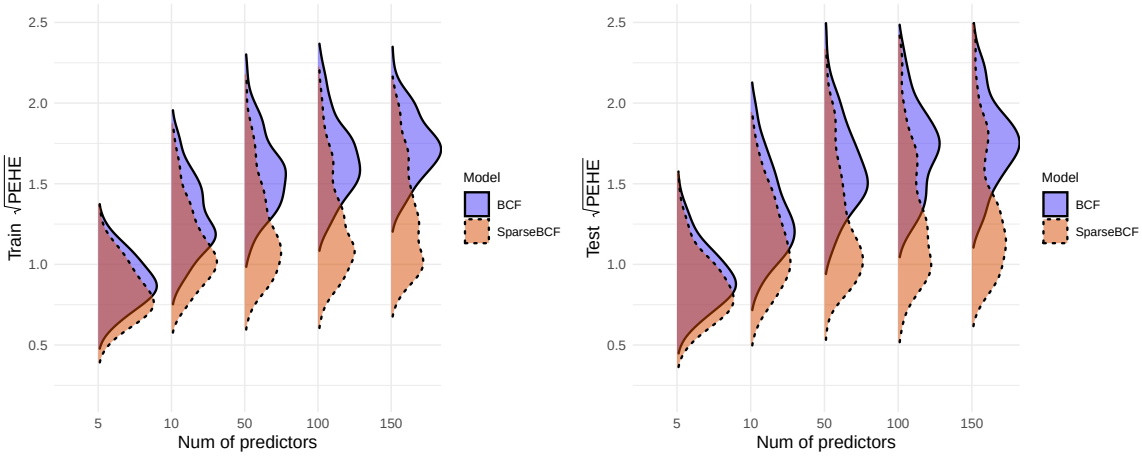


Figure B.1: Estimated train (left plot) and test (right plot) $\sqrt{\text{PEHE}}$ distributions generated by BCF and SH-BCF respectively, over an increasing number of predictors.

B.1.3 High-dimensional P

In this third additional simulated experiment we study what happens to BCF and SH-BCF with an increasing number of predictors P. To this end, we consider setups with $P \in \{5, 10, 50, 100, 150\}$ respectively. Sample size is fixed at $n = 250$, and we run $H = 200$ Monte Carlo replications for each different P. The covariates $\boldsymbol{X}_i$ are simulated from a Gaussian Copula where elements of the correlation matrix are $\Theta_{jk} = 0.3^{|j-k|} + 0.1\mathbb{I}(j \neq k)$. The DGP is the following:

$$
\begin{aligned}
\mu(\boldsymbol{x}_i) &= 3 + 1.5\sin(\pi X_{i,1}) + 0.5(X_{i,2} - 0.5)^2 + 1.5(2 - |X_{i,3}|) + X_{i,4}(X_{i,\frac{P}{2}} + 1) \ , \\
\tau(\boldsymbol{x}_i) &= 0.1 + 1|X_{i,1} - 1|(X_{i,\frac{P}{2}} + 2) \ , \\
\pi(\boldsymbol{x}_i) &= \Phi\big(-0.5 + 0.2X_{i,1} + 0.1X_{i,2} + 0.4X_{i,\frac{P}{2}} + \nu_i \big) \ , \\
A_i &\sim \text{Bernoulli}\big(\pi(\boldsymbol{x}_i)\big) \ , \\
Y_i &= \mu(\boldsymbol{x}_i) + \tau(\boldsymbol{x}_i)A_i + \varepsilon_i \ , \quad \text{where} \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \ ,
\end{aligned}
\tag{B.1}
$$

where: $\Phi(\cdot)$ is the standard Normal c.d.f.; $\nu_i \sim \text{Uniform}(0, 0.1)$ is uniform noise; error standard deviation is set to $\sigma = 0.5\,\hat{\sigma}_\mu$, where $\hat{\sigma}_\mu$ is the sample standard deviation of the simulated $\mu(\boldsymbol{x}_i)$.

A 70%-30% train-test set split is utilized. Results are shown in Table B.3, which depicts performance in terms of $\sqrt{\text{PEHE}}$, differentiated between train and test

sets. We can appreciate how, compared to SH-BCF, default BCF's performance deteriorates as P increases, suffering from the curse of dimensionality.

	BCF		**SH-BCF**	
P	Train	Test	Train	Test
5	0.91 ± 0.02	0.94 ± 0.03	0.83 ± 0.02	0.87 ± 0.03
10	1.30 ± 0.03	1.33 ± 0.04	1.12 ± 0.04	1.15 ± 0.04
50	1.57 ± 0.03	1.62 ± 0.04	1.23 ± 0.04	1.27 ± 0.05
100	1.66 ± 0.03	1.71 ± 0.04	1.26 ± 0.05	1.30 ± 0.05
150	1.74 ± 0.03	1.78 ± 0.04	1.32 ± 0.05	1.35 ± 0.06

Table B.3: Train and test set average $\sqrt{\text{PEHE}}$, plus/minus 95% Monte Carlo standard error, for BCF and SH-BCF with an increasing P.

B.1.4 Different types of sparse DGPs

We study the performance of BCF and SH-BCF on four different types of sparse DGPs. In particular, we consider a setting with fixed $n = 500$ and $P = 5$, where covariates are generated again from a Gaussian Copula with correlation matrix elements set to $\Theta_{jk} = 0.3^{|j-k|} + 0.1\mathbb{I}(j \neq k)$. We run BCF and SH-BCF for $H = 200$ Monte Carlo replications on each of the following four different versions of a DGP, according to what surface is generated as sparse:

1) **No Sparsity**. The first version features no sparsity at all, meaning that all the covariates are relevant for estimating every function of interest. The DGP reads:

$$
\begin{aligned}
\mu(\boldsymbol{x}_i) &= 3 + 1.5\sin(\pi X_{i,1}) + 0.5(X_{i,2} - 0.5)^2 + 1.5(2 - |X_{i,3}|) + 1.5X_{i,4}(X_{i,5} + 1) \ , \\
\tau(\boldsymbol{x}_i) &= 0.1 + 1|X_{i,1} - 1|(X_{i,5} + 2) - 0.4X_{i,3} + 0.6X_{i,2}X_{i,4} \ , \\
\pi(\boldsymbol{x}_i) &= \Phi\big(-0.2 + 0.8X_{i,1} - 0.1X_{i,2} + 0.1X_{i,3}X_{i,4} - 0.4X_{i,5} + \nu_i\big) \ , \qquad \text{(B.2)} \\
A_i &\sim \text{Bernoulli}\big(\pi(\boldsymbol{x}_i)\big) \ , \\
Y_i &= \mu(\boldsymbol{x}_i) + \tau(\boldsymbol{x}_i)A_i + \varepsilon_i \ , \quad \text{where} \quad \varepsilon_i \sim \mathcal{N}(0,1) \ ,
\end{aligned}
$$

where $\Phi(\cdot)$ is the standard Normal c.d.f. and $\nu_i \sim \text{Uniform}(0, 0.1)$ is uniform noise

2) **Sparse $\boldsymbol{\pi(\cdot)}$**. The second DGP is exactly the same as (B.2), but propensity score is a sparse surface, equal to $\pi(\boldsymbol{x}_i) = \Phi(-0.2 + 0.8X_{i,1} + \nu_i)$

3) **Sparse $\boldsymbol{\mu(x_i)}$**. The third DGP is the same as (B.2), but prognostic score is a sparse surface, equal to $\mu(\boldsymbol{x}_i) = 3 + 1.5(2 - |X_{i,3}|)$

4) **Sparse $\boldsymbol{\tau(x_i)}$**. Finally, the last DGP is the same as (B.2), but treatement effect is a sparse surface $\tau(\boldsymbol{x}_i) = 0.1 + 1|X_{i,1} - 1|$

Again, a 70%-30% train-test split is considered. Table B.4 shows BCF's and SH-BCF's results in terms of train and test $\sqrt{\text{PEHE}}$.

	BCF		**SH-BCF**	
DGP	Train	Test	Train	Test
Not Sparse	0.96 ± 0.02	0.98 ± 0.03	0.92 ± 0.02	0.95 ± 0.03
Sparse $\pi(\cdot)$	0.91 ± 0.02	0.93 ± 0.03	0.88 ± 0.02	0.93 ± 0.03
Sparse $\mu(\cdot)$	0.85 ± 0.02	0.87 ± 0.03	0.81 ± 0.02	0.83 ± 0.03
Sparse $\tau(\cdot)$	0.67 ± 0.02	0.67 ± 0.02	0.66 ± 0.02	0.66 ± 0.02

Table B.4: Train and test set $\sqrt{\text{PEHE}}$, plus/minus 95% Monte Carlo standard error, for BCF and SH-BCF on the four different version of sparse DGPs.

B.1.5 Fully sparse *vs* non-fully sparse DGP

In this last additional simulated study we compare BCF and SH-BCF again in two different scenarios with $n = 500$ and $P = 10$ correlated covariates (the first 5 continuous and the rest binary):

i) the first scenario is similar to the ones encountered in the previous sections and it is fully sparse, in that only 5 out of $P = 10$ covariates feature both in $\pi(\cdot)$ and either of $\mu(\cdot)$ and $\tau(\cdot)$:

$$\begin{aligned}
\mu(\boldsymbol{x}_i) &= 3 + 0.5(X_{i,1} - 0.5)^2 + 1.5(2 - |X_{i,2}|) + 1.5(X_{i,8} + 1) \ , \\
\tau(\boldsymbol{x}_i) &= 0.2 + 1|X_{i,9} - 1| - 0.4X_{i,10} \ , \\
\pi(\boldsymbol{x}_i) &= \Phi\big(-0.2 + 0.8X_{i,1} - 0.1X_{i,2} + 0.2X_{i,8} + 0.5X_{i,9}X_{i,10} + \nu_i \big) \ ,
\end{aligned}$$

Table B.5: Train and test set $\sqrt{\text{PEHE}}$, plus/minus 95% Monte Carlo standard error, for BCF and SH-BCF on fully sparse and non-fully sparse DGPs.

	BCF		**SH-BCF**	
DGP	Train	Test	Train	Test
Fully Sparse	0.48 ± 0.01	0.49 ± 0.01	0.36 ± 0.01	0.37 ± 0.01
Non-Fully Sparse	0.32 ± 0.01	0.31 ± 0.01	0.27 ± 0.01	0.27 ± 0.01

ii) the second scenario instead is not "fully" sparse, in that all of the $P = 10$ are relevant, but half of them enters $\pi(\cdot)$ while the remaining half features in either $\mu(\cdot)$ or $\tau(\cdot)$:

$$\mu(\boldsymbol{x}_i) = 3 + 0.5(X_{i,3} - 0.5)^2 + 1.5(2 - |X_{i,4}|) + 1.5(X_{i,6} + 1) \ ,$$
$$\tau(\boldsymbol{x}_i) = 0.2 + 1|X_{i,5} - 1| - 0.4X_{i,7} \ ,$$
$$\pi(\boldsymbol{x}_i) = \Phi\big(-0.2 + 0.8X_{i,1} - 0.1X_{i,2} + 0.2X_{i,8} + 0.5X_{i,9}X_{i,10} + \nu_i \big) \ ,$$

The results in terms of $\sqrt{\text{PEHE}}$ are gathered in table B.5. SH-BCF's performance gain relative to BCF are more pronounced in the first scenario as the DGP is fully sparse, given that $\{X_{i,3}, X_{i,4}, X_{i,5}, X_{i,6}, X_{i,7}\}$ do not appear in any of the functions of interest. In the second scenario, the DGP employs all the covariates in either the outcome function or the propensity score function, so the gap in performance between SH-BCF and BCF slightly deteriorates, with SH-BCF being marginally better nonetheless for its computational advantages.

B.2 Variables included in the analysis

Table B.6 here below provide a full list of variables used for the analysis in Section 6.

Table B.6: Variables from the Infant Health and Development Program (IHDP)

Variable	**Description**	**Type**
iq	Score in IQ test (outcome Y)	Numeric
treat	Participation to the program (treatment A)	Binary

bw	Child's weight at birth (in grams)	Numeric
momage	Mother's age	Numeric
nnhealth	Neo-natal health index	Numeric
birth.o	Child's order of birth	Numeric
parity	Number of children	Numery
moreprem	Number of children born prematurely	Numeric
cigs	Smoke during pregnancy	Numeric
alcohol	Drinks during pregnancy	Numeric
ppvt.imp	Mother's PPVT test result 1 year post birth	Numeric
bw_2000	Birth weight above/below 2kg	Binary
female	Child is a female	Binary
mlt.birt	Number of multiple births	Ordinal
b.marry	Marital status at birth	Binary
livwho	What family member lives with the child	Ordinal
language	Language spoken at home	Binary
whenpren	Trimester when prenatal care started	Ordinal
drugs	Drug use during pregnancy	Binary
othstudy	Participating in other studies at the same time	Binary
site1	Site number 1	Binary
⋮	⋮	⋮
site8	Site number 8	Binary
momblack	Mother's ethnicity black	Binary
momhisp	Mother's ethnicity hispanic	Binary
momwhite	Mother's ethnicity white	Binary
workdur.imp	Mother worked during pregnancy	Binary
momed4F	Mother's education level	Ordinal

B.3 Interpretable Deep Causal Learning

B.3.1 Data Generating Process

In this appendix section we briefly describe the data generating process utilized for the ICNN simulated experiment. We generated $n = 2000$ data points on $P = 10$ correlated covariates, of which 5 continuous and 5 binary, drawn from a Gaussian Copula $C_{\Theta}^{\text{Gauss}}(u) = \Phi_{\Theta}\big(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_P)\big)$, where the covariance matrix is such that $\Theta_{jk} = 0.1^{|j-k|} + 0.1\mathbb{I}(j \neq k)$. The data generating process is fully described by the following quantities:

$$
\begin{aligned}
\mu(\boldsymbol{x}_i) = \;& 6 + 0.3\exp(X_{i,1}) + 1X_{i,2}^2 + 1.5|X_{i,3}| + 0.8X_{i,4} \;, \\
\tau(\boldsymbol{x}_i) = \;& 3 + 0.8X_{i,1}^2 \;, \\
\pi(\boldsymbol{x}_i) = \;& \Lambda\left(-1.5 + 0.5X_{i,1} + \frac{\nu_i}{10}\right) \;, \\
A_i \sim \;& \text{Bernoulli}\big(\pi(\boldsymbol{x}_i)\big) \;, \\
Y_i = \;& \mu(\boldsymbol{x}_i) + \tau(\boldsymbol{x}_i)A_i + \varepsilon_i \;, \quad \text{where} \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \;,
\end{aligned}
\tag{B.3}
$$

where: $\Lambda(\cdot)$ is the logistic cumulative distribution function; the error's standard deviation is $\sigma^2 = 0.5$; and $\nu_i \sim \text{Uniform}(0, 1)$.

Appendix C

Supplementary Material for Chapter 4

C.1 Data Generating Processes

We hereby describe the causal data generating processes in the simulated examples of the paper (Section 3.2 and Section 5.1).

C.1.1 One covariate example

For the simple one-covariate example in Section 3.2 (Figure 2), where we discuss the benefits of multitask counterfactual learning, we generated $n = 300$ data points from one, uniformly distributed covariate, $X_i \sim \text{Uniform}(-3, 3)$. Then we generated a binary action variable $A_i \sim \text{Bernoulli}\big(p(A_i = 1|x_i)\big)$, where $p(A_i = 1|x_i) = \Phi\big(0.2 + X_i\big)$ and $\Phi(\cdot)$ is the standard normal cdf. Finally, the two counterfactual outcome surfaces were generated as $f_0(x_i) = 2 + 0.3 \exp X_i$ and $f_1(x_i) = 3 + f_0(x_i)$, with the final outcome being $Y = f_0(x_i) + \tau(x_i)A_i + \varepsilon_i$ where $\tau(x_i) = f_1(x_i) - f_0(x_i)$ is the CATE function and $\varepsilon_i \sim \mathcal{N}(0, 0.75^2)$.

C.1.2 Simulated

The causal data generating process for the simulated experiment of Section 5.1 is described as follows. The P covariates are generated from a uniform distribution $X_{i,j} \sim \text{Unif}(-3, 3)$ for $j \in \{1, ..., P\}$ and $i \in \{1, ..., n\}$. The action allocation policy is simulated according to a multinomial distribution where

the probabilities of being assigned to action $A_i = a$ are generated as a softmax function of the covariates $p(A_i = a|\boldsymbol{X}_i = \boldsymbol{x}_i) = \exp\{X_i\boldsymbol{\beta}_a\} \,/\, \sum_{a\in\mathcal{A}} \exp\{X_i\boldsymbol{\beta}_a\}$, where $\boldsymbol{\beta}_a$ is an action-specific P-dimensional sparse vector of action-specific coefficients defined as follow:

$$
\begin{aligned}
\boldsymbol{\beta}_1 &= \begin{bmatrix} -1 & -0.8 & -0.1 & -0.1 & 0 & ... & 0 \end{bmatrix}, \\
\boldsymbol{\beta}_2 &= \begin{bmatrix} 0 & 0 & 1 & 0.8 & 0.2 & 0 & ... & 0 \end{bmatrix}, \\
\boldsymbol{\beta}_3 &= \begin{bmatrix} 1.5 & -0.8 & -0.1 & -0.1 & 0 & ... & 0 \end{bmatrix}, \\
\boldsymbol{\beta}_4 &= \begin{bmatrix} -1 & -0.8 & -0.1 & -0.1 & 0 & ... & 0 \end{bmatrix}.
\end{aligned}
$$

Thus A_i is drawn from a multinomial with vector probabilities parameter $\boldsymbol{p}(A_i = a|\boldsymbol{X}_i = \boldsymbol{x}_i)$. The $M = 2$ action-specific correlated counterfactual outcomes $\boldsymbol{Y}_i \mid do(A_i = a)$ instead are generated as

$$
\begin{aligned}
&\boldsymbol{Y}_i \mid do(A_i = a) = \boldsymbol{f}_{\boldsymbol{Y}\,a}(\boldsymbol{X}_i) + \boldsymbol{\varepsilon}_i\,, \quad \boldsymbol{\varepsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{\varepsilon_i}), \quad \text{where:} \\
&f_{Y1_1} = 3 + 0.4X_0X_1 - 0.3X_2^2 + 0.2\exp(X_3) + 0.6\sin(X_4) \\
&f_{Y1_2} = -1 + f_{Y1_1} + 0.1X_5 \\
&f_{Y1_3} = 1 + f_{Y1_1} + 0.3X_5 \\
&f_{Y1_4} = 0.5 + f_{Y1_1} + 0.5X_6 \\
&f_{Y2_1} = 1 + 0.2X_0X_1 - 0.2X_2^2 + 0.1\exp(X_3) \\
&f_{Y2_2} = -2 + f_{Y2_1} + 0.2X_5 \\
&f_{Y2_3} = 2 + f_{Y2_1} + 0.4X_5 \\
&f_{Y2_4} = 1 + f_{Y2_1} + 0.5X_6
\end{aligned}
$$

and where $\text{diag}(\Sigma_\varepsilon) = [\sigma_1, ..., \sigma_4]$, with $\sigma_1 = ... = \sigma_4 = 0.5$, and off-diagonal elements are 0. Finally, we briefly describe the main specifications of the methods compared. The GP models (GP, CounterGP and MOGP) all employed a RBF base kernel, while the DKL models employed a three [50, 50, 2] hidden layers feedforward neural network before the GP ∞-layer, which itself employs

a RBF base kernel. The multitask and multioutput models (both GPs and DKLs) all make use of the Intrinsic Coregionalization Model (ICM), such that $K(\boldsymbol{x}_i, \boldsymbol{x}_i') = B_Y \otimes B_A \otimes K_q(\boldsymbol{x}_i, \boldsymbol{x}_i')$. All model were optimized through the Adam solver.

C.2 The Job Training Data

The Job Training data (LaLonde, 1986) are a popular case study in the causal inference literature. They comprise a portion of data pertaining to a randomized experiment and a portion of observational data, with the randomized experiment featuring 297 treated and 425 control units, while the observational data being of 2490 control units only. Given the randomized subsample of the data, we can obtain an unbiased estimate (computed on the randomized units only) for the Average Treatment Effect on the Treated group (ATT) as $\text{ATT} = T^{-1}\sum_{i=1}^{T_e} y_i - C^{-1}\sum_{i=1}^{C} y_i$, where T and C are the number of treated and control units in the experimental data, and treat this as the ground truth for estimating performance of the methods; and also for the policy risk measure $\mathcal{R}_{\text{pol}} = 1 - \big[\mathbb{E}\big(Y|do(A_i = 1), \pi(\boldsymbol{x}_i) = 1\big)p(\pi(\boldsymbol{x}_i) = 1) + \mathbb{E}\big(Y|do(A_i = 0), \pi(\boldsymbol{x}_i) = 0\big)p(\pi(\boldsymbol{x}_i) = 0)\big]$.

A brief overview on the specifications of the models employed follows. All GPs employ RBF base kernel (also DKL's specifications in the last hidden layer). DKL and CounterDKL deep NN structure features three [10, 5, 2] hidden layers. The AutoEncoder deep structure employed for the "AutoEnc + GP" and "AutoEnc + CounterGP" models similarly learns a 2-dimensional encoded lower-dimensional representation, where the encoder has two [10, 5] hidden layers before the 2-dim representation and the decoder has [5, 10] hidden layers before the reconstruction loss.

C.3 Marginal Likelihood Maximization in Multioutput Deep Kernels

In the multitask deep kernel learning class of models, the parameter space $\Theta = (W, \phi, B)$ is made of the deep neural network's weights W, the base kernel's hyperparameters ϕ (variance, lengthscales, etc.) and the coregionalization matrix B entries. These parameters are learnt jointly by maximizing the log-marginal likelihood $\mathcal{L}$ at the end of the GP layer. Using the chain rule, the derivatives are:

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial K_\phi} \frac{\partial K_\phi}{\partial g(\boldsymbol{x}, W)} \frac{\partial g(\boldsymbol{x}, W)}{\partial W}$$

$$\frac{\partial \mathcal{L}}{\partial \phi} = \frac{\partial \mathcal{L}}{\partial K_\phi} \frac{\partial K_\phi}{\partial \phi}$$

$$\frac{\partial \mathcal{L}}{\partial B} = \frac{\partial \mathcal{L}}{\partial K} \frac{\partial K}{\partial B}$$

where $g(\boldsymbol{x}, W)$ is the function mapping the inputs to the lower representation space parametrized by W, K_ϕ is the base kernel and $K(\cdot) = B \otimes K_\phi(\cdot)$ is the coregionalized kernel.

C.4 Additional Simulated Experiments

Finally, we describe and present results on a few additional simulated examples that we conducted to assess CounterDKL performance compared to some other specifications seen in Section 5.2, on datasets with varying sample size, predictor space and action space dimensions. In particular, following Dudík et al. (2011) and Farajtabar et al. (2018), we make use of some of the popular datasets for classification in the open-source UCI Machine Learning Repository (`https://archive.ics.uci.edu/ml/index.php`), by transforming the classification task in a causal Off-Policy Evaluation task in the following way. Each dataset is equipped with a pair of covariates $\boldsymbol{X}_i$ and classification labels L_i. We view the classification labels L_i as our discrete actions $L_i = A_i$, and consequently

generate the action-specific outcome Y_{a_i} as function of the covariates as follows:

$$Y_{a_i} = \exp\{\boldsymbol{X_i}\boldsymbol{\beta_a}\} + \varepsilon_i, \quad \text{where } \mathcal{N}(0, 0.5)$$

and $\boldsymbol{\beta_a}$ is a P-dimensional vector of action-specific coefficients, where entries are $\{0.4, 0.2, 0.0\}$ sampled from a Multinomial$(0.6, 0.25, 0.15)$, with replacement. The datasets utilized are summarized in Table C.1 in terms of sample size n, number of covariates P and number of actions. We compare GP, CounterGP, DKL and CounterDKL models on an Off-Policy Evaluation task, where we evaluate the uniformly at random generated policy, via the absoulte regret or risk measure, defined as $\mathbb{E}\big[\ |\ \mathcal{V}(\pi_e) - \hat{\mathcal{V}}(\pi_e)\ |\ \big]$. All models employ a RBF base kernel, either directly on the inputs or on the lower dimensional layer. Results averaged over $B = 20$ replications of the experiments for each dataset are gathered in Table C.2.

Data	$\boldsymbol{n}$	$\boldsymbol{P}$	**# actions**
indian	573	10	2
heart	270	13	2
yeast	1484	8	10
contracept	1473	9	3

Table C.1: UCI datasets characteristics.

	GP	CounterGP	DKL	CounterDKL
indian	0.390	0.392	0.376	**0.347**
heart	2.553	1.076	0.433	**0.410**
yeast	0.534	0.657	1.3144	**0.081**
contracep	0.339	**0.003**	0.008	0.007

Table C.2: OPE absolute regret on UCI datasets. Bold denotes best performance.

Bibliography

M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. E. Hinton. Neural additive models: Interpretable machine learning with neural nets. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, volume 34, pages 4699–4711, 2021.

A. Alaa and M. van der Schaar. Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 129–138, 2018.

A. Alaa and M. Van Der Schaar. Validating causal inference models via influence functions. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 191–201, 2019.

A. M. Alaa and M. van der Schaar. Bayesian inference of individualized treatment effects using multi-task Gaussian Processes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3427–3435, 2017.

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous

response data. *Journal of the American Statistical Association*, 88(422): 669–679, 1993.

M. A. Alvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3): 195–266, 2012.

J. Angrist and J.-S. Pischke. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, 1 edition, 2009.

J. D. Angrist, G. W. Imbens, and D. B. Rubin. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*, 91(434):444–455, 1996.

M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization, 2020.

P. M. Aronow and C. Samii. Estimating average causal effects under general interference, with application to a social network experiment. *The Annals of Applied Statistics*, 11(4):1912 – 1947, 2017.

S. Athey and G. Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016. ISSN 0027-8424.

S. Athey and S. Wager. Estimating treatment effects with causal forests: An application. *Observational Studies*, 5(2):37–51, 2019.

S. Athey and S. Wager. Policy Learning With Observational Data. *Econometrica*, 89(1):133–161, January 2021.

S. Athey, D. Eckles, and G. W. Imbens. Exact p-values for network interference. *Journal of the American Statistical Association*, 113(521):230–240, 2018.

A. Balke and J. Pearl. Bounds on treatment effects from studies with imperfect compliance. *Journal of the American Statistical Association*, 92(439):1171–1176, 1997.

H. Bang and J. M. Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005.

E. Bareinboim and J. Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

E. Bareinboim, A. Forney, and J. Pearl. Bandits with unobserved confounders: A causal approach. In *Advances in Neural Information Processing Systems 29*, volume 28, 2015.

A. P. Blaisdell and R. R. Miller. *Causal Learning*, pages 520–523. Springer US, Boston, MA, 2012.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, page 1613–1622, 2015.

B. Bohn, C. Rieger, and M. Griebel. A representer theorem for deep kernel learning. *J. Mach. Learn. Res.*, 20(1):2302–2333, 2019.

E. V. Bonilla, K. Chai, and C. Williams. Multi-task Gaussian Process prediction. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001. ISSN 0885-6125.

L. Breiman. *Classification and regression trees*. Routledge, 2017.

M. A. Brookhart, S. Schneeweiss, K. J. Rothman, R. J. Glynn, J. Avorn, and T. Stürmer. Variable selection for propensity score models. *American journal of epidemiology*, 163(12):1149–1156, 2006.

J. Brooks-Gunn, F. ruey Liaw, and P. K. Klebanov. Effects of early intervention on cognitive function of low birth weight preterm infants. *The Journal of Pediatrics*, 120(3):350–359, 1992. ISSN 0022-3476.

J. Brooks-Gunn, C. McCarton, P. Casey, M. McCormick, C. Bauer, J. Bernbaum, J. Tyson, M. Swanson, F. Bennett, and D. Scott. Early intervention in low-birth-weight premature infants. results through age 5 years from the infant health and development program. *JAMA*, 272(16):1257—1262, October 1994. ISSN 0098-7484.

A. Caron, G. Baio, and I. Manolopoulou. Estimating individual treatment effects using non-parametric regression models: A review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 185(3):1115–1149, 2022a. doi: https://doi.org/10.1111/rssa.12824.

A. Caron, G. Baio, and I. Manolopoulou. Shrinkage Bayesian Causal Forests for heterogeneous treatment effects estimation. *Journal of Computational and Graphical Statistics*, pages 1–13, 2022b. doi: 10.1080/10618600.2022.2067549.

A. Caron, G. Baio, and I. Manolopoulou. Interpretable deep causal learning for moderation effects. In *ICML 2022, 2nd Interpretable Machine Learning for Healthcare Workshop*, 2022c. URL `https://arxiv.org/abs/2206.10261`.

A. Caron, I. Manolopoulou, and G. Baio. Counterfactual learning with multioutput deep kernels. *Transactions on Machine Learning Research*, 2022d. URL `https://openreview.net/forum?id=iGREAJdULX`.

C. M. Cassel, C. E. Sarndal, and J. H. Wretman. Some results on generalized difference estimation and generalized regression estimation for finite populations. *Biometrika*, 63(3):615–620, 1976.

K. C. G. Chan, S. C. P. Yam, and Z. Zhang. Globally efficient non-parametric inference of average treatment effects by empirical balancing calibration weighting. *J. R. Stat. Soc. Series B (Statistical Methodology)*, 78(3):673–700, 2016.

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

K.-W. Cheung, J. T. Kwok, M. H. Law, and K.-C. Tsui. Mining customer product ratings for personalized marketing. *Decision Support Systems*, 35 (2):231–243, 2003.

S. Chiappa. Path-specific counterfactual fairness. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7801–7808, 2019.

H. Chipman. Bayesian variable selection with related predictors. *Canadian Journal of Statistics*, 24(1):17–36, 1996.

H. A. Chipman, E. I. George, and R. E. McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998. ISSN 01621459.

H. A. Chipman, E. I. George, and R. E. McCulloch. BART: Bayesian additive regression trees. *Ann. Appl. Stat.*, 4(1):266–298, 03 2010.

F. S. Collins and H. Varmus. A new initiative on precision medicine. *New England Journal of Medicine*, 372(9):793–795, 2015.

Y. Cui, H. Pu, X. Shi, W. Miao, and E. Tchetgen Tchetgen. Semiparametric proximal causal inference. *Journal of the American Statistical Association*, pages 1–12, 2023.

A. P. Dawid. Causal inference without counterfactuals. *Journal of the American Statistical Association*, 95(450):407–424, 2000. ISSN 01621459.

A. P. Dawid. Statistical causality from a decision-theoretic perspective. *Annual Review of Statistics and Its Application*, 2(1):273–303, 2015.

D. G. Denison, B. K. Mallick, and A. F. Smith. A bayesian cart algorithm. *Biometrika*, 85(2):363–377, 1998.

V. Dorie, J. Hill, U. Shalit, M. Scott, and D. Cervone. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statist. Sci.*, 34(1):43–68, 02 2019.

M. Drton and M. H. Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393, 2017.

M. Dudík, J. Langford, and L. Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 1097–1104, 2011.

M. Dudík, D. Erhan, J. Langford, and L. Li. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014. ISSN 08834237, 21688745.

Q. Fan, Y.-C. Hsu, R. P. Lieli, and Y. Zhang. Estimation of conditional average treatment effects with high-dimensional data. *Journal of Business & Economic Statistics*, 0(0):1–15, 2020.

M. Farajtabar, Y. Chow, and M. Ghavamzadeh. More robust doubly robust off-policy evaluation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1447–1456, 2018.

L. Forastiere, E. M. Airoldi, and F. Mealli. Identification and estimation of treatment and interference effects in observational studies on networks. *Journal of the American Statistical Association*, 116(534):901–918, 2021.

D. J. Foster and V. Syrgkanis. Orthogonal statistical learning, 2020.

J. C. Foster, J. m. G. Taylor, and S. J. Ruberg. Subgroup identification from randomized clinical trial data. *Statistics in medicine*, 30 24:2867–80, 2011.

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1050–1059, 2016.

J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 7587–7597, 2018.

H. Geffner, R. Dechter, and J. Y. Halpern, editors. *Probabilistic and Causal Inference: The Works of Judea Pearl*, volume 36. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022.

Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015.

S. Ghosal and A. van der Vaart. *Fundamentals of Nonparametric Bayesian Inference*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2017. doi: 10.1017/9781139029834.

C. Glymour, K. Zhang, and P. Spirtes. Review of Causal Discovery Methods Based on Graphical Models. *Frontiers in Genetics*, 10, 2019.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

D. P. Green and H. L. Kern. Modeling Heterogeneous Treatment Effects in Survey Experiments with Bayesian Additive Regression Trees. *Public Opinion Quarterly*, 76(3):491–511, 09 2012. ISSN 0033-362X.

P. R. Hahn, C. M. Carvalho, D. Puelz, and J. He. Regularization and confounding in linear regression for treatment effect estimation. *Bayesian Anal.*, 13(1):163–182, 03 2018.

P. R. Hahn, J. S. Murray, and C. M. Carvalho. Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects. *Bayesian Anal.*, 2020. Advance publication.

W. L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.

S. M. Hammer, D. A. Katzenstein, M. D. Hughes, H. Gundacker, R. T. Schooley, R. H. Haubrich, W. K. Henry, M. M. Lederman, J. P. Phair, M. Niu, M. S. Hirsch, and T. C. Merigan. A trial comparing nucleoside monotherapy with combination therapy in hiv-infected adults with CD4 cell counts from 200 to 500 per cubic millimeter. *N. Engl. J. Med.*, 335:1081–1090, 1996.

J. Hartford, G. Lewis, K. Leyton-Brown, and M. Taddy. Deep IV: A flexible approach for counterfactual prediction. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1414–1423, 2017.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

T. J. Hastie. Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge, 2017.

W. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97, 1970.

J. He, S. Yalov, and P. R. Hahn. Xbart: Accelerated bayesian additive regression trees. In K. Chaudhuri and M. Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1130–1138. PMLR, 16–18 Apr 2019.

J. J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47 (1):153–161, 1979. ISSN 00129682, 14680262.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, 2013.

J. Hill and Y.-S. Su. Assessing lack of common support in causal inference using Bayesian nonparametrics: Implications for evaluating the effect of

breastfeeding on children's cognitive outcomes. *The Annals of Applied Statistics*, 7(3):1386 – 1420, 2013.

J. L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

K. Hirano, G. W. Imbens, and G. Ridder. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4): 1161–1189, 2003.

R. Hodson. Precision medicine. *Nature*, 547(7619), 2016.

P. W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

K. Hornik, M. Stinchcombe, and H. White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5):551–560, 1990.

D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

M. G. Hudgens and M. E. Halloran. Toward causal inference with interference. *J. Am. Stat. Assoc.*, 103:832–842, 2008.

K. Imai and D. A. van Dyk. Causal inference with general treatment regimes. *Journal of the American Statistical Association*, 99(467):854–866, 2004.

G. W. Imbens. Nonparametric estimation of average treatment effects under exogeneity: A review. *The Review of Economics and Statistics*, 86(1):4–29, 2004.

G. W. Imbens and D. B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction.* Cambridge University Press, 2015.

P. E. Jacob, L. M. Murray, C. C. Holmes, and C. P. Robert. Better together? statistical learning in models made of modules, 2017.

N. Jiang and L. Li. Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 652–661, 20–22 Jun 2016.

F. D. Johansson, U. Shalit, and D. Sontag. Learning representations for counterfactual inference. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, page 3020–3029, 2016.

N. Kallus. Balanced policy evaluation and learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8909–8920, 2018.

N. Kallus. More efficient policy learning via optimal retargeting. *Journal of the American Statistical Association*, 116(534):646–658, 2021.

M. Kato, M. Uehara, and S. Yasui. Off-policy evaluation and learning for external validity under a covariate shift, 2020.

G. King and R. Nielsen. Why propensity scores should not be used for matching. *Political Analysis*, 27(4):435–454, 2019.

T. Kitagawa and A. Tetenov. Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616, 2018.

M. C. Knaus, M. Lechner, and A. Strittmatter. Machine learning estimation of heterogeneous causal effects: Empirical Monte Carlo evidence. *The Econometrics Journal*, 24(1):134–161, 2020.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

S. Künzel, J. Sekhon, P. Bickel, and B. Yu. Meta-learners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116, 06 2017.

B. Lakshminarayanan, D. Roy, and Y. W. Teh. Particle gibbs for bayesian additive regression trees. In *Artificial Intelligence and Statistics*, pages 553–561. PMLR, 2015.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6405–6416. Curran Associates Inc., 2017.

R. J. LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American Economic Review*, 76:604–620, 1986.

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

A. R. Linero. Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, 113(522): 626–636, 2018.

A. R. Linero and Y. Yang. Bayesian regression tree ensembles that adapt to smoothness and sparsity. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5):1087–1110, 2018.

C. Louizos, U. Shalit, J. Mooij, D. Sontag, R. Zemel, and M. Welling. Causal effect inference with deep latent-variable models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*,

NIPS'17, page 6449–6459, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

M. Lu, S. Sadiq, D. J. Feaster, and H. Ishwaran. Estimating individual treatment effect in observational data using random forest methods. *Journal of Computational and Graphical Statistics*, 27(1):209–219, 2018.

T. Lumley, R. A. Kronmal, and S. Ma. Relative risk regression in medical research: Models, contrasts, estimators, and algorithms. *UW Biostatistics Working Paper Series*, 2006.

Y. Ma and V. Tresp. Causal inference under networked interference and intervention policy enhancement. In A. Banerjee and K. Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pages 3700–3708, 2021.

C. F. Manski. Nonparametric bounds on treatment effects. *The American Economic Review*, 80(2):319–323, 1990.

A. Mastouri, Y. Zhu, L. Gultchin, A. Korba, R. Silva, M. Kusner, A. Gretton, and K. Muandet. Proximal causal learning with kernels: Two-stage estimation and moment restriction. In *International conference on machine learning*, pages 7512–7523. PMLR, 2021.

C. McCarton, J. Brooks-Gunn, I. Wallace, C. Bauer, F. Bennett, J. Bernbaum, R. Broyles, P. Casey, M. McCormick, D. Scott, J. Tyson, J. Tonascia, and C. Meinert. Results at age 8 years of early intervention for low-birth-weight premature infants. the infant health and development program. *JAMA*, 277 (2):126—132, January 1997. ISSN 0098-7484.

M. McCormick. The contribution of low birth weight to infant mortality and childhood morbidity. *The New England journal of medicine*, 312(2):82—90, January 1985. ISSN 0028-4793.

M. C. McCormick, S. L. Gortmaker, and A. M. Sobol. Very low birth weight children: Behavior problems and school difficulty in a national sample. *The Journal of Pediatrics*, 117(5):687–693, 1990. ISSN 0022-3476.

L.-A. McNutt, C. Wu, X. Xue, and J. P. Hafner. Estimating the Relative Risk in Cohort Studies and Clinical Trials of Common Outcomes. *American Journal of Epidemiology*, 157(10):940–943, 05 2003.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

D. Milios, R. Camoriano, P. Michiardi, L. Rosasco, and M. Filippone. Dirichlet-based gaussian processes for large-scale calibrated classification. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 6008–6018, 2018.

T. P. Morris, I. R. White, and M. J. Crowther. Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, 38(11):2074–2102, 2019.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

S. A. Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–355, 2003.

J. S. Murray. Log-linear bayesian additive regression trees for multinomial logistic and count regression models. *Journal of the American Statistical Association*, 0(ja):1–35, 2020.

W. K. Newey and J. L. Powell. Instrumental variable estimation of nonparametric models. *Econometrica*, 71(5):1565–1578, 2003.

J. Neyman. Sur les applications de la theorie des probabilites aux experiences agricoles: Essai des principes. *Reprinted in English in Statistical Science*, 1923.

X. Nie and S. Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 09 2020.

X. Nie, E. Brunskill, and S. Wager. Learning when-to-treat policies. *Journal of the American Statistical Association*, 116(533):392–409, 2021.

S. W. Ober, C. E. Rasmussen, and M. van der Wilk. The promises and pitfalls of deep kernel learning. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161, pages 1206–1216, 2021.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

T. Pearce, F. Leibfried, and A. Brintrup. Uncertainty in neural networks: Approximately bayesian ensembling. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pages 234–244, 2020.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009a.

J. Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3 (none):96 – 146, 2009b. doi: 10.1214/09-SS057. URL `https://doi.org/10.1214/09-SS057`.

J. Pearl. Remarks on the method of propensity score. *Statistics in Medicine*, 28(9):1415–1416, 2009c.

J. Pearl. Invited commentary: understanding bias amplification. *American journal of epidemiology*, 174(11):1223–1227, 2011.

J. Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM '18*, 2018.

J. Pearl and D. Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, Inc., USA, 1st edition, 2018.

J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012, 2016a.

J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 78(5):947–1012, 2016b.

J. Peters, D. Janzing, and B. Schlkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, 2017.

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

M. T. Pratola. Efficient Metropolis–Hastings Proposal Mechanisms for Bayesian Regression Tree Models. *Bayesian Analysis*, 11(3):885 – 911, 2016.

M. T. Pratola, H. A. Chipman, E. I. George, and R. E. McCulloch. Heteroscedastic BART via multiplicative regression trees. *Journal of Computational and Graphical Statistics*, 29(2):405–417, 2020.

M. Qian and S. A. Murphy. Performance guarantees for individualized treatment rules. *Ann. Statist.*, 39(2):1180–1210, 04 2011.

J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6 (65):1939–1959, 2005.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

H. Reichenbach. *The Direction of Time*. Dover Publications, 1956.

T. S. Richardson, J. M. Robins, and L. Wang. On modeling and estimation for the relative risk and risk difference. *Journal of the American Statistical Association*, 112(519):1121–1130, 2017.

J. M. Robins. Optimal structural nested models for optimal sequential decisions. In *Proceedings of the Second Seattle Symposium in Biostatistics: analysis of correlated data*, pages 189–326. Springer, 2004.

P. M. Robinson. Root-n-consistent semiparametric regression. *Econometrica*, 56(4):931–954, 1988. ISSN 00129682, 14680262.

V. Rocková, S. Van der Pas, et al. Posterior concentration for bayesian regression trees and forests. *Annals of Statistics*, 48(4):2108–2131, 2020.

P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 04 1983. ISSN 0006-3444.

P. R. Rosenbaum and D. B. Rubin. Reducing bias in observational studies using subclassification on the propensity score. *J. Am. Stat. Assoc.*, 79: 516–524, 1984.

P. R. Rosenbaum and D. B. Rubin. Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *Am. Stat.*, 39:33–38, 1985.

V. Ročková and E. Saha. On theory for BART. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89, pages 2839–2848, 2019.

D. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

D. B. Rubin. Matching to remove bias in observational studies. *Biometrics*, 29 (1):159–183, 1973.

D. B. Rubin. Bayesian inference for causal effects: The role of randomization. *Ann. Statist.*, 6(1):34–58, 01 1978.

F. Sävje, P. Aronow, and M. Hudgens. Average treatment effects in the presence of unknown interference. *Annals of statistics*, 49(2):673, 2021.

B. Schölkopf. Causality for machine learning. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pages 765–804. Association for Computing Machinery, 2022.

A. Schuler, M. Baiocchi, R. Tibshirani, and N. Shah. A comparison of methods for model selection when estimating individual treatment effects, 2018.

P. J. Schulte, A. A. Tsiatis, E. B. Laber, and M. Davidian. Q-and a-learning methods for estimating optimal dynamic treatment regimes. *Statistical Science*, 29(4):640–661, 2014. ISSN 08834237, 21688745.

B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Towards causal representation learning, 2021.

U. Shalit, F. D. Johansson, and D. Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3076–3085, 2017.

L. S. Shapley. A value for n-person games. In *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, 1953.

S. Sivaganesan, P. Müller, and B. Huang. Subgroup finding via bayesian additive regression trees. *Statistics in Medicine*, 36(15):2391–2403, 2017.

R. Sparapani, C. Spanbauer, and R. McCulloch. Nonparametric machine learning and efficient computation with bayesian additive regression trees: the bart r package. *Journal of Statistical Software*, 97:1–66, 2021.

J. Splawa-Neyman, D. M. Dabrowska, and T. P. Speed. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, 5(4):465–472, 1990.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

J. Starling, J. Murray, P. Lohr, A. Aiken, C. Carvalho, and J. Scott. Targeted smooth bayesian causal forests: An analysis of heterogeneous treatment effects for simultaneous versus interval medical abortion regimens over gestation, 05 2019.

C. J. Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, 13(2):689–705, 1985. ISSN 00905364.

E. A. Stuart. Matching Methods for Causal Inference: A Review and a Look Forward. *Statistical Science*, 25(1):1 – 21, 2010.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

E. Tchetgen and T. VanderWeele. On causal inference in the presence of interference. *Stat. Methods Med. Res.*, 21:55–75, 2010.

E. J. T. Tchetgen, A. Ying, Y. Cui, X. Shi, and W. Miao. An introduction to proximal causal learning, 2020.

Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, volume R5, pages 333–340, 2005.

P. Thomas and E. Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2139–2148. PMLR, 2016.

J. Tian and J. Pearl. A general identification condition for causal effects. In *Eighteenth National Conference on Artificial Intelligence*, page 567–573, 2002.

M. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, volume 5, pages 567–574, 2009.

P. Toulis and E. Kao. Estimation of causal peer influence effects. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, volume 28, page III–1489–III–1497, 2013.

M. Traskin and D. S. Small. Defining the study population for an observational study to ensure sufficient overlap: A tree approach. *Statistics in Biosciences*, 3:94–118, 2011.

V. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2 edition, 2000.

S. Verma, V. Boonsanong, M. Hoang, K. E. Hines, J. P. Dickerson, and C. Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.

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

Y. Wang and D. M. Blei. The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528):1574–1596, 2019.

A. G. Wilson and H. Nickisch. Kernel interpolation for scalable structured gaussian processes (KISS-GP). In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, page 1775–1784, 2015.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In A. Gretton and C. C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 370–378. PMLR, 2016.

S. Wright. On the nature of size factors. *Genetics*, 3(4):367–374, 1918.

S. Wright. Correlation and causation. *Journal of Agricultural Research*, 20(7): 557–585, 1921.

L. Yao, S. Li, Y. Li, M. Huai, J. Gao, and A. Zhang. Representation learning for treatment effect estimation from observational data. In *Advances in Neural Information Processing Systems 31*, pages 2633–2643. Curran Associates, Inc., 2018.

B. Zhang, A. A. Tsiatis, E. B. Laber, and M. Davidian. A robust method for estimating optimal treatment regimes. *Biometrics*, 68(4):1010–1018, 2012.

J. Zhang and E. Bareinboim. Non-parametric methods for partial identification of causal effects. *Columbia CausalAI Laboratory Technical Report*, 2021.

Q. Zhao, D. S. Small, and A. Ertefaie. Selective inference for effect modification via the lasso. *Journal of the Royal Statistical Society Series B*, 84(2):382–413, 2022.

Y. Zhao, D. Zeng, A. J. Rush, and M. R. Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.

Y.-Q. Zhao, D. Zeng, E. B. Laber, and M. R. Kosorok. New statistical learning methods for estimating optimal dynamic treatment regimes. *Journal of the American Statistical Association*, 110(510):583–598, 2015.

X. Zhou, N. Mayer-Hamblett, U. Khan, and M. R. Kosorok. Residual weighted learning for estimating individualized treatment rules. *Journal of the American Statistical Association*, 112(517):169–187, 2017.

C. Zigler, K. Watts, R. Yeh, Y. Wang, B. Coull, and F. Dominici. Model feedback in bayesian propensity score estimation. *Biometrics*, 69, 02 2013.

C. M. Zigler and F. Dominici. Uncertainty in propensity score estimation: Bayesian methods for variable selection and model-averaged causal effects. *Journal of the American Statistical Association*, 109(505):95–107, 2014.