TREATMENT EFFECT ESTIMATION AND THERAPEUTIC OPTIMIZATION USING OBSERVATIONAL DATA

Ruohong Li

Submitted to the faculty of the University Graduate School
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the Department of Biostatistics,
Indiana University

May 2021

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral Committee

Wanzhu Tu, Ph.D., CoChair

Honglang Wang, Ph.D., CoChair

March 11, 2021

Yi Zhao, Ph.D.

Kun Huang, Ph.D.

Mohammad Al Hasan, Ph.D.

DEDICATION

To my parents Mr. Wei Li and Mrs. Qing Li.

ACKNOWLEDGMENTS

I am sincerely thankful for my two co-advisors, Dr. Wanzhu Tu and Dr. Honglang Wang. I am grateful for their guidance, encouragement, and support of my Ph.D. research and life. They helped me when I had difficulties and encouraged me to explore new ideas. Without their advice, my Ph.D. research is unlikely to have gone so smoothly. In working with them, I have gained valuable research experience and learn to be a better scholar. I would also like to thank Dr. Yi Zhao, Dr. Kun Huang, and Dr. Mohammad Al Hasan for kindly serving on my research committee and for their many valuable comments and suggestions.

In addition to my dissertation committee members, I would like to give special thanks to Dr. Malaz Boustani and Dr. Christopher M. Callahan. They have supported me financially during my graduate study. They have given me numerous opportunities to practice and develop new statistical skills in real data analysis. Finally, I would like to express my gratitude to all Biostatistics program members, including faculty, staff, and fellow students, for the support and friendship.

Finally, I thank my loving parents and my two pet companions Pidan and Aoli.

Ruohong Li

TREATMENT EFFECT ESTIMATION AND THERAPEUTIC OPTIMIZATION USING OBSERVATIONAL DATA

In this dissertation, I address two essential questions of modern therapeutics: (1) to quantify the effects of pharmacological agents as functions of patient's clinical characteristics; (2) to optimize individual treatment regimen in the presence of multiple treatment options. To address the first question, I proposed a unified framework for the estimation of heterogeneous treatment effect $\tau(x)$, which is expressed as a function of the patient characteristics x. The proposed framework not only covers most of the existing advantage-learning methods in the literature, but also enhances the robustness of different learning methods against outliers by allowing the selection of appropriate loss functions. To cope with high-dimensionality in x, I incorporated into the method modern machine learning algorithms including random forests, gradient boosting machines, and neural networks, for a more scalable implementation. To facilitate the wider use of the developed methods, I developed an R package RCATE, which is now posted on Github for public access. For therapeutic optimization, I developed a treatment recommendation system using offline reinforcement learning. Offline reinforcement learning is a type of machine learning method that enables an agent to learn an optimal policy in the absence of an interactive environment, such as those encountered in the analysis of therapeutics data. The recommendation system optimizes long-term reward, while accounting for the safety of treatment regimens. I tested the method using data from the Systolic Blood Pressure Trial (SPRINT), which included multiple years of follow-up data from thousands of patients on many different antihypertensive drugs. Using

the SPRINT data, I developed a treatment recommendation system for antihypertensive therapies.

Wanzhu Tu, Ph.D., CoChair

Honglang Wang, Ph.D., CoChair

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Modern evidence-based medicine has been evolving in the last few decades from universal treatment rules to individualized treatment decisions tailored to patient characteristics. While treatment is personalized, the evidence supporting a specific therapy in a given individual still has to come from real observable clinical data. An important source of such data is electronic health records (EHR). We hope to distill relevant information about various treatments from EHR and achieve certain decision rules that would guide clinical decisions. Ultimately, these decisions have to be based on a *causal understanding* the treatment effects, derived from the real treatment outcome data, usually from observational settings.

In this dissertation, I address two practical challenges in treatment recommendation: (1) Quantifying treatment effects in individuals of given characteristics, also known as the heterogeneous treatment effect estimation, in situations of strong data irregularities. (2) Selecting treatments in a dynamic therapeutic setting, by optimizing treatment outcomes using observed data.

For the first task, one needs to effectively deal with data irregularities such as outliers to maintain the robustness of the estimator. Additionally, a practically usable estimator must be able to handle the high dimensionality of patient characteristics.

In the simplest situations involving only two treatments, decisions can be made by comparing patient outcomes under the two therapies. But since the patient outcomes were not observed under both treatments, we must consider the potential outcomes. Here, we express the *causal* effect τ of a drug for a given set of patient characteristics x. We then

compare the patient's responses Y when the treatment is applied $T = 1$ and when it is not applied $T = -1$. The causal effect of the drug is therefore expressed as $\tau(x) = E[Y^{(1)}|X = x] - E[Y^{(-1)}|X = x]$, where $Y^{(1)}$ and $Y^{(-1)}$ are potential outcomes. The quantity $\tau(x)$ is also known as the conditional average treatment effect (CATE). Because of the lack of simultaneously observation of outcomes under $T = 1$ and $T = -1$, we have to operate within a *counter-factual* framework, such as the one developed by Rubin and colleagues (Rubin, 1974; Rosenbaum and Rubin, 1983).

For the single-stage problem described above, I develop a broad class of methods for the estimation of $\tau(x)$ in the presence of data outliers and high dimensionality of x. The robustness of the methods is gained by the incorporation of an L_1-based loss function. In comparison with the L_2-based estimators, the L_1-based methods drastically reduce the influences of outlying values of the data set. The estimation framework that I put forward unites many existing estimators, including the popular efficient A-learning Robins (2004), R-learning (Nie and Wager, 2020), inverse propensity score weighting (Horvitz and Thompson, 1952; Hirano et al., 2003), augmented inverse propensity score weighting (Robins and Rotnitzky, 1995), modified covariates method (Tian et al., 2014; Chen et al., 2017), and modified covariates method with efficiency augmentation (Tian et al., 2014; Chen et al., 2017) through a general estimating equation, allowing both L_1 and L_2-loss functions to be employed. Theoretical properties of the proposed methods are studied to ensure the validity of statistical inference.

I also extend the methods for $\tau(x)$ estimation from model-based estimators to algorithm-based estimators. The incorporation of supervised learning algorithms in the estimation process further enhances the estimators' robustness against model mis-specification. The automation of the analytical process also makes the methods more scalable in practical

observational data analysis. Towards that end, I present a general-purpose **R** package **RCATE** (https://github.com/rhli-Hannah/RCATE) for the estimation of $\tau(x)$.

The above estimators and related computational algorithms are extensively tested. As a whole, they represent a new toolbox for causal estimation of CATE that can be used broadly in EHR data analysis. Real data applications are presented to illustrate the use of the proposed methods/algorithms as practical analytical tools.

For the second question of optimizing treatment in a dynamic setting, I propose a recommendation system for dynamic treatment regimens (DTR) based on patients' responses. I describe the development of the system within the context of hypertension treatment. Hypertension is a common condition characterized by sustained blood pressure elevation. The goal of antihypertensive treatment is to control blood pressure to the desirable range set by the clinical guidelines (cite JNC-8), to reduce the risk of hypertension-related complications and end-organ damage. Despite and perhaps because of the availability of a large number of antihypertensive agents, treatment regimens are often unnecessarily complicated, with less than optimal results.

To optimize the antihypertensive treatment, I propose a supervised batch-constrained dueling double deep Q-network (SBC-BDQ) to help clinicians make treatment decisions that maximize the long-term reward in blood pressure control. I choose this approach because the standard reinforcement is known to not work well in settings of offline learning – the data that I use to develop the system are static. Because of the large number of drugs, and the exponentially increased number of drug combinations, exploring the full treatment space is unrealistic. Finally, I want to make sure that the recommended treatments do not deviate too drastically from the therapy that the treating physician has recommended for safety and other practical reasons.

The dissertation is structured as follows. In Chapter 2, I present the unified formulation for robust estimation of CATE estimation and related theoretical properties of the estimators. In Chapter 3, I present a scalable solution for CATE estimation by combining the unified estimation formulation with supervised learning algorithms. In Chapter 4, I describe a new reinforcement learning algorithm for dynamic antihypertensive treatment recommendations. Literature reviews and methodological details concerning the proposed methods are presented within each chapter.

Chapter 2

Robust Estimation of Heterogeneous Treatment Effect using Electronic Health Record Data

2.1 Introduction

The ultimate goal of precision medicine is to optimize therapeutic outcomes by tailoring medical treatment and care provision according to individual patient characteristics. In practice, such tailoring must be guided by causal treatment effects expressed as functions of the observed patient characteristics $\mathbf{x}$ (Gabriel, 2012), which account for patient heterogeneity in a given clinical population. But in reality, the true treatment effect function $\tau_0(\mathbf{x})$ is almost never known and cannot be easily ascertained from clinical trials.

There is a sizable literature on the estimation of treatment effects in the form of $\tau_0(\mathbf{x})$. With covariates averaged out, $\tau_0(\mathbf{x})$ is reduced to the average treatment effect (ATE) $\tau_0 = \int \tau_0(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$, which can be estimated from clinical trials as well as observational studies (Imai et al., 2008). While randomized experiments provide by far the most straightforward estimation of τ_0, valid estimates can also be ascertained from observational data, by using the Neyman-Rubin causal model under appropriate assumptions (Sekhon, 2008). Estimating treatment effect in the presence of heterogeneity, however, is a much involved task. Popular approaches include the *advantage* or A-learning methods that directly model the contrasts among treatments (Murphy, 2003; Robins, 2004), and the *quality* or Q-learners that regress the outcomes on patient characteristics (Watkins, 1989; Watkins and Dayan, 1992). Under the general umbrella of A-learners, Tian et al. (2014); Chen et al. (2017) described a covariate-modification method. More recently, Nie and Wager (2020) proposed

a two-step learning algorithm that possesses a quasi-oracle property for estimating $\tau_0(\mathbf{x})$. Xiao et al. (2019) further improved the algorithm for enhanced robustness.

The performance of the above causal estimators is often influenced by the features of the observed data. An attractive and readily available data source for causal inference is EHR, digitalized medical records collected and maintained by health care organizations (Gunter and Terry, 2005). While statisticians have long recognized the values of EHR data in causal analysis (Stuart et al., 2013), they are also keenly aware of the challenges presented by such data, including data outliers and high dimensionality. The former could result in biased estimation and questionable inference, whereas the latter leads to a "curse of dimensionality" (Donoho et al., 2000).

In this research, we address the above issues in a broader context of heterogeneous treatment effect estimation. Specifically, we put forward a general estimation framework based on weighted score equations. The new formulation unifies many of the existing learners, while retaining the flexibility to accommodate different loss functions, permitting for example robust least absolute deviation (LAD) regression. The estimating formula enhances the capacity against outliers of modified-covariate method by Chen et al. (2017) and extends the ability to handle higher dimensionality of robust R-learner by Xiao et al. (2019), giving each an improvement. The approach's direct targeting of $\tau_0(\mathbf{x})$ relates it nicely to the concept of the A-learning methods. We performed extensive simulation studies to investigate the new methods' operational performance, in comparison with the existing ones. We also described a real data application to illustrate the use of the proposed methods.

2.2 Proposed Methods

2.2.1 Models and Assumptions

We consider the estimation of $\tau_0(\mathbf{x})$, the conditional average treatment effect, within the Neyman-Rubin potential outcome framework (Rubin, 1974). The binary treatment indicator T takes values 1 or -1, i.e., $T \in \{\pm 1\}$. We let $Y^{(1)}$ and $Y^{(-1)}$ be the *potential* outcomes under $T = 1$ and $T = -1$, respectively. We assume that data $\{(Y_i, T_i, \mathbf{X}_i)\}_{i=1}^n$ are independent and identically distributed (i.i.d.), where the pre-treatment covariates $\mathbf{X}_i$ could be high dimensional as in EHR analyses. We require the stable unit treatment value assumption (SUTVA) (Cox, 1958; Rosenthal and Jacobson, 1968) and write the *observed* outcome as $Y = I(T = 1)Y^{(1)} + I(T = -1)Y^{(-1)}$, where $I(\cdot)$ is an indicator function.

Within this framework, we focus on

$$\begin{aligned} \tau_0(\mathbf{x}) &= E[Y^{(1)} - Y^{(-1)}|\mathbf{X} = \mathbf{x}] = E[Y|\mathbf{X} = \mathbf{x}, T = 1] - E[Y|\mathbf{X} = \mathbf{x}, T = -1] \\ &= \mu_1(\mathbf{x}) - \mu_{-1}(\mathbf{x}), \end{aligned} \tag{2.1}$$

where the last part comes from the ignorability assumption defined below. This makes CATE estimation possible when $\mathbf{X}$ contains all confounders. When $T \in \{\pm 1\}$, we can always express the conditional mean outcome as

$$E(Y|\mathbf{X}, T) = b_0(\mathbf{X}) + \frac{T}{2}\tau_0(\mathbf{X}),$$

where $b_0(\mathbf{x}) = \frac{1}{2}(E[Y^{(1)}|\mathbf{X} = \mathbf{x}] + E[Y^{(-1)}|\mathbf{X} = \mathbf{x}])$. This leads to a general interaction model

$$Y_i = b_0(\mathbf{X}_i) + \frac{T_i}{2}\tau_0(\mathbf{X}_i) + \varepsilon_i, \tag{2.2}$$

where ε_i is subject to Assumption 3 below, along with the other assumptions stipulated by Rubin (1974); Rosenbaum and Rubin (1983).

In the existing literature, $\tau_0(\mathbf{x})$ is often depicted by a simple parametric model (Tian et al., 2014; Xiao et al., 2019). With $\mu(\mathbf{x}) = E[Y|\mathbf{X} = \mathbf{x}] = b_0(\mathbf{x}) + \frac{p(\mathbf{x})-1}{2}\tau_0(\mathbf{x})$, one has $Y_i - \mu(\mathbf{X}_i) = \frac{T_i - 2p(\mathbf{X}_i)+1}{2}\tau_0(\mathbf{X}_i) + \varepsilon_i$, which is exactly the Robinson decomposition used by the R-learner (Nie and Wager, 2020).

Assumption 1 (Ignorability) Treatment assignment T_i is independent of the potential outcomes $(Y_i^{(1)}, Y_i^{(-1)})$ given the covariates $\mathbf{X}_i$, i.e., $\{Y_i^{(1)}, Y_i^{(-1)} \perp\!\!\!\perp T_i|\mathbf{X}_i\}$.

Assumption 2 (Positivity) The propensity score $p(\mathbf{x}) := P(T = 1|\mathbf{X} = \mathbf{x}) \in (0, 1)$.

Assumption 3 (Conditional Independence Error) The error is independent of the treatment assignment conditional on covariates, i.e. $\{\varepsilon_i \perp\!\!\!\perp T_i|\mathbf{X}_i\}$. We further assume that the conditional expectation of error exists.

2.2.2 A Unified Formulation for Heterogeneous Treatment Effect Estimation

There are two general strategies for estimating $\tau_0(\mathbf{x})$ in (2.2). The first is to depict the conditional mean function $\mu_t(\mathbf{x}) = E[Y|\mathbf{X} = \mathbf{x}, T = t]$ with a regression model and then obtain the treatment effect estimator $\hat{\tau}(\mathbf{x}) = \hat{\mu}_1(\mathbf{x}) - \hat{\mu}_{-1}(\mathbf{x})$. For example, from the objective function $\sum_{i=1}^{n} \rho\left(Y_i - b(\mathbf{X}_i; \boldsymbol{\gamma}) - \frac{T_i}{2}\tau(\mathbf{X}_i; \boldsymbol{\beta})\right)$, one can estimate $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ simultaneously, and then achieves a CATE estimate $\hat{\tau}(\mathbf{x}) = \tau(\mathbf{x}; \hat{\boldsymbol{\beta}})$ (Chakraborty, 2013). Such an approach is often referred to as the Q-learning, because its objective function plays a role similar to that of the Q or reward function in reinforcement learning (Chakraborty, 2013). The frequently used Two- or Single-learners (T or S-learners for short) are variants of this approach (Künzel et al., 2019).

An alternative strategy, one that we follow in the current 75 research, is to directly target $\tau_0(\mathbf{x})$ in a predefined objective function. This approach is often referred to as the A-learning

(Schulte et al., 2014). A-learning first emerged in the context of dynamic treatment regime (Murphy, 2003; Robins, 2004), and was later generalized to one-stage case for treatment effect estimation (Tian et al., 2014; Chen et al., 2017). In this chapter, we show that there exists a unified formulation for the objective function, written in the form of score equations, that covers many of the existing learners.

Before introducing the general formulation, we first review the existing methods to highlight their connections.

1. *The modified outcome methods.* Certain transformations of Y could be used to facilitate the estimation of $\tau_0(\mathbf{x})$. Estimation methods relying on such transformations are collectively known as the modified outcome methods. This class of methods includes the *inverse propensity score weighting* (IPW) (Horvitz and Thompson, 1952; Hirano et al., 2003) and the *augmented IPW* (AIPW) methods (Robins and Rotnitzky, 1995). A common feature of this class of methods is to express the true treatment effect $\tau_0(\mathbf{x})$ as a conditional expectation of the *transformed* outcome variables. For IPW and AIPW, the transformations are

$$
\begin{aligned}
Y^{IPW} &= \frac{T - 2p(\mathbf{X}) + 1}{2p(\mathbf{X})(1 - p(\mathbf{X}))} \times Y; \\
Y^{AIPW} &= \frac{T - 2p(\mathbf{X}) + 1}{2p(\mathbf{X})(1 - p(\mathbf{X}))} \times [Y - (\mu_{-1}(\mathbf{X})p(\mathbf{X}) + \mu_1(\mathbf{X})(1 - p(\mathbf{X})))].
\end{aligned}
$$

Writing the modified outcome as Y^* one has $E(Y^*|\mathbf{X}) = \tau_0(\mathbf{X})$. An estimate can therefore be obtained by minimizing the square error loss, i.e., $\min_{\tau(\cdot)} \sum_{i=1}^{n} (Y_i^* - \tau(\mathbf{X}_i))^2$.

2. *The modified covariates methods (MCM).* An alternative set of methods, collectively known as the modified covariates methods, have been derived from the model (2.2). The central idea of this approach is to estimate $\tau_0(\mathbf{X})$ by re-weighting the loss function instead of the response variable (Tian et al., 2014; Chen et al., 2017)

$$
L(\tau(\cdot)) = \sum_{i=1}^{n} \left(D_i \frac{1}{p(\mathbf{X}_i)} + (1 - D_i) \frac{1}{1 - p(\mathbf{X}_i)} \right) \left(Y_i - \frac{T_i}{2} \tau(\mathbf{X}_i) \right)^2,
$$

where $D_i = (T_i+1)/2 \in \{0,1\}$. With appropriate weighting, the minimizer of the population version of the objective function equals to $\tau_0(\mathbf{x})$ as elaborated in Remark 1 below. Further, as shown in Appendix A.1, Y_i can be replaced by $Y_i - g(\mathbf{X}_i)$, where $g(\mathbf{X}_i)$ is an arbitrary function of $\mathbf{X}_i$. When $p(\mathbf{X}_i) = \frac{1}{2}$, the variance of the estimator is minimized when we replace Y_i with $Y_i - \mu(\mathbf{X}_i)$. This is known as the *modified covariates method with efficiency augmentation (MCM-EA)* (Tian et al., 2014).

3. *The R-learning method.* Nie and Wager (2020) recently proposed a method that they referred to as the R-learner (RL), named after Robinson's decomposition, a technique for estimating the parametric components in partially linear models (Robinson, 1988). The efficient A-learning introduced later in this section shared the same estimating equation of the R-learner, but the two were derived from different perspectives (Robins, 2004; Lu et al., 2013). Subtracting the marginal mean $E[Y_i|\mathbf{X}_i]$ from the outcome, Nie and Wager worked with the following equation

$$Y_i - E[Y_i|\mathbf{X}_i] = \left(\frac{T_i}{2} - p(\mathbf{X}_i) + \frac{1}{2}\right)\tau_0(\mathbf{X}_i) + \varepsilon_i,$$

where $E[\varepsilon_i|\mathbf{X}_i, T_i] = 0$. The treatment effect $\tau_0(\mathbf{x})$ can therefore be estimated by minimizing the following objective function,

$$L(\tau(\cdot)) = \sum_{i=1}^{n}\left(Y_i - \mu(\mathbf{X}_i) - \frac{T_i - 2p(\mathbf{X}_i) + 1}{2}\tau(\mathbf{X}_i)\right)^2,$$

where $\mu(\mathbf{X}_i)$ and $p(\mathbf{X}_i)$ are nuisance quantities estimated in advance.

Examining the relations between MCM-EA and R-learner, we note that in MCM-EA, since $E[Y_i - \mu(\mathbf{X}_i) - \frac{T_i}{2}\tau_0(\mathbf{X}_i)|\mathbf{X}_i] \neq 0$, one uses IPW as an adjustment so that $E[\frac{T_i}{2T_ip(\mathbf{X}_i)+(1-T_i)}\left(Y_i - \frac{T_i}{2}\tau_0(\mathbf{X}_i)\right)|\mathbf{X}_i] = 0$. In the R-learning, one has $E[Y_i - \mu(\mathbf{X}_i) -$

$(\frac{T_i}{2} - p(\mathbf{X}_i) + \frac{1}{2})\tau_0(\mathbf{X}_i)|\mathbf{X}_i] = 0$, so propensity score adjustment becomes unnecessary. This shows the difference and the connection between the R-learning and MCM-EA.

4. *The A-learning (AL) methods.* By directly targeting at the contrast function (treatment effect function), Robins (2004) derived the following equation for CATE estimation,

$$E\left[\left(Y_i - \frac{T_i+1}{2}\tau_0(\mathbf{X}_i)\right)(T_i - 2p(\mathbf{X}_i) + 1)\,\middle|\,\mathbf{X}_i\right] = 0,$$

where $\theta(\cdot)$ is an arbitrary function, or a more efficient version

$$E\left[\left(Y_i - \mu(\mathbf{X}_i) - \frac{T_i - 2p(\mathbf{X}_i)+1}{2}\tau_0(\mathbf{X}_i)\right)(T_i - 2p(\mathbf{X}_i) + 1)\,\middle|\,\mathbf{X}_i\right] = 0, \tag{2.3}$$

where the first term $Y_i - \mu(\mathbf{X}_i) - \frac{T_i-2p(\mathbf{X}_i)+1}{2}\tau_0(\mathbf{X}_i)$ has mean 0 conditional on $\mathbf{X}_i$. This corresponds exactly to Robinson's decomposition $Y_i - \mu(\mathbf{X}_i) = \frac{T_i-2p(\mathbf{X}_i)+1}{2}\tau_0(\mathbf{X}_i) + \epsilon_i$ when $E(\epsilon_i|\mathbf{X}_i) = 0$. Note that $Y_i - \mu(\mathbf{X}_i) - \frac{T_i-2p(\mathbf{X}_i)+1}{2}\tau_0(\mathbf{X}_i) = Y_i - \mu_{-1}(\mathbf{X}_i) - \frac{T_i+1}{2}\tau_0(\mathbf{X}_i)$ since $\mu(\mathbf{x}) = \mu_{-1}(\mathbf{x}) + p(\mathbf{x})\tau(\mathbf{x})$. The $\mu_{-1}(\mathbf{x})$ version was used by several authors (Schulte et al., 2014; Tsiatis, 2019).

This shows that Nie's R-learner shares the same conceptual essence with Robin's efficient A-learner, although the two were derived from different perspectives.

In summary, the methods reviewed above, including IPW, AIPW, MCM, MCM-EA, and RL could all be viewed as variants of AL, since they all target $\tau_0(\cdot)$ directly, with the pre-estimated plug-in nuisance quantities. We now show that these methods can be formulated under a unified presentation of the objective functions, at the level of score equations.

Noting that the above learners are all based on solutions to some score equations corresponding to the objective functions under the square error loss, we specify the score equations for these methods:

- Modified covariates:

$S_{MCM} = \frac{T}{2Tp(\mathbf{X})+(1-T)}\left(Y - \frac{T}{2}\tau_0(\mathbf{X})\right)$;

- Modified covariates with efficiency augmentation:

$S_{MCM-EA} = \frac{T}{2Tp(\mathbf{X})+(1-T)}\left(Y - \mu(\mathbf{X}) - \frac{T}{2}\tau_0(\mathbf{X})\right)$;

- R learning (efficient A learning):

$S_{RL} = \frac{T-2p(\mathbf{X})+1}{2}\left(Y - \mu(\mathbf{X}) - \frac{T-2p(\mathbf{X})+1}{2}\tau_0(\mathbf{X})\right)$;

- Inverse probability weighting:

$S_{IPW} = \frac{T-2p(\mathbf{X})+1}{2p(\mathbf{X})(1-p(\mathbf{X}))}\left(Y - \frac{2p(\mathbf{X})(1-p(\mathbf{X}))}{T-2p(\mathbf{X})+1}\tau_0(\mathbf{X})\right)$;

- AIPW:

$S_{AIPW} = \frac{T-2p(\mathbf{X})+1}{2p(\mathbf{X})(1-p(\mathbf{X}))} \times$
$\left(Y - ((1-p(\mathbf{X}))\mu_1(\mathbf{X}) + p(\mathbf{X})\mu_{-1}(\mathbf{X})) - \frac{2p(\mathbf{X})(1-p(\mathbf{X}))}{T-2p(\mathbf{X})+1}\tau_0(\mathbf{X})\right)$.

We note that all score equations listed above can be expressed in one general formulation

$$S = w(\mathbf{X}, T)c(\mathbf{X}, T)[Y - g(\mathbf{X}) - c(\mathbf{X}, T)\tau_0(\mathbf{X})], \tag{2.4}$$

where the two weight functions $w(\mathbf{x}, t)$ and $c(\mathbf{x}, t)$ are subject to the following constraints for all x and t:

C1. $p(\mathbf{x})w(\mathbf{x}, 1)c(\mathbf{x}, 1) + (1 - p(\mathbf{x}))w(\mathbf{x}, -1)c(\mathbf{x}, -1) = 0$;

C2. $c(\mathbf{x}, 1) - c(\mathbf{x}, -1) = 1$;

C3. $w(\mathbf{x}, t)c(\mathbf{x}, t) \neq 0$.

One can show that the existing estimation methods, including MCM, MCM-EA, RL, IPW, and AIPW, are all covered by this general formulation. In Appendix A.1, we show that for each of the above methods, the corresponding functions c and w meet the three conditions.

A few additional remarks are in order for this general expression:

Remark 1. Conditions C1-C3 are put in place to assure $E(S|\mathbf{X}) = 0$. It can be shown that under the square error loss function, the estimates derived from (2.4) are indeed minimizers

of the target function, i.e., $\tau_0(\mathbf{x}) = argmin_{\tau(\mathbf{x})} E[w(\mathbf{X}_i, T_i)(y - g(\mathbf{X}_i) - c(\mathbf{X}_i, T_i)\tau(\mathbf{x}))^2 | \mathbf{X}_i = \mathbf{x}]$. For detailed proof, see Property 1 in Appendix A.2. A similar results can be obtained under the absolute error loss function; see Property 3 in the same section of the appendix.

Remark 2. For given $w(\mathbf{x}, t)$ and $c(\mathbf{x}, t)$, one might be able to choose an appropriate $g(\mathbf{x})$ to achieve robustness to model mis-specification. For example, the $g(\mathbf{x}) = (1 - p(\mathbf{x}))\mu_1(\mathbf{x}) + p(\mathbf{x})\mu_{-1}(\mathbf{x})$ in the AIPW method with equation

$$E\left[\frac{T_i - 2p(\mathbf{X}_i) + 1}{2p(\mathbf{X}_i)(1 - p(\mathbf{X}_i))}\left(Y_i - g(\mathbf{X}_i) - \frac{2p(\mathbf{X_i})(1 - p(\mathbf{X}_i))}{T_i - 2p(\mathbf{X}_i) + 1}\tau_0(\mathbf{X}_i)\right)\middle|\mathbf{X}_i\right] = 0$$

leads to double robustness. Specifically, AIPW is robust against mis-specification of either propensity score model or both $\mu_{-1}(\mathbf{x})$ and $\mu_1(\mathbf{x})$.

Remark 3. When an additional condition $c(\mathbf{x}, 1) = 1 - p(\mathbf{x})$ holds and $g(\mathbf{x}) = \mu(\mathbf{x})$, the score equation in (2.4) leads to an estimator with the minimized variance. For an R learner, we have $c(\mathbf{x}, 1) = 1 - p(\mathbf{x})$, and the choice of $g(\mathbf{x}) = \mu(\mathbf{x})$ leads to the most efficient estimator. For MCM, this additional condition also holds when $p(\mathbf{x}) = \frac{1}{2}$, as in the case of randomized clinical trials.

Remark 4. With the unified formulation for the score functions, new estimators can be derived, for example, $E[(T_i - 2p(\mathbf{X}_i) + 1)(Y_i - g(\mathbf{X}_i) - \frac{T_i}{2}\tau_0(\mathbf{X}_i))|\mathbf{X}_i] = 0$, where $g(\mathbf{X})$ is an arbitrary augmented function of $\mathbf{X}$.

With the score function expressed as in (2.4), we propose an estimation procedure for CATE $\tau(\cdot)$,

$$\min_{\tau(\cdot)} \frac{1}{n}\sum_{i=1}^{n} w(\mathbf{X}_i, T_i)\rho(Y_i - g(\mathbf{X}_i) - c(\mathbf{X}_i, T_i)\tau(\mathbf{X}_i)) + \Lambda_n(\tau(\cdot)), \tag{2.5}$$

where $\rho(\cdot)$ is a user-specified loss function, and $\Lambda_n(\cdot)$ is a structural penalty function for $\tau(\cdot)$. This general procedure covers most of the existing methods for heterogeneous treatment effect estimation through a unified formulation.

2.2.3 Estimation Methods under the L_1 Loss

The estimation procedure described in (2.5) is general and flexible in the sense that it allows the analyst: (1) to choose different estimators through the specification of $w(\cdot)$ and $c(\cdot)$; (2) to select $g(\cdot)$ for efficiency enhancement; and (3) to specify a loss function $\rho(\cdot)$ that is most appropriate for the application. This general formulation provides a natural remedy to two practical issues in EHR data analysis: (1) lack of robustness of the L_2-based methods against outliers, (2) lack of accommodation of the high dimensionality of $\mathbf{X}$, and nonlinearity of $\tau(\mathbf{X})$.

Specifically, we put forward a class of robust estimators within the confines of the general estimating function (2.5). The method accommodates nonlinearity in $\tau(\cdot)$, further enhancing the modeling flexibility. Estimation is implemented under the usual causal inference Assumptions $1-3$.

Under the L_1-loss function, we show in Appendix A.2 that with Conditions C1-C3, we have

$$argmin_{\tau(\cdot)} E\left[w(\mathbf{X}_i, T_i)\cdot|Y_i - g(\mathbf{X}_i) - c(\mathbf{X}_i, T_i)\tau(\mathbf{X}_i)|\Big|\mathbf{X}_i = \mathbf{x}\right] = \tau_0(\mathbf{x}).$$

To increase efficiency, we opt to use $g(\mathbf{X}_i) = \mu(\mathbf{X}_i)$ in proposed methods.

Herein, we consider the following penalized least absolute deviation estimator

$$\min_{\tau(\cdot)} \frac{1}{n}\sum_{i=1}^{n} w(\mathbf{X}_i, T_i)|Y_i - \mu(\mathbf{X}_i) - c(\mathbf{X}_i, T_i)\tau(\mathbf{X}_i)| + \Lambda_n(\tau(\cdot)), \tag{2.6}$$

where Λ_n is added to ensure sparsity at the function level. For simultaneous variable selection and smooth estimation, we adopt a similar penalty term in (2.6) as described by Meier et al. (2009).

We further assume an additive structure for the treatment effect function $\tau(\cdot)$:

$$\tau(\mathbf{x}) = \alpha + m_1(x_1) + m_2(x_2) + ... + m_p(x_p),$$

where α is the intercept, and $m_j(\cdot)$ is the jth additive component corresponding to x_j. We write

$$m_j(x_j) = \sum_{k=1}^{K_n+q} B_{jk}(x_j)\beta_{jk},$$

where $\{B_{jk}(x_j)\}_{k=1}^{K_n+q}$ are the B-spline basis functions, K_n and q are number of knots and degree.

Rewriting the spline bases and coefficients as vectors, we have $\tau(\mathbf{x}) = \alpha + \boldsymbol{\beta}^T B(\mathbf{x})$, where $B(\mathbf{x}) = (B_1^T(x_1), \cdots, B_p^T(x_p))^T = (B_{11}(x_1), B_{12}(x_1), \cdots, B_{1(K+q)}(x_1), \cdots, B_{p(K+q)}(x_p))^T$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \cdots, \boldsymbol{\beta}_p^T)^T = (\beta_{11}, \beta_{12}, \cdots, \beta_{1(K+q)}, \cdots, \beta_{p(K+q)})^T$. For simplicity, we choose a common K_n+q for all spline components. Following a suggestion of $K_n \asymp \sqrt{n}+4$ by Meier et al. (2009), we use $K_n = \sqrt{n}/2$, which is of the same order and not too large for implementation.

With this, we define the penalty term in (2.6) as

$$\Lambda_n(\tau(\cdot)) = \sum_{j=1}^{p} P_{\lambda_1,\gamma}(J(m_j)), \text{ with } J(m_j) = \sqrt{||m_j||_n^2 + \lambda_2 I^2(m_j)}, \tag{2.7}$$

where $||m_j||_n^2 = \frac{1}{n}\sum_{i=1}^n m_j^2(\mathbf{X}_i) = \frac{1}{n}\boldsymbol{\beta}_j^T \mathbf{D}_j \boldsymbol{\beta}_j$ is for variable selection in a group-wise manner, and $I^2(m_j) = \int (m_j''(\mathbf{x}))^2 d\mathbf{x} = \boldsymbol{\beta}_j^T \boldsymbol{\Omega}_j \boldsymbol{\beta}_j$ is for smoothness of the nonzero components.

The integrals $\int B_{jl_1}(\mathbf{x}) B_{jl_2}(\mathbf{x}) d\mathbf{x}$ and $\int B''_{jl_1}(\mathbf{x}) B''_{jl_2}(\mathbf{x}) d\mathbf{x}$ are the (l_1, l_2)th entry of the $(K_n + q) \times (K_n + q)$ matrices $\mathbf{D}_j$ and $\boldsymbol{\Omega}_j$ respectively. And $P_{\lambda_1,\gamma}(\cdot)$ is the smoothly clipped

absolute deviation (SCAD) penalty defined by its first derivative

$$P'_{\lambda_1,\gamma}(x) = \lambda_1\{I(x \leq \lambda_1) + \frac{(\gamma\lambda_1 - x)_+}{(\gamma-1)\lambda_1} I(x > \lambda_1)\},$$

with $\gamma > 2$ and $P_{\lambda_1,\gamma}(0) = 0$. We use $\gamma = 3.7$ as suggested by Yuan and Lin (2006).

Hence, optimization of (2.6) can be expressed as a general group SCAD problem

$$\begin{aligned}(\hat{\alpha}, \hat{\beta}) =& argmin_{(\alpha,\beta)} \frac{1}{n} \sum_{i=1}^{n} w(\mathbf{X}_i, T_i) \left| Y_i - g(\mathbf{X}_i) - c(\mathbf{X}_i, T_i)(\alpha + \beta^T B(\mathbf{X}_i)) \right| \\ &+ \sum_{j=1}^{p} P_{\lambda_1,\gamma}(\sqrt{\boldsymbol{\beta}_j^T M_j(\lambda_2) \boldsymbol{\beta}_j}),\end{aligned}$$

where $\mathbf{M}_j(\lambda_2) = \frac{1}{n}\mathbf{D}_j + \lambda_2 \boldsymbol{\Omega}_j$. By decomposing $\mathbf{M}_j = \mathbf{R}_j^T \mathbf{R}_j$ for some invertible matrix $\mathbf{R}_j \in \mathbb{R}^{(K_n+q)\times(K_n+q)}$, we define

$$\tilde{\boldsymbol{\beta}}_j^T = \boldsymbol{\beta}_j^T \mathbf{R}_j \quad \text{and} \quad \tilde{B}_j(X_j) = \mathbf{R}_j^{-1} B_j(X_j). \tag{2.8}$$

With these transformations, the optimization of (2.6) becomes an ordinary least absolute deviation (LAD) regression with a group SCAD penalty

$$(\hat{\alpha}, \hat{\tilde{\boldsymbol{\beta}}}) = argmin_{(\alpha,\tilde{\boldsymbol{\beta}})} \frac{1}{n} \sum_{i=1}^{n} |Y_i^* - w_i^*(\mathbf{X}_i, T_i)(\alpha + \tilde{\boldsymbol{\beta}}^T \tilde{B}(\mathbf{X}_i))| + \sum_{j=1}^{p} P_{\lambda_1,\gamma}(||\tilde{\boldsymbol{\beta}}_j||), \tag{2.9}$$

where $||\tilde{\boldsymbol{\beta}}_j||$ is the Euclidean norm, $Y_i^* = w(\mathbf{X}_i, T_i)(Y_i - g(\mathbf{X}_i))$ and $w_i^*(\mathbf{X}_i, T_i) = w_i(\mathbf{X}_i, T_i) \times |c(\mathbf{X}_i, T_i)|$. The estimation of CATE is therefore $\hat{\tau}(\mathbf{x}) = \hat{\alpha} + \hat{\tilde{\boldsymbol{\beta}}}^T \tilde{B}(\mathbf{x})$.

2.2.4 A Computational Algorithm

To optimize (2.9), one has to estimate $\mu(\cdot)$ and $p(\cdot)$, as they are involved in the weight functions $w(x,t)$ and $c(x,t)$. Herein, we use pre-estimated $\hat{\mu}(\cdot)$ and $\hat{p}(\cdot)$ as plug-in estimates for solving (2.9). Estimation accuracy of these quantities, however, can be impeded by the

dimension of $\mathbf{x}_i$ and the uncertainty of the functional forms of the $\mathbf{x}_i$'s associations with T_i and Y_i. To remedy, we use a gradient boosting machine (GBM) to estimate these two functions (McCaffrey et al., 2004), with packages `gbm` (Ridgeway and Ridgeway, 2004) and `caret` (Kuhn, 2012). In cases of ultra-high dimensional $\mathbf{x}_i$, one could first use non-parametric independence screening (NIS) method (Fan et al., 2011) to reduce the dimensionality to a moderate one ($n-1$ or $log(n)$) as suggested by Fan and Lv (2008)), before applying our proposed method.

With the plug-in estimates of $\mu(\cdot)$ and $p(\cdot)$, we solve the L_1 optimization problem in (2.9), by using R package `rqPen` (Sherwood and Maidman, 2016), which is designed for penalized quantile regression in general. The nonconvex group penalized optimization with quantile loss is solved by the extension of quantile iterative coordinate descent (QICD) algorithm proposed by Peng and Wang (2015). For comparison purposes, we also use R package `oem` (Xiong et al., 2016) to ascertain the L_2 estimators.

The main steps of the procedure are described in Algorithm 1.

Algorithm 1: Robust model-based CATE estimating algorithm

1 **Input:** Outcome Y, treatment assignment T, and pre-treatment covariates $\mathbf{X}$

2 **Data screening:** Screen covariates with NIS when in situations of ultra-high dimension;

3 **Nuisance quantity estimation:** Estimate $p(\mathbf{x})$ by using GBM with cross-validation (CV) and estimate $\mu(\mathbf{x})$ by using L_1-based GBM with CV;

4 **Data transformation:** Construct the B-spline design matrix $B(\mathbf{X})$, calculate $w(\mathbf{X}_i, T_i)$, $c(\mathbf{X}_i, T_i)$, and $g(\mathbf{X}_i)$ following Conditions C1-C3, and transform $B(\mathbf{X})$ to $\tilde{B}(\mathbf{X})$ using (2.8);

5 **Optimization:** Solve penalized LAD regression (2.9) with a group SCAD penalty to achieve estimates of α and $\tilde{\boldsymbol{\beta}}$ with regularization parameters selected by CV.

6 **Output:** Calculate $\hat{\tau}(\mathbf{x}) = \hat{\alpha} + \hat{\tilde{\boldsymbol{\beta}}}^T \tilde{B}(\mathbf{x})$.

2.3 Asymptotic Properties of $\hat{\tau}(x)$

For theoretical examination, we consider the simple case of a univariate covariate $X_i \in \mathbb{R}$:

$$\min_{\tau(\cdot)} \frac{1}{n} \sum_{i=1}^{n} w(X_i, T_i) \rho(Y_i - g(X_i) - c(X_i, T_i)\tau(X_i)), \tag{2.10}$$

where $\rho(\cdot)$ is a loss function that is convex and has unique minimizer at origin. This simplification will not diminish the contribution of the asymptotic analysis, which is complicated by the B-spline approximation and the various loss functions including the L_1, L_2, Huber, and Bisquare loss functions.

With a B-spline approximation, we write $\tau(x) := \sum_{k=1}^{K_n+q} \beta_k B_k(x) = B(x)^T \boldsymbol{\beta}$, where q is the degree of the B-splines and K_n is the number of knots, which we assume depending on sample size n. Zhou et al. (1998) provided the L_∞ approximation error for B-splines. In particular, with $\tau^*(x) := B(x)^T \boldsymbol{\beta}^*$ as the best L_∞ approximation to the true function

$\tau_0(x)$, it satisfies

$$sup_{x\in(0,1)}|\tau^*(x) - \tau_0(x) - b^a(x)| = o(K_n^{-(q+1)}), \tag{2.11}$$

where

$$b^a(x) = -\frac{\tau_0^{(q+1)}(x)}{K_n^{(q+1)}(q+1)!}\sum_{k=1}^{K_n} I(\kappa_{k-1} \le x < \kappa_k) Br_{q+1}\left(\frac{x-\kappa_{k-1}}{K_n^{-1}}\right) = O(K_n^{-(q+1)}),$$

with $\{\kappa_k\}_{k=0}^{K_n}$ are the knots in the B-spline approximation, $\tau_0^{(q+1)}(x)$ is the $(q+1)$th order derivative of $\tau_0(x)$, and $Br_q(x)$ is the q-th Bernoulli polynomial.

We focus on the asymptotic theory of the L_1 spline estimator $\hat{\tau}(x) = B(x)^T\hat{\boldsymbol{\beta}}$, where

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{K_n+q}}{\arg\min}\, L_n(\boldsymbol{\beta}) := \sum_{i=1}^{n} w(X_i,T_i)\rho(Y_i - g(X_i) - c(X_i,T_i)B(X_i)^T\boldsymbol{\beta}). \tag{2.12}$$

The error for $\hat{\tau}(x)$ can be decomposed as a summation of the estimation error and approximation error

$$\hat{\tau}(x) - \tau_0(x) = \underbrace{\hat{\tau}(x) - \tau^*(x)}_{\text{estimation error}} + \underbrace{\tau^*(x) - \tau_0(x)}_{\text{approximation error}} = \hat{\tau}(x) - \tau^*(x) + b^a(x) + o(K_n^{-(q+1)}).$$

We only need to study the estimation error $\hat{\tau}(x) - \tau^*(x) = B(x)^T(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)$ thanks to the L_∞ approximation result by Zhou et al. (1998).

To show a pointwise asymptotic normality of $\sqrt{a_n}(\hat{\tau}(x) - \tau^*(x))$ with a convergence rate a_n to be specified later in Appendix A.3, we only need to prove the convergence of $\sqrt{a_n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)$ since $\hat{\tau}(x) - \tau^*(x) = B(x)^T(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)$. For this, denote $\boldsymbol{\delta} = \sqrt{\alpha_n}(\boldsymbol{\beta} - \boldsymbol{\beta}^*)$ and

$$U_n(\boldsymbol{\delta}) = \sum_{i=1}^{n}\left[w(X_i,T_i)\left(\rho\left(U_i - \frac{1}{\sqrt{\alpha_n}}c(X_i,T_i)B(X_i)^T\boldsymbol{\delta}\right) - \rho\left(U_i\right)\right)\right],$$

where $U_i = Y_i - g(X_i) - c(X_i, T_i)B(X_i)^T\boldsymbol{\beta}^*$. Then the minimizer $\hat{\boldsymbol{\delta}}_n$ of $U_n(\boldsymbol{\delta})$ is simply our target, i.e., $\hat{\boldsymbol{\delta}}_n = \sqrt{a_n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)$.

If one regards $\{U_n(\boldsymbol{\delta})\}$ as a sequence of random functions and the finite-dimensional distributions of $U_n(\boldsymbol{\delta})$ converge in distribution to those of some random function $U(\boldsymbol{\delta})$ which has a unique minimum, then it will follow that $\hat{\delta}_n = \sqrt{a_n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \to_d argmin(U(\boldsymbol{\delta}))$, as $n \to \infty$ per Hjort and Pollard (1993), and Geyer (1996).

With a given loss function $\rho(\cdot)$, we define $\Phi(s|X = x, T = t) = E[\rho(Y - g(x) - c(x,t)B(x)^T\boldsymbol{\beta}^* - s)|X = x, T = t]$. Let $\Phi'(s|X = x, T = t)$ and $\Phi''(s|X = x, T = t)$ be the first and second derivative of $\Phi(s|X = x, T = t)$ with for $\hat{\delta}_n$; respect to s. Several additional conditions are required for the proof of asymptotic normality:

C4. X is distributed as $Q(x)$ on a compact set in $\mathbb{R}$. Without loss of generality, we assume $X \in [0, 1]$.

C5. The B-spline knots are equidistantly located as $\kappa_k = k/K_n, k = 0, ..., K_n$ and the number of knots satisfies $K_n = O(n^{1/(2q+3)})$.

C6. The true CATE $\tau_0(x)$ is $(q + 1)$th order continuously differentiable.

C7. The function $\rho(u)$ is convex, it has a unique minimizer at zero, and its first and second derivatives exist.

C8. For $x \in [0, 1]$ and $t \in \{\pm 1\}$, $E[\rho'(Y - g(X) - c(X,T)\tau_0(X))^2|X = x, T = t] < \infty$.

C9. $\Phi(s|X = x, T = t)$, $\Phi'(s|X = x, T = t)$, and $\Phi''(s|X = x, T = t)$ are functions of s and they are bounded and continuous in a neighborhood of zero.

C10. As $s \to 0$, $E[\{w(X,T)\big(\rho(U - s) - \rho(U) - \rho'(U)s\big)\}^2] = o(s^2)$.

C11. There exists a $\gamma > 0$ such that for any $x \in [0, 1]$ and $t \in \{\pm 1\}$,

$$E[|w(X,T)c(X,T)\rho'(U)|^{2+\gamma}|X = x, T = t] < \infty.$$

Remark 5. The above conditions are needed for establishing an asymptotic normality of the estimator. Conditions C4-C6 are standard assumptions for B-spline regression. C5 provides the appropriate conditions of the knots. It suggests that the locations of the knots are set to some extent at regular intervals and the number of knots increases with the sample size. C4-6 are needed for controlling the spline approximation bias. C7-C8 are the general conditions for the loss function. The commonly used L_1, Huber, and Bisquare loss functions for robust regression all satisfy these conditions. C7 also guarantees the uniqueness of the estimator. C9 and C10 ensure the smoothness of the loss function ρ, which are needed for controlling the remainder term in the Taylor expansion. C11 is needed for satisfying the Lyapunov condition of the Central Limit Theorem.

To describe the asymptotic normality of the spline estimator $\hat{\tau}(x)$, we introduce two matrices: We define a square matrix $\mathbf{G} \in \mathbb{R}^{(K_n+q)\times(K_n+q)}$ with (i,j)-th element G_{ij}

$$\mathbf{G}_{ij} = \int_0^1 \frac{p(x)}{1-p(x)} w^2(x,1)c^2(x,1)\rho'(U_i)^2 B_i(x)B_j(x)dQ(x),$$

and another square matrix $\mathbf{D}$ of the same dimension with its (i,j)-th element being

$$\mathbf{D}_{ij} = \int_0^1 \nu(x)B_i(x)B_j(x)dQ(x),$$

where $\nu(x) = p(x)w(x,1)c(x,1)^2\rho''(y^{(1)} - g(x) - c(x,1)B(x)^T\boldsymbol{\beta}^*) + (1-p(x))w(x,-1)$ $c(x,-1)^2\rho''(y^{(-1)} - g(x) - c(x,-1)B(x)^T\boldsymbol{\beta}^*)$.

Theorem 1 *Assuming C1-C11, as* $n \to \infty$, *we have* $\sqrt{n/K_n}(\hat{\tau}(x) - \tau_0(x) - b^a(x)) \xrightarrow{D} N(0, \Psi(x))$, *where* $\Psi(x) = lim_{n\to\infty}\frac{1}{4K_n}B(x)^T\mathbf{D}^{-1}\mathbf{G}\mathbf{D}^{-1}B(x)$.

Remark 6. With the order of K_n larger than $O(n^{\frac{1}{2q+3}})$, the B-spline approximation error $b^a(x)$ can be ignored relative to the order of its variance.

For the rest of the chapter, we focus on the LAD loss where Conditions C7-C10 are naturally satisfied, and C11 can be simplified as the following:

C12. There exists a constant $\gamma \geq 0$ such that $E\left\{|w(X,T)c(X,T)|^{2+\gamma}|X=x\right\} < \infty$.

To describe the asymptotic normality of the spline estimator $\hat{\tau}(x)$ under the L_1 loss, we write matrix $\mathbf{D}$ with the (i,j)-th element being

$$\mathbf{D}_{ij} = \int_0^1 \Big[p(x)w(x,1)c^2(x,1)f_1(g(x)+c(x,1)\tau_0(x)|x) \\ + (1-p(x))w(x,-1)^2c(x,-1)f_{-1}(g(x)+c(x,-1)\tau_0(x)|x)\Big]B_i(x)B_j(x)dQ(x),$$

where $f_1(y|x)$ and $f_{-1}(y|x)$ are the conditional density functions of $Y^{(1)}$ and $Y^{(-1)}$ given $X=x$, respectively. We give the following theorem for the spline-based LAD regression:

Theorem 2 *With conditions C1-C6 and C12, as $n \to \infty$, we have $\sqrt{n/K_n}(\hat{\tau}(x) - \tau_0(x) - b^a(x)) \xrightarrow{D} N(0, \Psi(x))$, where $\Psi(x) = lim_{n\to\infty}\frac{1}{4K_n}B(x)^T\mathbf{D}^{-1}\mathbf{G}\mathbf{D}^{-1}B(x)$.*

Remark 7. For inference concerning $\tau_0(x)$, the variance of the estimator can be obtained by using resampling methods, as the asymptotic variance is difficult to work with. In a simulation experiment in Appendix A.4, we show that the bootstrap C.I. consistent with theoretical C.I..

2.4 A Simulation Study

We conducted an extensive simulation study to evaluate the finite sample performance of the proposed methods. We considered a large number of parameter settings, including four different learners under two different loss functions: (1) a robust version of the modified covariate method with efficiency augmentation (L_1-MCM-EA), (2) a robust R-learner (L_1-RL), (3) a robust A-learner (L_1-AL), (4) an L_2-based MCM-EA, (5) an L_2-based RL, (6) an L_2-based AL, and (7) a robust Q-learner (L_1-QL), and (8) an L_2-based Q-learner (L_2-QL).

The first six methods are under the umbrella of A-learning and they are covered by the general formulation in (2.5). The last two are Q-learning methods, which are not the focus of the current chapter; we included them only for comparison. The first three methods are what we recommend for situations with a significant number of outliers; Methods 4-6 are standard L_2-based learners.

We used the A-learner described by Lu et al. (2013). The objective functions of the A-learning methods 3 and 6 shared the same structure, except for the loss function ρ

$$L_n(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} \rho\Big(Y_i - \mathbf{X}_i^T\hat{\boldsymbol{\gamma}} - \Big[\frac{T_i+1}{2} - \hat{p}(\mathbf{X}_i)\Big] B(\mathbf{X}_i)^T\boldsymbol{\beta}\Big) + \Lambda_n(\boldsymbol{\beta}),$$

where $\boldsymbol{\gamma}$ and $p(\mathbf{x})$ are estimated in advance. We estimated $\boldsymbol{\gamma}$ by regressing Y on $\mathbf{X}$ using a linear regression, and $p(\mathbf{x})$ by regressing $\frac{T+1}{2}$ on $\mathbf{X}$ using GBM. The objective functions of the Q-learning methods 7 and 8 shared the same structure

$$L_n(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} \rho\left(Y_i - B(\mathbf{X}_i)^T\boldsymbol{\gamma} - \frac{T_i}{2}B(\mathbf{X}_i)^T\boldsymbol{\beta}\right) + \Lambda_n(\boldsymbol{\gamma}, \boldsymbol{\beta}),$$

where we used L_1 or L_2 loss function for ρ. Note that a difference between the A-learner and R-learner is the choice of the augmentation. For A-learner we used a linear function as suggested by Lu et al. (2013) to estimate $\mu(\mathbf{X}_i)$; we used L_1-based GBM to estimate $\mu(\mathbf{X}_i)$ in the R-learner.

We designed the simulation study to assess the robustness of the L_1 and L_2-based methods, and to contrast the performance of the A and Q-learners. We also examined the performance of the methods under different sample sizes, dimensionality, and proportions of outliers.

We assessed the performance of the methods using the standard metrics, including bias, variance, mean square error as well as mean absolute error. In addition, we compared the

value function $Q(\hat{\eta}) = E(Y(\hat{\eta}))$, i.e., the expected *average* outcome under treatment $\hat{\eta}$, where $\hat{\eta}(\mathbf{x}) = 2I(\hat{\tau}(\mathbf{x}) > 0) - 1$, as recommended by each method (Zhang et al., 2012). To estimate the $Q(\hat{\eta})$ for a given regimen, we conducted a Monte Carlo simulation using model $Y(\hat{\eta}) = b_0(\mathbf{X}) + \frac{\hat{\eta}}{2}\tau_0(\mathbf{X}) + \varepsilon$, replacing T in (2.2) by $\hat{\eta}$, and we set the number of replicates is 10^6. The value function calculated based on the true treatment effects was $E[Y(\eta^{opt})] = 1.25$, where $\eta^{opt}(\mathbf{x}) = 2I(\tau_0(\mathbf{x}) > 0) - 1$. We also assessed the sensitivity and specificity for variable selection under our penalty. With the number of simulation replication R, we defined

$$MAE_v = \frac{1}{R}\sum_{r=1}^{R}|\hat{\tau}^{(r)}(\mathbf{x}_v) - \tau_0(\mathbf{x}_v)|, \quad MSE_v = \frac{1}{R}\sum_{r=1}^{R}[\hat{\tau}^{(r)}(\mathbf{x}_v) - \tau_0(\mathbf{x}_v)]^2,$$
$$|Bias_v|^2 = |\frac{1}{R}\sum_{r=1}^{R}\hat{\tau}^{(r)}(\mathbf{x}_v) - \tau_0(\mathbf{x}_v)|^2, \quad Var_v = \frac{1}{R}\sum_{r=1}^{R}[\hat{\tau}^{(r)}(\mathbf{x}_v) - \overline{\hat{\tau}(\mathbf{x}_v)}]^2$$
$$Sensitivity = \frac{TP}{TP+FN}, \quad Specificity = \frac{TN}{TN+FP},$$

where $\mathbf{x}_v$ is the v-th observation from the validation set, $\hat{\tau}^{(r)}(\mathbf{x})$ is the estimator of $\tau(\mathbf{x})$ based on the r-th data replication, and $\overline{\hat{\tau}(\mathbf{x}_v)}$ is the average of all estimators of the v-th observation. TP, FN, TN, and FP represented the numbers of true positive, false negative, true negative, and false positive. In this research, the size of the validation set n_v was set to 200; we summarized the performance over the whole validation set by taking the averages (i.e.,$\overline{MSE} = \frac{1}{n_v}\sum_{v=1}^{n_v} MSE_v$). For simplicity, we reported MSE, MAE, $|Bias|^2$, and Var.

2.4.1 Data Generation

We generated data as follows, the dimension of the covariates was indexed by p:

$$\mathbf{X}_i \sim N_p(0, \mathbf{\Sigma}), diag(\mathbf{\Sigma}) = \mathbf{1}, Corr(X_{ij}, X_{ik}) = 0.5^{|j-k|}, i = 1, ..., n,$$

$$D_i|\mathbf{X}_i \sim Bernoulli(p(\mathbf{X}_i)), T_i = 2D_i - 1, logit(p(\mathbf{X}_i)) = X_{i1} - X_{i2},$$

$$Y_i = b_0(\mathbf{X}_i) + \frac{T_i}{2}\tau_0(\mathbf{X}_i) + \varepsilon_i, \varepsilon_i \sim (1 - \xi_o)N(0, 1) + \xi_o Laplace(0, 10),$$

$$b_0(\mathbf{X}_i) = 0.5 + 4X_{i1} + X_{i2} - 3X_{i3}, \tau_0(\mathbf{X}_i) = 2sin(2X_{i1}) - X_{i2} + 3tanh(0.5X_{i3}),$$

where η_o represented the proportion of outliers. We considered three settings: *(1) Various levels of outliers* $\xi_o \in \{0, 0.05, 0.1, 0.15, 0.2\}$, with $n = 1000$ and $p = 10$; *(2) Various training sample sizes* $n \in \{200, 500, 1000\}$, with $p = 10$ and $\xi_o \in \{0, 0.05\}$; *(3) Various dimension of training sample* $p \in \{10, 30, 50\}$, with $n = 1000$ and $\xi_o \in \{0, 0.05\}$.

2.4.2 Simulation Results

Figure 2.1 showed that when there were outliers, the L_1-based methods uniformly outperformed the L_2-based methods under the MSE, MAE, and $Q(\hat{\eta})$ value. Advantage of the robust methods increased with the proportion of outliers. The robust R-learner outperformed the robust A-learner because $\mu(x)$ was not a linear function. And there were little practical differences between the robust R-learner and robust MCM-EA. The Q-learner performed the best under MSE and MAE because it is a one-step estimation procedure, and thus avoiding the errors associated with the nuisance quantity estimation. This is consistent with the observations made by Schulte et al. (2014) that the Q-learner tended to perform better than the standard A-learner *when all models were correctly specified.* We conducted a separate simulation for a setting where the Q-function was mis-specified. The results reported in Appendix A.5 showed that in the presence of outliers, bias in the mis-specified L_1-QL was larger than that of the L_1-MCMEA, L_1-RL, and L_1-AL. The same was also true

for MSE and MAE. In terms of the value function $Q(\hat{\eta})$, L_1-QL had smaller $Q(\hat{\eta})$ values than methods under the A-learning umbrella; findings were consistent with MSE.

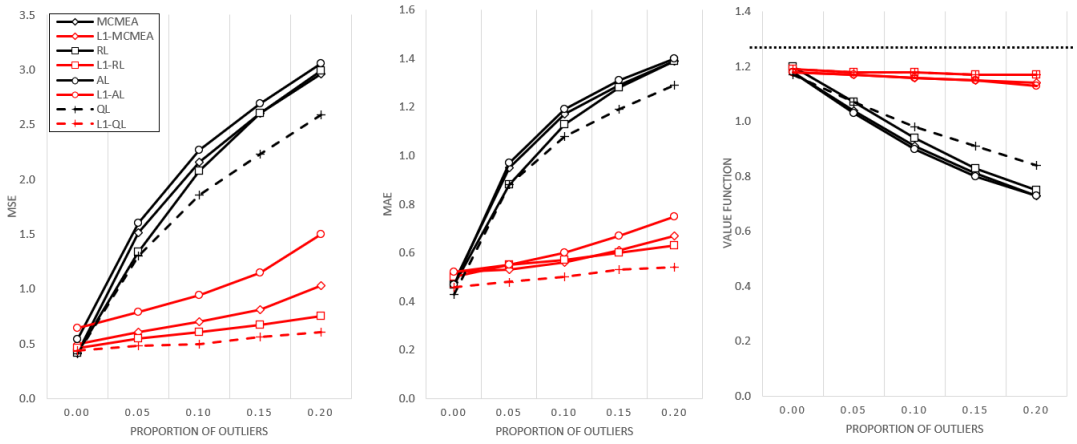


Figure 2.1: Simulation results of Setting 1.

Note: Comparison of mean squared error (MSE), mean absolute error (MAE), and value function ($Q(\hat{\eta})$) of the L_1-MCM-EA (red solid line), L_1-RL (red solid line), L_1-AL (red solid line), L_1-QL (red dashed line), MCM-EA (black solid line), RL (black solid line), AL (black solid line), and QL (black dashed line) under various levels of outliers. When there were outliers, both L_1-based methods outperformed the L_2-based methods. Advantage of the L_1-based methods increased with the proportion of outliers, under MSE, MAE, and $Q(\hat{\eta})$.

Figure 2.2 (A-D) showed the effects of sample size. Regardless of the presence or absence of outliers, as the sample size increased, MSE and MAE decreased for all methods. When there were no outliers, at a given sample size, the L_2-based methods tended to perform slightly better than the L_1-based methods, because the L_2-based methods were more efficient when the errors were normally distributed. But when there were even a small proportion of outliers, only 5% of errors generated from a different distribution, the robust methods outperformed L_2-based methods by a noticeable margin. Figure 2 (E-H) showed that the performance of proposed methods without NIS did not change substantially as the dimension of the covariates increased.

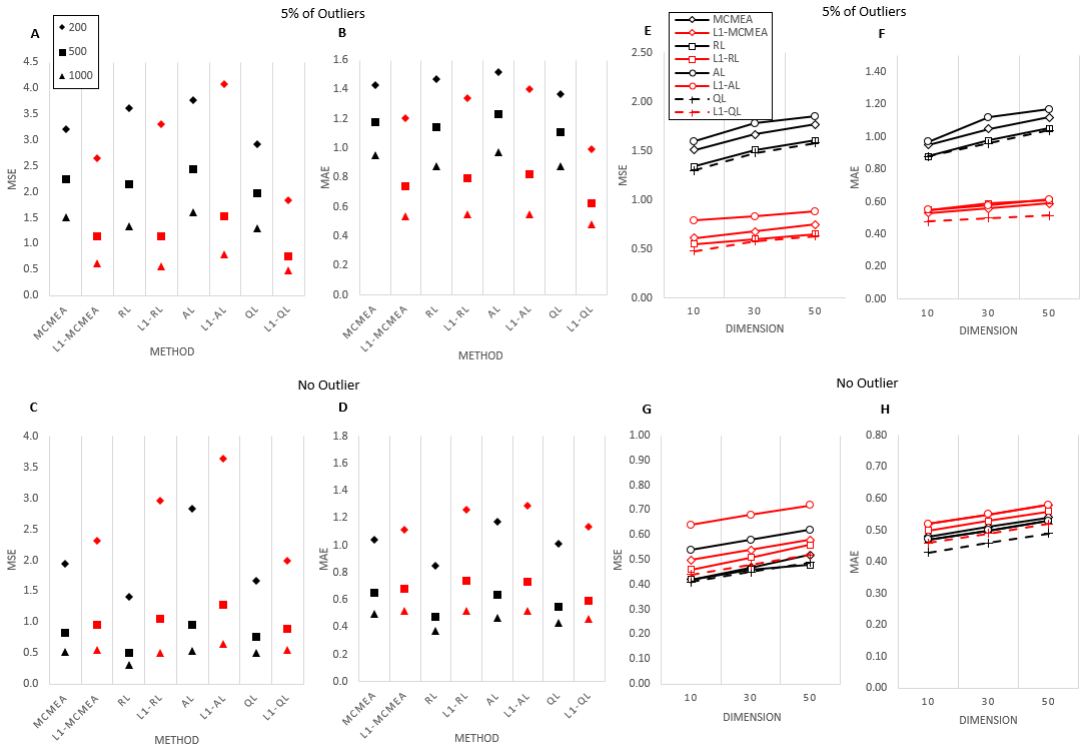


Figure 2.2: Simulation results of Setting 2

Note: Panels A-D – Mean squared error (MSE) and mean absolute error (MAE) values of different methods under different sample sizes, with and without outliers. The L_1-based methods are indicated by red symbols, whereas the L_2-based methods are indicated by black symbols. Panels E-H – Impact of the dimension on different methods. L_1-based methods (red lines) are robust to outliers, whereas the L_2-based methods (black lines) are standard methods.

Additional simulation details, including the squared bias, variance, MSE, MAE, sensitivity, specificity, and value function of the eight methods were reported in Appendix A.5. We have also examined the effects of dimension and smoothing on treatment effect estimation. Those results are included in Appendix A.5.

We conducted additional simulation in one covariate setting, where we calculated the pointwise bootstrap confidence intervals for $\tau(x)$, under both L_1 and L_2 versions of the MCM-EA and RL methods, with and without penalty. The L_1-based methods generally produced coverage probabilities very close to the nominal level, even with the presence of

outliers, whereas the L_2-based methods' coverage sometimes deviated strongly from 0.95. See Appendix A Table A.5.

2.5 Real Data Application

To illustrate the methods we propose, we estimated the treatment effects of two different antihypertensive therapies by analyzing the observed clinical data set from the Indiana Network of Patient Care, a local EHR system. The data were a subset of a previous study assessing the blood pressure (BP)-lowering effects of various antihypertensive agents (Tu et al., 2016). This analysis compared the BP effects of angiotensin-converting-enzyme inhibitors (ACEI) alone and a combination of ACEI and hydrochlorothiazide (HCTZ). We considered those on ACEI alone as in treatment group A, and those on ACEI+HCTZ as in group B. The primary outcome of interest is clinically recorded systolic BP in response to these therapies. Independent variables included the demographic and clinical characteristics, as well as medication-use behaviors of the study participants. Data from 882 participants were used in the current analysis. Among these, 350 were on the monotherapy of ACEI, and 532 were on the combination therapy of ACEI+HCTZ. Characteristics of the study participants are presented in Table 2.1. There were 4 continuous variables (pulse, BMI, age, and medication adherence) and 12 binary variables (gender, race, and ten comorbidities). The continuous variables were standardized before the analysis and expressed as linear combinations of splines.

We expressed the treatment effect of treatment B, in comparison against treatment A, as a function of the patient characteristics $\mathbf{x}$

$$\tau_0(\mathbf{x}) = E[Y^{(B)} - Y^{(A)} | \mathbf{X} = \mathbf{x}],$$

where $Y^{(A)}, Y^{(B)}$ represented the potential systolic BP of ACEI alone group and ACEI+ HCTZ group. Since the antihypertensive effect of a therapy is measured by its ability to lower BP, a negative $\tau(\mathbf{x})$ indicates a superior effect of the combination therapy over the monotherapy, for a given $\mathbf{x}$. An important covariate of interest was the level of medication adherence, which we measured with the proportions of days covered (PDC) by the medication.

Preliminary data examination showed that the observed systolic BP was right-skewed in both groups. The Shapiro–Wilk's test further confirmed that the systolic BP was not normally distributed, and there were outliers in the observed outcome (ACEI alone: $W = 0.9912$, $p = 0.035$; ACEI+HCTZ: $W = 0.9617$, $p = 1.498e - 10$). See Appendix A.6. We, therefore, used the L_1-based methods with additive B-splines to analyze the data. Here the B-splines were used to accommodate the possible nonlinear influences of the independent variables on the treatment effect.

Naive comparison of the systolic BP-effects between the two treatment strategies suggested that the combination therapy (ACEI+HCTZ) was significantly worse than the monotherapy (ACEI alone) in its ability to lower systolic BP (Table 1, 134.86mm Hg in ACEI vs. 137.49 mm Hg in ACEI+HCTZ; $p = 0.004$). A similar difference was seen in diastolic BP (80.98mm Hg in ACEI vs. 82.26 mm Hg in ACEI+HCTZ; $p = 0.046$). The observation is counterintuitive because there are no known mechanisms that would explain the attenuated BP benefit of ACEI when HCTZ is added to the treatment regimen. In fact, the current clinical guidelines recommend HCTZ as the first-line therapy for essential hypertension James et al. (2014). BP is regulated by hormones in the renin-angiotensin-aldosterone system (RAAS) (Tu et al., 2017). ACE inhibitors block the conversion of angiotensin I to angiotensin II, diminishing the latter's effects on aldosterone production and sodium retention and causing BP reduction. Thiazide diuretics lower BP by suppressing the extra-

cellular fluid volume, which in turn reduces aldosterone secretion. Together, the two drugs are expected to have additive effects in lowering BP. In clinical practice, the two are often used concurrently.

Table 2.1: Demographic and Clinical Characteristics of Study Subjects

Variable	ACEI (n=350)	ACEI+HCTZ (n=532)	p-value
	mean (sd)		
Average Systolic BP	134.86 (11.72)	137.49 (14.11)	0.004*
Average Diastolic BP	80.98 (8.64)	82.26 (9.77)	0.046*
Pulse	83.67 (10.36)	81.12 (10.51)	<0.001*
BMI	31.75 (8.65)	33.39 (8.79)	0.007*
Age	47.83 (12.84)	50.03 (12.43)	0.012*
Medication Adherence (PDC)	0.45 (0.30)	0.52 (0.27)	<0.001*
	n (percentage)		
Male	158 (45.1%)	189 (35.5%)	0.005*
Black	144 (41.1%)	290 (54.5%)	<0.001*
Diabetes	155 (44.3%)	114 (21.4%)	<0.001*
Chronic Kidney Disease (CKD)	8 (2.3%)	13 (2.4%)	1.000
Coronary Artery Disease (CAD)	10 (2.9%)	15 (2.8%)	1.000
Myocardial Infraction (MI)	2 (0.6%)	3 (0.6%)	1.000
Congestive Heart Failure (CHF)	7 (2.0%)	11 (2.1%)	1.000
Hyperlipidemia	53 (15.1%)	88 (16.5%)	0.645
Atrial fibrillation	1 (0.3%)	5 (0.9%)	0.461
Stroke	9 (2.6%)	6 (1.1%)	0.175
Chronic Obstructive Pulmonary Disease (COPD)	40 (11.4%)	51 (9.6%)	0.443
Depression	88 (25.1%)	132 (24.8%)	0.975

A closer examination of the characteristics of the patients on these therapies showed that patients on the combination therapy were older, more likely to be female, and overweight. Using GBM described in Section 2.3, we examined the mean function of systolic BP $\hat{\mu}(\mathbf{x})$ and the propensity of a patient receiving the combination therapy $\hat{p}(\mathbf{x})$. The estimated propensity score distributions were clearly different for the two treatment groups, whereas the mean functions were similar. See Appendix A.7. More specifically, the histogram of mean functions overlapped, indicating no apparent differences between the mean systolic BP between the two treatment groups. The different propensity score distributions of the two groups clearly showed that non-random treatment assignment. The importance levels of the covariates from GBM and additional modeling details were summarized in Appendix

A.7. The systematic differences in patient characteristics between the two treatment groups suggested that a naive comparison was not appropriate and should not be trusted.

We then analyzed the data with the proposed methods. Importantly, both the L_1-MCM-EA and L_1-RL selected BMI and PDC in the final models. The L_2-based methods, on the other hand, only selected PDC. As we have shown in the simulation study, in the presence of outliers, the rates of correct selection of patient characteristics in the proposed methods were substantially greater than that of the L_2-based methods. The estimated treatment effects as functions of BMI and PDC were depicted in Figure 2.3.

Figure 2.3 showed that $\hat{\tau}$ gradually decreased as the medication adherence measure PDC increased. Lower $\hat{\tau}$ indicated a stronger efficacy of the combination therapy than the monotherapy. Although decreasing trends were observed in both L_1 and L_2-based methods, the L_2 methods failed to detect any differences between the two therapies, as the 95% confidence intervals for $\hat{\tau}(PDC)$ consistently covered zero. The L_1-based estimators, however, showed a superior blood pressure-lowering effect of the combination therapy, but only *when PDC*$>$ 90%. The fact that treatment effects varied with medication adherence should not be surprising. As the former US Surgeon General, Dr. C. Everett Koop, wisely observed, "Drugs don't work in patients who don't take them." Osterberg and Blaschke (2005) In this analysis, we do not expect significant differences between the treatments *when patients are not adherent to the prescribed regimen.* Findings such as this are not unexpected in comparative effectiveness analysis of EHR data. Because unlike well-controlled clinical trials, few measures are in place to ensure patients faithfully take their medications in the real-world of clinical care. In the current application, the fact that the L_1-based estimators detected significant differences highlights the proposed methods' advantage. Using L_1-based estimators, we also examined the influences of BMI on τ, which did not reach the level of statistical significance (data not shown).

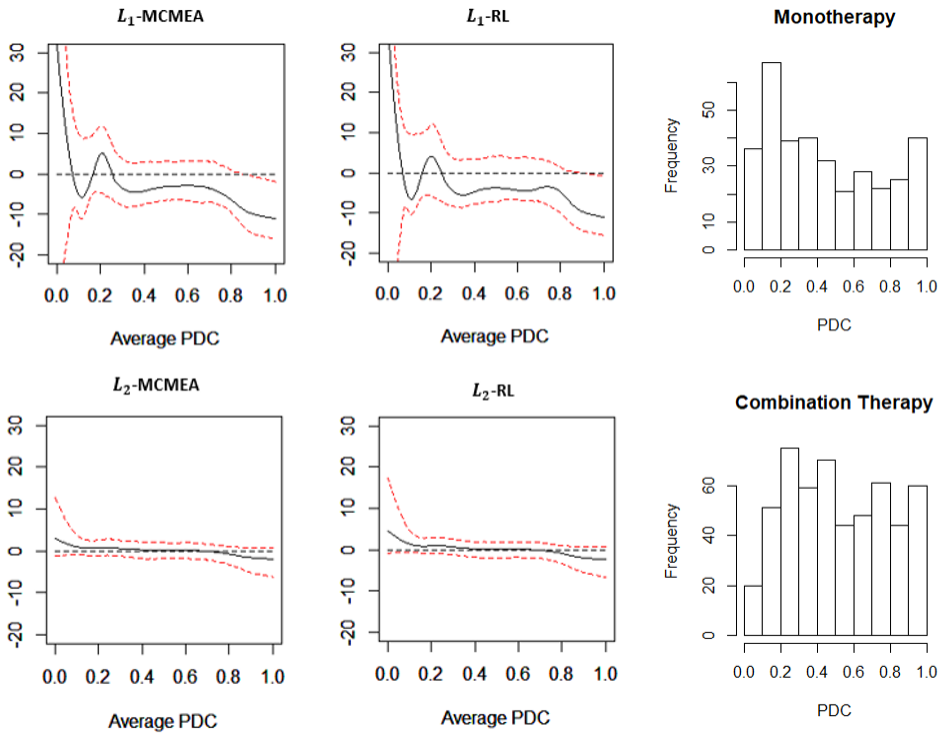


Figure 2.3: Marginal treatment effect of PDC

Note: Estimated treatment effects as functions of medication adherence (Proportion of Days Covered or PDC) under different methods. To plot these marginal effects, we fixed the continuous covariates at their mean values, and binary covariates at zero.

To check the conditional independence error assumption, we performed the invariant residual distribution test (IRD-test), invariant environment prediction test (IEP-test), invariant conditional quantile prediction test (ICQP-test), invariant targeted prediction test (ITP-test) Heinze-Deml et al. (2018), and invariant residual prediction test (IRP-test) (Shah and Bühlmann, 2018). The conditional independence error assumption held for both proposed methods at the significant level of 0.05.

Table 2.2: Conditional independence test results

Method	IRD-test	IEP-test	ICQP-test	ITP-test	IRP-test
L_1-RL	0.11	0.71	0.82	0.58	0.26
L_1-MCM-EA	0.25	0.71	1.00	0.57	0.20

Note: The values in the table are p-values. The conditional independence error assumption holds for both proposed methods at the significant level of 0.05.

In addition to the marginal treatment effect, we also examined the value function $\hat{Q}(\hat{\eta})$, which is the expected SBP under the estimated treatment regime $\hat{\eta}(\mathbf{x}) = 2I(\hat{\tau}(\mathbf{x}) < 0) - 1$. In the absence of a true value function, we used a 10-fold cross validation to estimate $\hat{Q}(\hat{\eta})$. For each fold F_j, we used the rest data for estimating $\hat{\mu}^{(-j)}(\mathbf{x})$, $\hat{p}^{(-j)}(\mathbf{x})$, and $\hat{\tau}^{(-j)}(\mathbf{x})$. Then we estimated the expected SBP by $\hat{Q}^{(j)}(\hat{\eta}) \triangleq \frac{1}{n_j}\sum_{i\in F_j}\hat{Y}^{(-j)}(\mathbf{x}_i)$ with $\hat{Y}^{(-j)}(\mathbf{x}_i) = \hat{\mu}^{(-j)}(\mathbf{x}_i) + [I(\hat{\tau}^{(-j)}(\mathbf{x}_i) < 0) - \hat{p}^{(-j)}(\mathbf{x}_i)]\hat{\tau}^{(-j)}(\mathbf{x}_i)$. By looping over $j = 1, 2, ..., 10$, we calculated $\hat{Q}(\hat{\eta}) = \frac{1}{10}\sum_{j=1}^{10}\hat{Q}^{(j)}(\hat{\eta})$. The observed average SBP was 136.45 mmHg, the estimates based on the L_1-MCMEA and L_1-RL were lower than the observed value. The estimates based on L_2-MCMEA and L_2-RL were slightly higher than the corresponding L_1-based methods. This results showed that the SBP could be reduced if treatment were to be assigned in accordance with the therapy recommended by the estimated treatment regime. Table 2.4 showed among the patients included in the analysis, 100 (11.3%) had PDC above 90%. We further examined the numbers of patients assigned to the two different treatment groups based on the estimated treatment effects. More patients would be assigned to the combination therapy group because it had a significantly greater blood pressure efficacy when patients take their medications. On the other hand, had we used the L_2 based methods, almost all of the patients would have been assigned to the monotherapy group, which contradicts the recommendations from the current clinical guidelines.

Table 2.3: Value functions of methods considered in application

Method	L_1-RL	L_2-RL	L_1-MCM-EA	L_2-MCM-EA
$\hat{Q}(\hat{\eta})$	134.98	135.13	133.96	134.64

Note: The value function is the expected systolic blood pressure under the estimated treatment regimen $\hat{\eta}(\mathbf{x}) = 2I(\hat{\tau}(\mathbf{x}) < 0) - 1$. Differences in value functions of the L_1 and L_2-based methods are minimal. However, the L_1-based methods outperform the L_2-based methods when data irregularities are present. See results from Table 2.4 and Figure 2.3.

Table 2.4: Treatment Assignment of Observations with $PDC > 0.9$

Method	Monotherapy (n)	Combination Therapy (n)
Real data	40	60
L_1-RL	4	96
L_1-MCM-EA	2	98
L_2-RL	100	0
L_2-MCM-EA	98	2

Note: Patients in the application data set whose $PDC > 0.9$ are reassigned treatments by estimated treatment effect, i.e. $\hat{\eta}(\mathbf{x}) = 2I(\hat{\tau}(\mathbf{x}) < 0) - 1$. Under the L_1-based methods, most of the patients will be assigned to the combination therapy group, consistent with the results in Figure 2.3. However, under the L_2-based methods, most of the patients will be assigned to the monotherapy group, which is counter-intuitive, because when patient adhere to prescription, the combination therapy is known to be more efficacious.

In summary, the naive and L_2-based methods showed that the combination therapy of ACEI and HCTZ had a worse BP-lowering effect than the monotherapy of ACEI, a finding that contradicts the recommendations of the current clinical guidelines of hypertension treatment. The L_1-based methods have produced results that are better explained by the existing clinical and biological evidence. The analysis showed that treatment effects tended to improve when patients adhere to their prescribed medications.

2.6 Discussion

We started this work searching for a robust estimator for heterogeneous treatment effects that could be used in EHR analysis, where outliers often undermine the validity of esti-

mation. In the process, we discovered a general formulation that not only addresses the issues of outliers but also covers a broad class of learners, including the commonly used A-learner, as well as other learning methods associated with it, such as the inverse propensity weighting, various modified outcome methods, modified covariate methods with or without efficiency augmentation, and the doubly robust method. Through a clever specification of the weight and efficiency augmentation functions, the formulation not only brings together a diverse set of methods under a unified presentation but also facilitates the development of a general-purpose implementational procedure. Although we have highlighted the use of the L_1 loss function for increased robustness against outliers in the EHR data, the score equation we described can readily accommodate other loss functions, giving the analyst much-enhanced flexibility in practical data analysis. As we have shown in our simulation studies, the use of L_1 loss function in heterogeneous treatment effect estimation substantially increases the estimation methods' robustness. Importantly, the gain in robustness does not appear to inflict a heavy toll on efficiency. Initial theoretical exploration suggests that reasonable asymptotic behavior can still be expected for the resultant estimators under various loss functions. Besides the flexibility in loss function selection, the general formulation also permits the incorporation of other useful features, such as nonparametric specifications of the mean and propensity functions and embedded dimensional reduction tools.

A theoretical examination of the proposed method shows that the resultant estimators possess the desirable property of asymptotic normally, under fairly general regularity conditions, and various commonly used loss functions. Simulation studies have provided strong and consistent empirical evidence on the utility of the proposed methods. Then through a real data application, we demonstrated how the proposed approach could be used in EHR data analysis to quantify treatment effects that varied with patient drug-taking behaviors.

The findings are in line with the existing clinical understanding of the therapeutic effects of the treatments. This said, the proposed method's performance remains to be tested in a wider range of clinical applications. Notwithstanding this limitation, we have taken the first steps in developing a scalable solution to estimate heterogeneous treatment effects in settings that are more prone to various forms of data irregularities.

Chapter 3

Algorithm-based Robust Estimation of Heterogeneous Treatment Effects

3.1 Introduction

In causal analyses of observational data, analysts face practical challenges in both methodology and implementation: (1) While there is a large literature on estimating treatment effects in observational studies, few methods are designed to deal with data irregularities and high dimensionality. Failure to accommodate these data characteristics tends to undermine the validity of causal estimation and inference. (2) Model-based causal inference methods are also vulnerable to model mis-specification, which could lead to erroneous conclusions. (3) For implementation, few software packages are available for use in an off-the-shelf fashion. Lack of ready-made analytical tools hinders practical use of innovative methods because practitioners are rarely in a position to implement and test complicated causal inference methods for practical data analyses.

In this research, we address the above challenges by putting forward a new set of tools to aid practical causal analysis of observational data. The tool kit includes a new class of estimators that we have recently developed for heterogeneous treatment effects (see Chapter 2). The methods are robust against data irregularities such as outliers, and they are also designed to handle high dimensional data, such as those encountered in electronic health records. We further extend the model-based estimators to algorithm-based methods to reduce the risk of model mis-specification. The supervised learning algorithms used in the estimating process enhance the methods' general usability and free analysts from the tedious model-fitting process. Finally, we implement these causal inferences tools in the form of

an **R** package - **RCATE**, which stands for Robust Estimation of the Conditional Average Treatment Effects.

For narrative convenience, we describe the development of estimation methods and related analysis tools in the context of treatment effects comparison of two antihypertensive therapies, by using the same real electronic health record data as Section 2.5. The chapter is structured as follows: In Section 3.2, we introduce the notation and assumptions for the estimation procedure. In Section 3.3, we present a simulation study to verify the performance of the proposed methods. In Section 3.4, we revisit the antihypertensive study and present the analytical results. Finally, we summarize our findings in Section 3.5. Details of the **R** package **RCATE** are provided in Appendix B.1.

3.2 Methods

3.2.1 The existing methods

There is a sizable literature on the estimation of CATE using observational data. Caron et al. (2020) and Zhang et al. (2020) provided reviews of state-of-the-art methods for CATE estimation. We summarize the key features of the existing methods in Table 3.1, which also provides the availability of analytical software. Importantly, most of these methods are based on the L_2-loss function, whose performance tends to deteriorate with data irregularity.

More recently, our research team developed a general formulation that has unified different learners (see Chapter 2). The formulation also accommodates other loss functions including L_1-loss, Huber loss, and Bi-square loss that enhance the estimators' robustness against data irregularities. In the next section, we briefly review this general formulation, and its coverage of the existing methods.

Table 3.1: Summary of existing popular CATE estimation algorithms

Base-learner/ Algorithm	Description	Pros(+) and Cons(-)	Available **R** packages
S-learner	Fits a single-model for the outcome with the covariates and treatment assignment indicator.	(+) If the treatment effect is simple, then pooling the data together will be beneficial. (–) Performs bad if the treatment effect is strongly heterogeneous and the response surfaces of two groups are very different.	**rlearner** **causalToolbox**
T-learner	Fits two models for the outcome of two treatment groups separately with the covariates.	(+) Performs well if the treatment effect is strongly heterogeneous and the response surfaces of two groups are very different. (–) Uses the data inefficiently.	**rlearner** **causalToolbox**
X-learner (Künzel et al., 2019)	A three step approach to crossover the information in the control and treated subjects.	(+) Has the advantages of both S and T-learner. (–) The three-step estimator increases the risk of over-fitting and the difficulty in tuning parameter.	**rlearner** **causalToolbox**
IPW	Transforms the outcome by inverse propensity score weighting, then the conditional expectation of the transformed outcome is the treatment effect.	(+) After transformation, the IPW provides the flexibility in choosing off-the-shelf supervised learning algorithms. (–) Relies on the accurate estimation of the propensity score.	
AIPW	Augmented IPW is robust to mis-specified mean or propensity score model.	(+) In addition to the advantage of IPW, AIPW has the property of double robustness.	**RCATE**
RL	Decomposes the outcome by subtracting the mean model and gets an estimating equation.	(+) In addition to the advantage of IPW, R-learner has quasi-oracle property.	**rlearner** **RCATE**
MCM-EA	Transforms the covariates to get an estimating equation.	(+) Same as IPW. (–) Relies on the accurate estimation of mean and propensity score.	**RCATE**
Q-learner	Fits the interaction model and the slope is the treatment effect function.	(+) No nuisance parameter need to be estimated. (–) Lacks of flexibility in algorithm choosing and sensitive to model mis-specification.	
Causal tree (Athey and Imbens, 2016)	Uses regression tree that splits by maximizing the difference between treatment effects in child nodes to fit the outcome.	(+) Easy to interpret and provides the grouping of subjects. (–) Suffers from the problem of high variance.	**causalTree**
Causal forest (Athey et al., 2019)	Uses randomly selected subsample and covariates to build causal trees, then aggregate the results.	(+) Addresses the high variance problem. (–) Lose the interpretability.	**grf**
Causal boosting (Powers et al., 2018)	An adaption of gradient boosting algorithm with causal trees as weak-learner.	(+) Well-tuned causal boosting outperforms the causal forest. (–) Takes longer to train than causal forest and could overfit the training data.	**causalLearning**
Causal MARS (Powers et al., 2018)	Fits two multivariate adaptive regression spline models in parallel in two arms of the data. In each step, it chooses the same basis function to add to each model.	(+) Alleviates the bias problem of tree-based algorithms because they use the average treatment effect within each leaf as the prediction for that leaf.	**causalLearning**

3.2.2 A unified formulation for heterogeneous treatment effect estimation

As studied in Chapter 2, we proposed a unified estimation formulation for CATE, $\tau_0(\cdot)$,

$$\min_{\tau(\cdot)} \frac{1}{n} \sum_{i=1}^{n} w(\mathbf{X}_i, T_i) M\{Y_i - g(\mathbf{X}_i) - c(\mathbf{X}_i, T_i)\tau(\mathbf{X}_i)\}, \tag{3.1}$$

where $M(\cdot)$ is a user-specified loss function, and the two weight functions $w(\mathbf{x}, t)$ and $c(\mathbf{x}, t)$ are subject to three constraints C1-C3. In Table 3.2, we list the functions c, w, and g that can be chosen to meet the constraints for popular A-learning methods.

Table 3.2: Parameters of some popular methods in the framework

Method	$w(\mathbf{X}_i, T_i)$	$g(\mathbf{X}_i)$	$c(\mathbf{X}_i, T_i)$
MCM	$\{T_i p(\mathbf{X}_i) + (1 - T_i)/2\}^{-1}$	0	$\frac{T_i}{2}$
MCM-EA	$\{T_i p(\mathbf{X}_i) + (1 - T_i)/2\}^{-1}$	$\mu(\mathbf{X}_i)$	$\frac{T_i}{2}$
RL	1	$\mu(\mathbf{X}_i)$	$\{T_i - 2p(\mathbf{X}_i) + 1\}/2$
IPW	$\left\{\frac{T_i - 2p(\mathbf{X}_i)+1}{2p(\mathbf{X}_i)(1-p(\mathbf{X}_i))}\right\}^2$	0	$\frac{2p(\mathbf{X}_i)(1-p(\mathbf{X}_i))}{T_i - 2p(\mathbf{X}_i)+1}$
AIPW	$\left\{\frac{T_i - 2p(\mathbf{X}_i)+1}{2p(\mathbf{X}_i)(1-p(\mathbf{X}_i))}\right\}^2$	$(1 - p(\mathbf{X}_i))\mu_1(\mathbf{X}_i) + p(\mathbf{X}_i)\mu_{-1}(\mathbf{X}_i)$	$\frac{2p(\mathbf{X}_i)(1-p(\mathbf{X}_i))}{T_i - 2p(\mathbf{X}_i)+1}$

In practice, the existence of outliers is common and L_1-loss based methods can naturally alleviate the impact of data irregularities and lead to a robust estimation. With an L_1-loss function, under Conditions C1-C3,

$$\tau_0(\cdot) = \arg\min_{\tau(\cdot)} E\big[w(\mathbf{X}_i, T_i) \cdot \|Y_i - g(\mathbf{X}_i) - c(\mathbf{X}_i, T_i)\tau(\mathbf{X}_i)\|_1 \big| \mathbf{X}_i\big]. \tag{3.2}$$

3.2.3 Supervised learning algorithms for robust CATE Estimation

Through a transformation, CATE estimation in (2.6) becomes a problem of ordinary least absolute deviation (LAD) optimization,

$$\hat{\tau} = argmin_{\tau \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} w_i^*(\mathbf{X}_i, T_i) |Y_i^* - \tau(\mathbf{X}_i))|, \tag{3.3}$$

where $Y_i^* = \frac{Y_i - g(\mathbf{X}_i)}{c(\mathbf{X}_i, T_i)}$ and $w_i^*(\mathbf{X}_i, T_i) = w_i(\mathbf{X}_i, T_i)|c(\mathbf{X}_i, T_i)|$. $\mathcal{F}$ depends on the algorithm one uses, the algorithms can also take care of variable selection in high dimensional situations. In Section 3.3, we compare the L_1 and L_2-based algorithms. For the L_2-based methods, the transformed weight is $w_i^*(\mathbf{X}_i, T_i) = w_i(\mathbf{X}_i, T_i)c(\mathbf{X}_i, T_i)^2$.

With the objective function in (3.3), different supervised learning algorithms can be used to estimate CATE - the optimization becomes a weighted supervised learning problem, where Y_i^* and w_i^* are the new outcome and new weight of each sample. The nuisance quantities in Y_i^* and w_i^* are pre-estimated and plugged in. Similarly, any supervised learning algorithm with weighted L_1 loss can be used with (3.3) to achieve robust CATE estimation. In this section, we describe three different algorithms for this purpose. The algorithms are based on Random Forests (RF), gradient boosting machine (GBM), and artificial neural network (ANN). The common process underlying these algorithms is depicted in Figure 3.1. The detailed algorithms are introduced in following subsections.

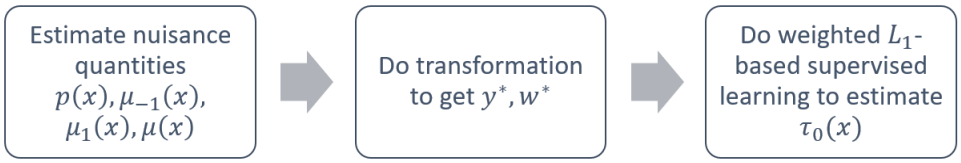


Figure 3.1: The estimating process of proposed algorithms.

Robust Random Forest Learner

We first used RF for robust treatment effect estimation. The building blocks of random forests are regression trees (Breiman et al., 1984), which recursively partition the sample by covariates to minimize the heterogeneity in the outcomes. The partition that minimizes the heterogeneity in child nodes is chosen, so that variables reducing heterogeneity most have greater chances of being selected than the background noise variables (Biau, 2012). Binary splits lead to trees, and then aggregated results within the terminal nodes are used for prediction. The random forests by create a more stable tree structure and reduce variance by combining a large number of de-correlated regression trees Breiman (2001).

Standard regression trees, by extension RF, minimize the MSE of the observed outcomes (i.e., $MSE = \sum_{i \in L_l}(y_i - \bar{y_l})^2 + \sum_{i \in L_r}(y_i - \bar{y_r})^2$, where $\bar{y_l}$ and $\bar{y_r}$ are the average values within left and right child nodes) (Hastie et al., 2009). But robust random forests for regression have been studied Roy and Larocque (2012). Several modifications based on standard RF can be made to gain robustness including LAD-based splitting rule (Breiman et al., 1984), aggregating the predictions over the trees using the median values. Empirical studies have demonstrated that these modifications offer more protection against outliers than the standard RF in most cases.

The robust RF-based CATE estimator we propose follows a similar structure. However, robust RF splits the samples based on weighted LAD (or WLAD) rule, a variant of the LAD rule in Roy and Larocque (2012). The best split at a node is the one that minimizes

$$WLAD = \sum_{i \in L_l} w_i^* |y_i^* - \tilde{y_l^*}'| + \sum_{i \in L_r} w_i^* |y_i^* - \tilde{y_r^*}'|, \tag{3.4}$$

where $\tilde{y_l^*}'$ and $\tilde{y_r^*}'$ are leaf node medians to increase robustness and w_i^* is the weight of each observation. For prediction, we used the mean of the medians that consist of the WLAD

splitting rule as the final prediction as in Meinshausen and Ridgeway (2006) instead of the median of mean in Roy and Larocque (2012).

Algorithm 2: Robust RF-based CATE estimating algorithm

Input: Data $\{(Y_i, T_i, \mathbf{X}_i)\}_{i=1}^n$, number of trees T, fraction of features used in splitting $p_{fraction}$, minimum node size k, boostrap sample size N.

Estimate nuisance quantities $p(x), \mu(x), \mu^{(1)}(x), \mu^{(-1)}(x)$ using (robust) GBM;

Calculate w_i^* and y_i^* according to Table 3.2 and Formulation (3.3);

for *t in 1,...,T* **do**

    a. Randomly select N observations with replacement from the dataset as the bootstrap sample and randomly select a subset of variables with size $p_{fraction} \times p$, where $p_{fraction} \in (0, 1)$;

    b. Fit a regression tree by repeating following steps until we reach the minimum node size k:

        b.1 Find the variable and the cutoff value that best split the data into two child nodes based on (3.4);

        b.2 Split the current node into two child nodes;

    c. Calculate the median of the transformed outcomes in each terminal node as CATE estimator;

end

Output: Splitting criterion and CATE estimators of terminal nodes.

The important tuning parameters T, $p_{fraction}$, k, and N can be selected by cross validation.

Robust Gradient Boosting Machine Learner

Gradient boosting machine (GBM) is a supervised learning technique that produces a prediction model $\hat{f}(x)$ in the form of sequential weak-learners, typically regression trees, so

that it performs better in high-dimensional settings (Friedman et al., 2000; Friedman, 2001, 2002). GBM builds the model in a stage-wise fashion by allowing optimization of a differentiable loss function $\Psi(y, f)$. The principle idea behind this algorithm is to construct weak-learners that are maximally correlated with the negative gradient of the loss function, associated with the whole ensemble.

Friedman's GBM algorithm initializes $\hat{f}(x)$ to be a constant. Then, in each iteration, it computes the negative gradient as the working response

$$z_i = -\frac{\partial}{\partial f(x_i)}\Psi(y_i, f(x_i))\Big|_{f(x_i)=\hat{f}(x_i).}$$

A regression model $g(x)$ is fitted to predict z from the covariates x. Finally, it updates the estimate of $f(x)$ as $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda g(x)$, where λ is the step size. Friedman (2001) also proposed LAD-TreeBoost algorithm, a variation of GBM, which is highly robust against outliers. Ridgeway (2007) later extended the LAD-TreeBoost algorithm to a weighted version.

In the proposed robust GBM for CATE estimation, we further extended Ridgeway's algorithm by combining it with the unified CATE estimation formulation as follows:

Algorithm 3: Robust GBM-based CATE estimating algorithm

Input: Data $\{(Y_i, T_i, \mathbf{X}_i)\}_{i=1}^n$, number of trees T, fraction of observations used in splitting p_{sample}, interaction depth c, and step size λ.

Estimate nuisance quantities $p(x), \mu(x), \mu^{(1)}(x), \mu^{(-1)}(x)$ using (robust) GBM;

Calculate w_i^* and y_i^* according to Table 3.2 and Formulation (3.3);

Initialize $\hat{f}(x)$ to be a constant, $\hat{f}(x) = median_{w^*}(y^*)$;

for *t in 1,...,T* **do**

  a. Compute the negative gradient as the working response $z_i = -sign(y_i^* - f(x_i))\big|_{f(x_i)=\hat{f}(x_i)}$;

  b. Randomly select $p_{sample} \times n$ observations without replacement from the dataset, where $p_{sample} \in (0, 1)$;

  c. Fit a regression tree to predict z_i using covariates x_i with interaction depth c and the number of leaf nodes K;

  d. Compute the optimal predictions of terminal nodes, $\rho_1, ..., \rho_K$, as $\rho_k = argmin_\rho \sum_{x_i \in S_k} \Psi(y_i^*, \hat{f}(x_i) + \rho, w_i^*)$, where $\Psi(y, x, w) = w|y - x|$ and k indicates the index of the terminal node S_k into which an observation with feature x would fall;

  e. Update $\hat{f}(x)$ as $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda\rho_k(x)$, where λ is step size.

end

Output: Splitting criterion and CATE estimates in terminal nodes.

For robust estimation, the terminal node estimate is the weighted median $median_{w^*}(z)$, defined as the solution to the equation $\frac{\sum w_i^* I(y_i^* \le \rho_k)}{\sum w_i^*} = \frac{1}{2}$. The important tuning parameters T, λ, c, and K can be selected by cross validation.

Robust Artificial Neural Network Learner

Artificial neural network (ANN) is a biologically inspired computer program designed to simulate the way in which the human brain processes information (Goodfellow et al., 2016). A no-hidden-layer ANN with identity activation function is similar to linear regression in its model structure. But ANN with multiple hidden layers offer great flexibility for use in real applications. A feed-forward neural network with two hidden layers can be represented as $g(x) := f^3(W^3 f^2(W^2 f^1(W^1 x)))$, where $W^l = (w^l_{jk})$ are the weights between layer $l-1$ and l, where w^l_{jk} is the weight between the k-th node in layer $l-1$ and the j-th node in layer l, and f^l is the activation function at layer l. Multi-layer networks use a variety of learning techniques to learn the weight factors, the most popular one is backpropagation (Rumelhart et al., 1986). In training, the loss of the model is defined based on the difference between the outcome y and the predicted output $\hat{y}$. The most popular loss function is Root Mean Squared Error (RMSE) (i.e., $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$). However, numerous studies have shown that the presence of outliers poses a serious threat to the standard least squares analysis (Liano, 1996). The L_1-loss provides an effective remedy that can be applied to ANN (i.e., $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$). The empirical study shows that L_1-based estimator gives better performance than that of the L_2-based algorithm (El-Melegy et al., 2009) when outliers exist.

As typical, for CATE estimation, the activation functions of hidden layers are rectified linear activation functions (ReLUs) (Nair and Hinton, 2010) and the last activation function is the identity function. ReLU is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero, and a model that uses it is easier to train and often achieves better performance. In training, the loss of the model is the

difference between the predicted output $g(x_i)$ and the target output y_i^*:

$$C(y_i^*, g(x_i); w_i^*), \tag{3.5}$$

where C is the loss function or cost function, w^* and y^* are the transformed weight and outcome in the unified formulation (3.3). Similarly, to increase the robustness, the loss function we suggest is weighted Mean Absolute Error (MAE) (i.e., $\frac{1}{n}\sum_{i=1}^{n} w_i^*|y_i^* - \hat{y}_i^*|$). The adaptive moment estimation (Adam) (Kingma and Ba, 2014), a gradient-based optimization algorithm, that run averages of both the gradients and the second moments of the gradients is adopted to train the ANN. An L_1 regularization is used in high-dimensional settings in the first layer to have a sparsifying effect by driving some weights to zero (Girosi et al., 1995; Feng and Simon, 2017). The algorithm is as follows:

Algorithm 4: Robust ANN-based CATE estimating algorithm

Input: Data $\{(Y_i, T_i, \mathbf{X}_i)\}_{i=1}^n$, number of iterations T, batch size $\mathcal{B}$, Adam parameters β_1, β_2, η, and ε, and L_1 regularization parameter λ in high-dimensional case.

Estimate nuisance quantities $p(x), \mu(x), \mu^{(1)}(x), \mu^{(-1)}(x)$ using (robust) GBM;

Calculate w_i^* and y_i^* according to Table 3.2 and Formulation (3.3);

Initialize an ANN with weights w, the decaying average of past gradients m to a zero vector, and the decaying average of past squared gradients v to a zero vector;

for *t in 1,...,T* **do**

  a. Sample a mini-batch of data $\{y^*, x, w^*\}$ without replacement with size $\mathcal{B}$;

  b. Compute the negative gradients $g^{(t)}$ based on weighted MAE;

  c. Update m and v by

  $m^{(t)} = \beta_1 m^{(t-1)} + (1-\beta_1) g^{(t)}, v^{(t)} = \beta_2 v^{(t-1)} + (1-\beta_2) {g^{(t)}}^2$;

  d. Compute bias correction terms $\hat{m}^{(t)} = \frac{m^{(t)}}{1-\beta_1^t}, \hat{v}^{(t)} = \frac{v^{(t)}}{1-\beta_2^t}$;

  e. Update the weights by $w^{(t)} = w^{(t-1)} - \eta \frac{\hat{m}^{(t)}}{\sqrt{\hat{v}^{(t)}}+\varepsilon}$.

end

Output: Weights w in the ANN.

The advantages and disadvantages of selected supervised learning algorithms are summarized in Table 3.3. Generally, GBM outperforms RF when it is well-tuned. As ANN is extremely flexible, it usually outperforms GBM and RF for image and text data. For structured data (not images or text), the representation problem is easier to solved so neural network could overkill.

Table 3.3: Comparison of selected supervised learning algorithms

Algorithm	Advantages	Disadvantages	Main Hyperparameters
Random Forest	Hard to overfit, easy to tune, good for parallel computing	Model can get large	Number of trees, number of features used in splitting
GBM	High-performing	Harder to tune than RF, sensitive to outliers, take longer to train than RF	Number of trees, depth of trees, learning rate
Neural Network	Can handle extremely complex task	Hard and slow to train	Number of neurons in hidden layers, size of mini-batches, learning rate

To make the proposed algorithms more accessible, we implemented these three algorithms in **R** package **RCATE**, every one of them can be combined with MCM-EA, RL, and AIPW to estimate CATE. The minimal required inputs are outcome, treatment assignment indicator, and pre-treatment covariates, which means there is no need for users to estimate nuisance quantities. The usage of **R** package **RCATE** is briefly introduced in Appendix B.1. [1]

Methods for estimating confidence intervals (CIs) when using proposed algorithms have not been developed. We calculated the empirical confidence interval using bootstrap. In simulation experiments, for each simulated dataset, we drew 1,000 bootstrap samples. For each subject in the testing set, the 2.5th and 97.5th percentiles of the 1,000 estimates were taken as the lower and upper bound of CI. Then the coverage probability was calculated by averaging across all subjects in the testing set and all simulation replications.

[1]The **randomForest** package (RColorBrewer and Liaw, RColorBrewer and Liaw) can be used to perform the standard L_2-based RF in **R** software (Team, 2013). The **gbm** package (Ridgeway et al., 2013) can be used to perform the standard and robust GBM. The **keras** (Allaire et al., 2019) and **tensorflow** (Allaire et al., 2016) packages can be used to perform the standard and robust ANN.

3.3 Simulation Studies

We conducted extensive simulations to evaluate the finite sample performance of the proposed algorithms. Our first set of simulations, Simulation 1, compared the proposed L_1-based algorithms, L_2-based algorithms, L_1-based generalized additive B-spline model (robust GAM) (Li, 2021), and L_1-based additive B-spline model combined with Q-learner (robust QL). The last two methods were defined below:

$$\text{Robust GAM: } \hat{\beta} = argmin_{\beta} \frac{1}{n} \sum_{i=1}^{n} w_i^*(X_i, T_i)|Y_i^* - B(X_i)^T\beta| + \Lambda_n(\beta),$$

$$\text{Robust QL: } \hat{\gamma}, \hat{\beta} = argmin_{\gamma,\beta} \frac{1}{n} \sum_{i=1}^{n} |Y_i - B(X_i)^T\gamma - \frac{T_i}{2} B(X_i)^T\beta| + \Lambda_n(\gamma, \beta),$$

where Λ is a smoothness-sparsity penalty for group-wise variable selection and controls smoothness of the regression line simultaneously. The second set, Simulation 2, compared the proposed L_1-based algorithms with existing machine learning algorithms for CATE estimation implemented in **R** in high-dimensional settings. Simulation 3, showed the trade-off between robustness to complex treatment effect function and the efficiency by comparing machine learning algorithms with additive models. We summarized the adopted methods in each setup in Table 3.4.

The design of the simulation setups followed the data structure of the real data in Section 2.5. The binary treatment levels (i.e., $T \in \{-1, 1\}$) and continuous outcome were used throughout. And we set the number of replications R is 1,000 times and the size of the validation set is $n_\nu = 1,000$. Performance was assessed using mean squared error (MSE), mean absolute error (MAE), and coverage probability (CP). The MSE and MAE

were defined as follows:

$$MAE_v = \frac{1}{R}\sum_{r=1}^{R}|\hat{\tau}^{(r)}(\mathbf{x}_v) - \tau_0(\mathbf{x}_v)|, \quad MSE_v = \frac{1}{R}\sum_{r=1}^{R}[\hat{\tau}^{(r)}(\mathbf{x}_v) - \tau_0(\mathbf{x}_v)]^2,$$

where $\mathbf{x}_v$ is the v-th observation from the validation set, $\hat{\tau}^{(r)}(\mathbf{x})$ is the estimator of $\tau(\mathbf{x})$ based on the r-th data replication. We summarized the performance over the whole validation set by taking the averages (i.e.,$\overline{MSE} = \frac{1}{n_v}\sum_{v=1}^{n_v} MSE_v$). For simplicity, we reported MSE and MAE. The tuning parameters were summarized in Appendix B.4.

Table 3.4: Candidate Methods in Each Setup

Methods under the Unified Formulation				Other Candidate Methods	
	MCM-EA	RL	AIPW	Method	
Robust RF	(1)(2)(3)	(1)(2)(3)	(1)(2)(3)	Robust QL	(1)
Robust GBM	(1)(2)(3)	(1)(2)(3)	(1)(2)(3)	Causal BART	(2)
Robust ANN	(1)(2)(3)	(1)(2)(3)	(1)(2)(3)	Causal Boosting	(2)
RF	(1)	(1)	(1)	Causal Forest	(2)
GBM	(1)	(1)	(1)	Causal MARS	(2)
ANN	(1)	(1)	(1)	X-learner+RF	(2)
Robust GAM	(1)(3)	(1)(3)	(1)(3)		

We designed following simulation setups to contrast the robustness to outcome irregularity and model complexity of the adopted methods, and to examine the performance of the methods under various proportions of outliers, distribution of errors, sample sizes, sample dimensions, and true treatment effect functions.

Simulation 1: Proposed algorithms are more robust to outliers in outcome.

We compared all considered approaches in Table 3.4 for Simulation 1 across a combination of two design factors: the proportions of outliers p_o and the outliers generating mechanisms.

We generated outcome as follows,

$$Y_i = b_0(\mathbf{X}_i) + \frac{T_i}{2}\tau_0(\mathbf{X}_i) + \varepsilon_i, \quad \varepsilon_i \sim (1-p_o)N(0,1) + p_o P.$$

We varied the two factors in two scenarios: (1) $p_o \in \{0, 0.05, 0.1, 0.15, 0.2, 0.3, 0.5\}$, $n = 1000$, and $P = N(0, 100)$, and (2) $p_o \in \{0, 0.05, 0.1, 0.15, 0.2, 0.3, 0.5\}$, $n = 1000$, and $P = Laplace(0, \sqrt{50})$.

We considered ten i.i.d. random variables, ($\mathbf{X}_i \sim N_{10}(0, 1)$) and five of them were included in the true treatment effect function. We assumed the treatment assignment mechanism is a generalized linear model of covariates and the response surface is a nonlinear model of covariates. Specifically, the treatment assignment followed a logistic model

$$D_i|\mathbf{X}_i \sim Bernoulli(p(\mathbf{X}_i)), T_i = 2D_i - 1, logit(p(\mathbf{X}_i)) = X_{i1} - X_{i2}.$$

Two functions in the response surface were

$$b_0(\mathbf{X}_i) = 100 + 4X_{i1} + X_{i2} - 3X_{i3},$$

$$\tau_0(\mathbf{X}_i) = 6sin(2X_{i1}) + 3(X_{i2} + 3)X_{i3} + 9tanh(0.5X_{i4}) + 3X_{i5}(2I(X_{i4}) - 1),$$

where the true treatment effect function is a nonlinear model of covariates and includes interactions of them.

The MSE and MAE of the CATE estimates obtained are described in Figure 3.2. The figure showed that all L_1-based algorithms outperform the L_2-based ones. Advantage of the robust algorithms increased with the proportion of outliers, under both MSE and MAE. And as the true treatment effect function was complex containing interactions of covariates, when $p_o < 0.2$, the proposed machine learning algorithms outperformed additive models in

MSE and CP (CPs were summarized in tabular form in Appendix B.2). The performance of robust GAMs were better than robust QL when the proportion of outliers was close to the breakdown point of LAD regression, i.e., $p_o = 0.5$. There were only little practical differences between the robust GBM, robust ANN and robust RF when combined with MCM-EA and R-learning. But the robust GBM didn't work well together with AIPW transformation because AIPW tends to generate transformed weights with large variance and GBM is more likely to overfit if the data is noisy (Oza and Tumer, 2008).

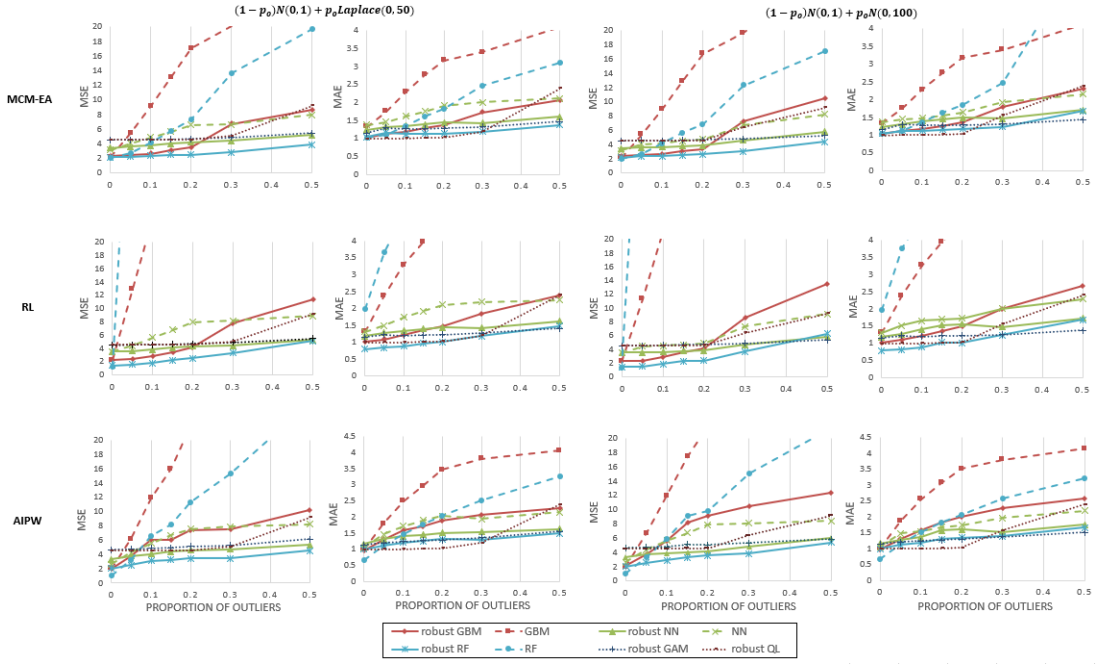


Figure 3.2: Simulation results of Simulation 1

Note: MSE and MAE values of different methods under different proportions of outliers and error generating mechanisms. The robust GBMs were indicated by red solid line, the robust RFs were indicated by blue solid line, the robust ANNs were indicated by green solid line. The GBMs, RFs, and ANNs were indicated by dashed red, blue, and green lines. The robust GAMs were indicated by blue dotted line, and robust QL was indicated by brown dotted line.

Simulation 2: Robust GBMs have good performance in high-dimensional settings.

Only the top performed methods in Simulation 1 and CATE estimating algorithms available in software **R** were used in Simulation 2. In Simulation 2, we deeply examined the performance when data is high-dimensional and outliers exist in outcome. We generated datasets with same outlier distributions P, baseline function, and propensity score function as the Simulation 1. And we fixed the proportion of outliers as 0.15, sample size as $n = 1,000$, and data dimension as $p \in \{100, 2000\}$ to compare the performance of different methods in high-dimensional case. This is a reasonable setup because many observational data are long and wide like EHR data and insurance claim data.

The true treatment effect functions when $p = 100$ and $p = 2000$ were

$$\begin{aligned}\tau_0(\mathbf{X}_i) =& 6sin(2X_{i1}) + 3(X_{i2} + 3)X_{i3} + 9tanh(0.5X_{i4}) + 3X_{i5}(2I(X_{i4}) - 1) + \\ & 3X_{i6} + 2X_{i7} + X_{i8} - 2X_{i9} - 4X_{i10},\end{aligned}$$

and

$$\begin{aligned}\tau_0(\mathbf{X}_i) =& 6sin(2X_{i1}) + 3(X_{i2} + 3)X_{i3} + 9tanh(0.5X_{i4}) + 3X_{i5}(2I(X_{i4}) - 1) + \\ & \sum_{j=6}^{50} \beta_j X_{ij}, \beta_j \sim Unif(-2, 2),\end{aligned}$$

correspondingly.

Figure 3.3 (A) and (C) showed the results when $p = 100$, the robust GBMs and robust ANN combined with AIPW and MCM-EA beated all other algorithms when outliers exist. The causal MARS was the best performing existing algorithm. Robust Rs and robust ANN combined with RL tied with causal MARS. Boosting-based algorithms generally performed better than forests-based ones. This is because a single deep tree struggles to achieve

low bias on large high dimensional data, so as the forest. Then we further increased the dimension to $p = 2000$ for robust GBMs, robust ANN combined with AIPW and MCM-EA, and causal MARS, Figure 3.3 (B) and (D) showed that robust GBMs perform the best when the data dimension is much larger than the sample size.

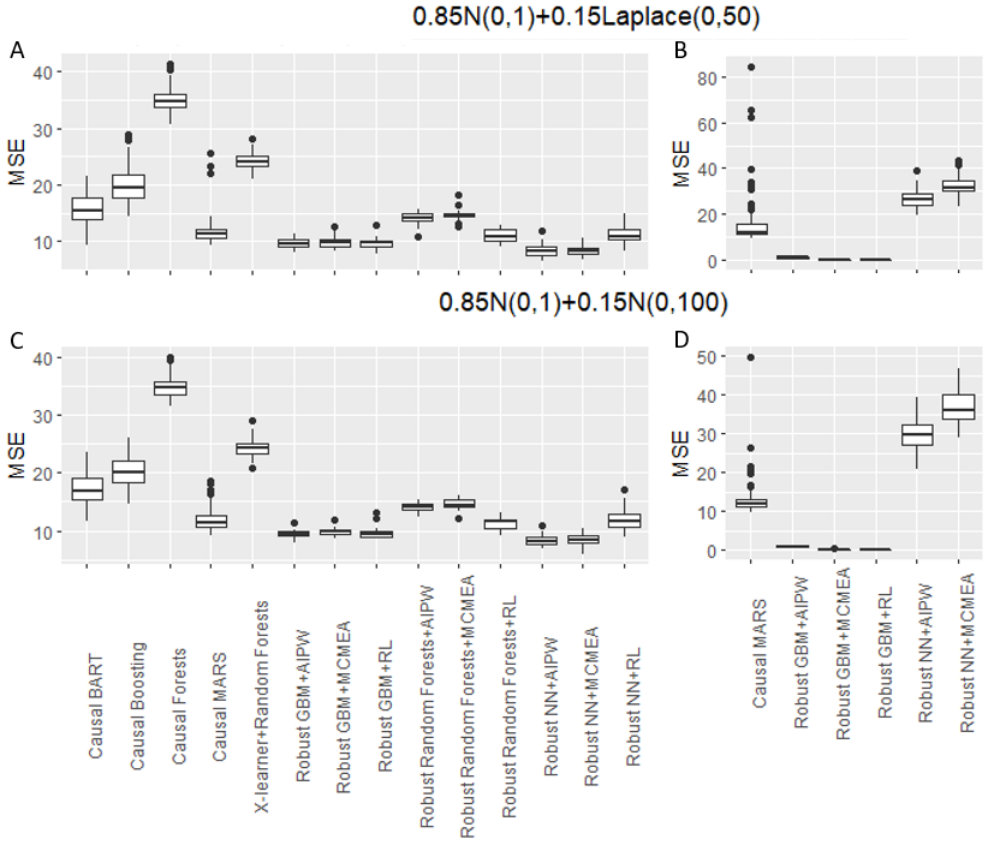


Figure 3.3: Simulation results of Simulation 2

Note: MSE of different algorithms when data is high-dimensional and outliers exist. Figures A and C showed the results when $p = 100$, Figures B and D showed the results when $p = 2000$.

Simulation 3: The trade-off between flexibility and efficiency.

In Simulation 3, we more deeply examined the trade-off between flexibility and efficiency. We conducted this setup because if the treatment effect function is additive, the flexible machine learning algorithms may be overkill, even though, in real application, the additivity

assumption is almost impossible to hold. We want to show that when the sample size available for proposed algorithms is large enough, their performance would be comparable to the performance of robust GAMs with a smaller sample size.

We used the same data dimension, outliers distribution P, the number of confounders, the treatment assignment, and outcome generating mechanisms as in Simulation 1. And we fixed the proportion of outliers as 0.15. The true treatment effect function was

$$\tau_0(\mathbf{X}_i) = 6sin(2X_{i1}) + 3X_{i2} + X_{i3} + 9tanh(0.5X_{i4}) + 3X_{i5}.$$

And for sample size, we considered two scenarios: (1) For robust GAMs, the sample size was fixed as $n_0 = 200$, and for robust GBMs, robust RFs, and robust ANNs, the sample size increased from 200 to 1000 by 200; (2) For robust GAMs, the sample size was fixed as $n_0 = 1000$, and for proposed robust algorithms, the sample size increased from 1000 to 7000 by 2000.

From Figure 3.4, we can see that as the sample size used by machine learning algorithms increase, their performance became better. Therefore, without making additive assumption, the machine learning algorithms could have as good performance when the sample size is around seven times of the sample size used by robust GAMs.

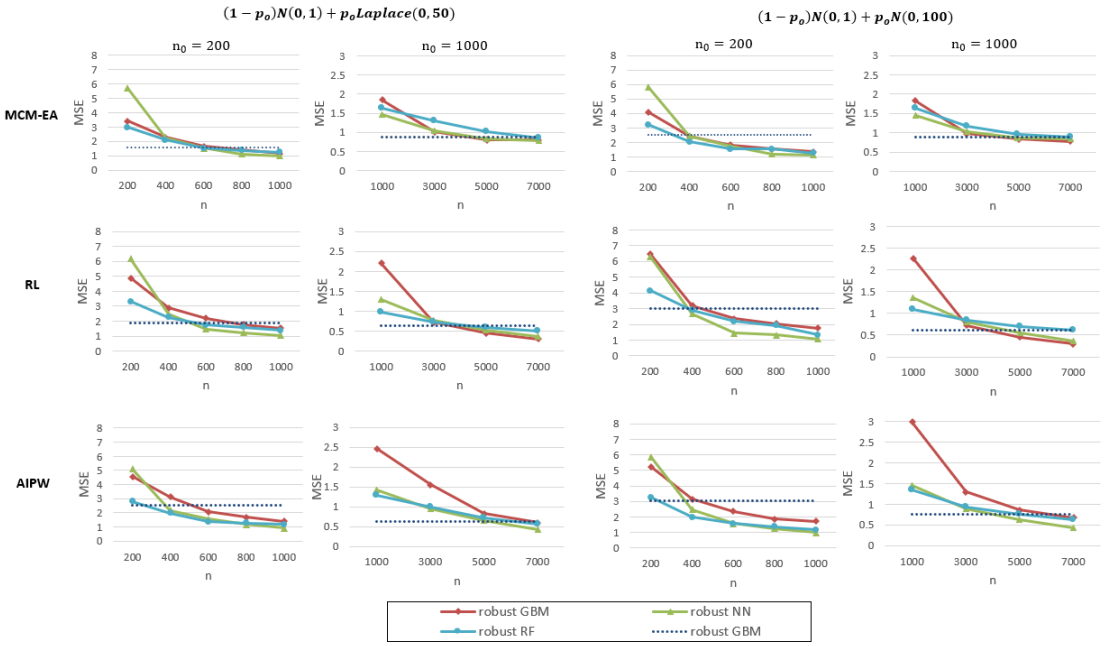


Figure 3.4: Simulation results of Simulation 3

Note: MSE of different methods under different sample sizes. The robust GBMs were indicated by red solid line, the robust RFs were indicated by blue solid line, the robust ANNs were indicated by green solid line. The robust GAMs were indicated by blue dotted line. In the first and third columns of figures, the sample size of robust GAMs methods was $n_0 = 200$; in the second the fourth columns of figures, the sample size of robust GAMs methods was $n_0 = 1000$.

We further compared the speed of proposed algorithms and additive models under difference sample sizes and dimensions of data. The robust RF was completely implemented in **R**, so that the speed is relatively slow and not included in the comparison here. We can see that (robust) GBM ran faster than other methods. And all proposed methods ran faster than robust GAM when the sample size or dimension is high.

Table 3.5: Comparison of the speed (s) of RF/GBM/ANN and additive model

Dimension	Algorithm	$n = 1000$	$n = 3000$	$n = 5000$	$n = 8000$
$p = 10$	Random Forests	0.30	1.67	3.34	7.41
	GBM	0.28	0.79	1.29	2.13
	Robust GBM	0.29	0.99	1.63	2.58
	ANN	4.72	12.87	21.43	35.89
	Robust ANN	4.51	12.63	20.90	35.25
	Robust GAM	1.65	18.94	38.23	86.18
$p = 100$	Random Forests	2.54	12.99	28.71	60.51
	GBM	2.27	6.64	11.33	18.75
	Robust GBM	2.29	7.13	12.13	19.02
	ANN	5.24	14.29	25.05	39.13
	Robust ANN	5.24	14.22	24.63	42.04
	Robust GAM	33.65	243.24	N/A	N/A

In summary, when data dimension is low and proportion of outliers is less than 0.2, proposed robust algorithms outperform other adopted methods; when data dimension is high, robust GBMs are suggested to use. In addition, robust GBMs have the fastest speed.

3.4 Data Application

We analyzed the data with the proposed algorithms. According to Section 3.3, robust RF and GBM combined with MCM-EA and R-learning performed better than the other methods when the sample size and proportion of outliers were relatively small. Therefore, we used these four methods to estimate CATE. When conditioned on the average of continuous covariates and majority of binary covariates, $\hat{\tau}$ gradually decreased with an increasing PDC (Figure 3.5), implying that the BP lowering effects of the combination therapy improved when patients becoming more adherent to the prescribed medicines. The Figure 3.5 showed that the BP-lowering effects of the combination therapy were significantly better than that of ACEI alone therapy when adherence level is close to 1 (meaning the patient is nearly

perfectly adherent to the prescribed medicines), a finding that is more in line with the clinical expectation and consistent with existing knowledge and the results in Section 2.5.

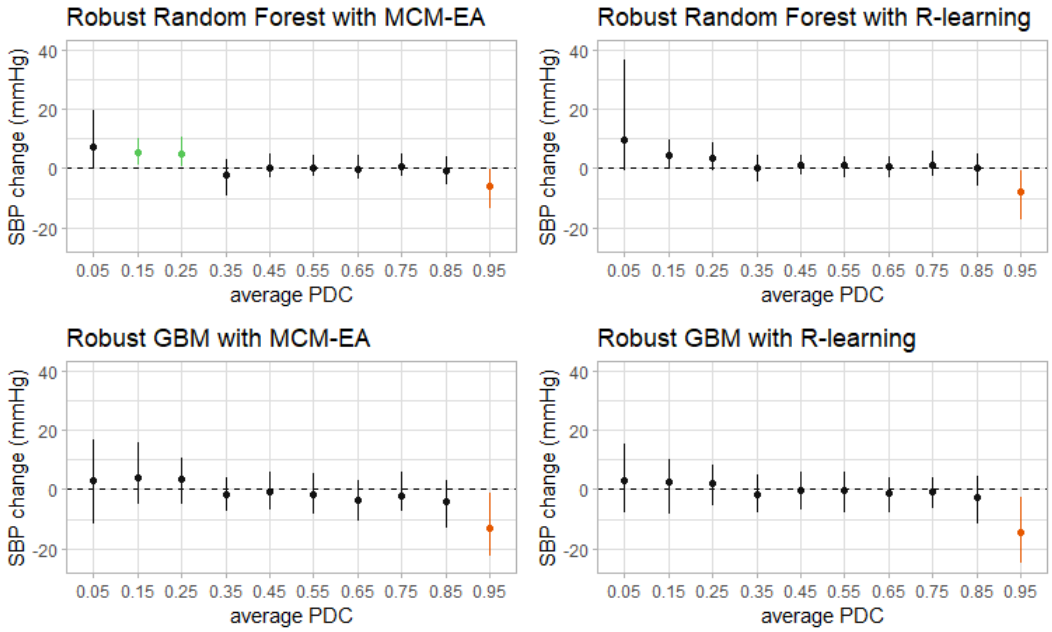


Figure 3.5: Marginal treatment effect of PDC.

Note: If the empirical 95% C.I. not covers zero, the interval segment was colored green or red.

We further showed the joint treatment effect of PDC and BMI. Figure 3.6 showed that the combination therapy works the best for hypertensive patients whose BMI is between 30 and 40.

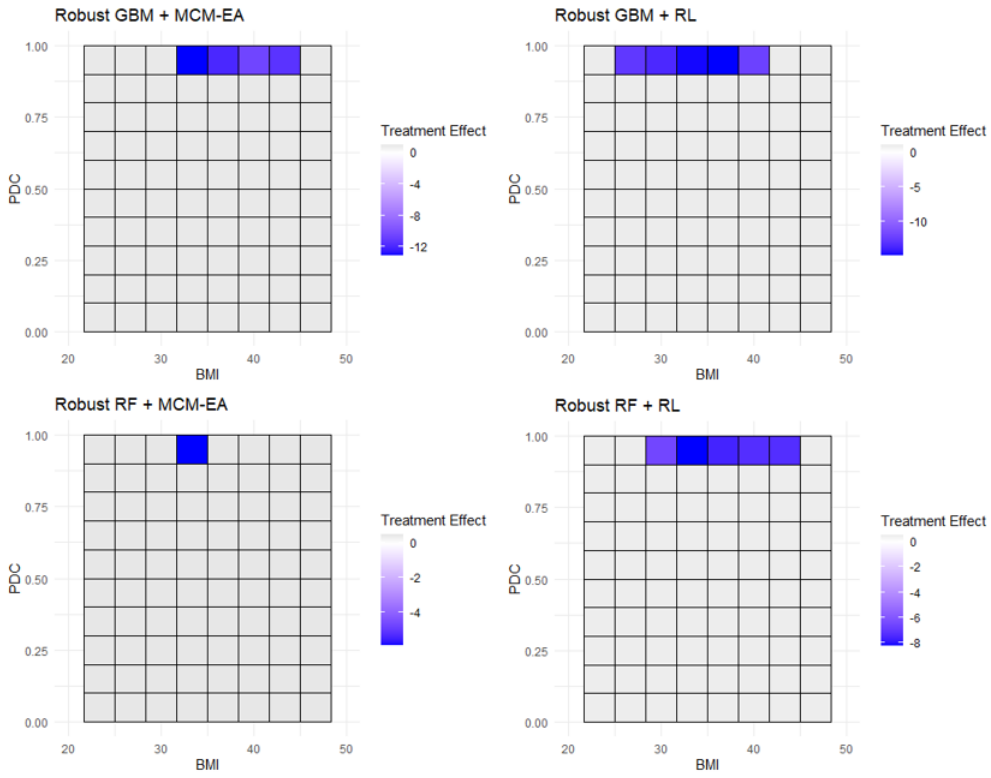


Figure 3.6: Joint treatment effect estimation of PDC and BMI.

Note: The estimates were represented by various colors. The estimates less than zero are in blue, and the estimates whose empirical 95% C.I. covers zero are in grey.

The p-values of the conditional independence test were summarized in Table 3.6. The conditional independence error assumption held for four adopted methods at the significant level of 0.05.

Table 3.6: Conditional independence test results (p-value)

Method	IRD-test	IEP-test	ICQP-test	ITP-test	IRP-test
Robust RF + MCM-EA	0.46	0.29	1.00	1.00	0.26
Robust RF + RL	0.33	0.29	0.57	1.00	0.28
Robust GBM + MCM-EA	0.65	0.81	0.24	0.43	0.66
Robust GBM + RL	0.21	0.71	1.00	0.43	0.34

3.5 Discussion

In the previous chapter, we described a general estimating equation for heterogeneous treatment effect estimation. The formulation is highly flexible and can accommodate high dimensional observational data with various forms of irregularity. Nevertheless, the method requires correct specification of the treatment effect function. Although we attempted to alleviate the constraint by using a more flexible additive structure, there is no guarantee that the additive structure is, in fact, correct. The approach, although theoretically appealing, cannot be readily applied in many analytical situations, especially when the treatment effect functions contained interactions among the independent variables.

In this chapter, we extend the robust estimation methods for heterogeneous treatment effects. We further reduce the methods' reliance on correct model specification by using algorithm-based machine learning techniques, including random forest, gradient boost machine, and artificial neural network, to determine the correct model formulation. In a sense, we let the model specification be data-driven. In doing so, we hope to retain the good theoretical properties described in the previous chapter while making the estimation procedure more robust against model misspecification. The essence of this general approach is to use machine learning techniques to optimize the common objective function. Simulation results confirm that the new procedures' good performance. In addition to the added robustness against model misspecification, the added flexibility of the methods further enhances the general scalability of the robust heterogeneous treatment effect estimators.

Chapter 4

Reinforcement Learning for Dynamic Treatment Recommendation

Recommendation of a particular regimen to optimize the treatment outcome in an individual patient is perhaps the most essential task of precision medicine. In real-world therapeutics, the task has to be accomplished in a dynamic setting: drugs are initiated or stopped based on a patient's response, often in an environment depicted by a Markov process. Such a setting would make reinforcement learning a suitable technique for treatment selection. However, depending on the management practice of specific diseases and the data sources used to train the policy, reinforcement learning techniques must be modified to achieve this goal.

4.1 Treatment recommendation: An application in hypertension

In this chapter, we describe a reinforcement learning (RL) algorithm for recommending antihypertensive therapies using a static data source. For narrative convenience, we present the methodological development in the context of hypertension treatment, a setting that presents some unique challenges. We first describe the context of this research.

Hypertension is a major contributor to mortality and morbidity in the United States. Nearly half a million of people died in 2018 because of hypertension-related sequelae. Widespread monitoring and control of blood pressure is not optimal; many individuals with elevated blood pressure do not know that they are hypertensive. By some estimates, only about 1 in 4 adults (24%) with hypertension have their blood pressure controlled at a level below 130/80 mmHg (Centers for Disease Control and Prevention, 2019). The latest

evidence suggests that reducing systolic blood pressure (SBP) to 120 mmHg would produce the greatest health benefit (SPRINT Research Group, 2015).

Pharmacological therapy is the mainstay approach for lowering blood pressure, while lifestyle modifications can aid in the management of blood pressure. However, the current pharmacological treatments have been largely empirical. With a large number of antihypertensive agents available, physicians rely on published evidence and their own experience to determine how to treat individual patients, often by trial and error. Adding to the difficulty, there are many antihypertensive drugs, relying on different mechanisms to lower blood pressure. In addition, combination therapies are commonly used to gain better treatment effects. Moreover, the optimal treatment may change over time, according to the patient's dynamic responses, and the treatment effect is also influenced by the variations in the patient's physiological and pathological states.

Therefore, an evidence-based personalized treatment recommendation system is needed to assist doctors in making treatment decisions. A well-designed system could protect patients from adverse events and save time and money spent trying different drugs or putting patients on more drugs than necessary. An excellent source of data for the development of such a system is electronic health records (EHRs). Large clinical studies also generate useful data. In this work, we built an evidence-based recommendation system for antihypertensive agents using data from the Systolic Blood Pressure Intervention Trial (SPRINT) (SPRINT Research Group, 2015). SPRINT was a large clinical trial aimed at lowering SBP in patients with essential hypertension. We used RL as a general approach but modified the algorithms to address the unique challenges in this study.

4.2 Data source

The research question and data source we used are from the SPRINT study that was a large clinical trial designed to reduce cardiovascular morbidity and mortality by setting a lower SBP target of less than or equal to 120 mmHg. Although it was a randomized trial, its purpose was not to test the efficacy of specific drugs. Instead, the SPRINT study left the therapeutic decisions to the physicians. For this reason, it provides a platform for the development of a drug recommendation system.

The SPRINT study recruited and followed 9,361 patients who were at least 50 years old, with an SBP of 130-180 mmHg and increased cardiovascular risk, but without diabetes. Participants were randomly assigned to the intensive treatment group (SBP target of 120 mmHg or lower, n=4,678), and the standard treatment group (SBP target of 120-140 mmHg, n=4,683). Participants were seen monthly for the first 3 months and every 3 months thereafter for up to 5 years. Medications for participants were adjusted based on the most recent visits to reach the target SBP. Demographic data and clinical/subclinical chronic vascular and kidney diseases were recorded at baseline. Laboratory data, cardiovascular and kidney diseases, prescriptions, and SBP measurements were updated in each visit. The most common antihypertensive regimens used in the SPRINT study were thiazide-type diuretics, angiotensin-converting-enzyme inhibitors (ACEIs), angiotensin II receptor blockers (ARBs), calcium channel blockers (CCBs), alpha/beta-blockers, and combinations of these drugs (see Figure 4.1).

We considered the optimization of treatments within the framework of a dynamic treatment regimen (DTR). A DTR is a sequence of regimens tailored by the dynamic states of patients. In DTR recommendation, not only the immediate but also the long-term treatment effects are considered in order to account for the delayed effects of the current treatment as well as the effects of the future treatments. Therefore, an optimal DTR is de-

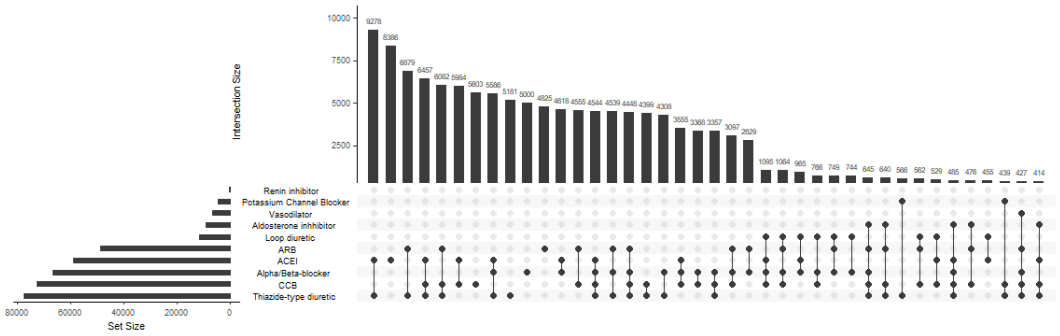


Figure 4.1: The frequency of use of each of the antihypertensive drug classes and their combinations in the SPRINT data.

termined by optimizing the long-term evaluation metric related to the outcome of interest. Developing such a system to find the optimal DTR is rarely a straightforward process. We approach the problem by taking into account the unique features of the SPRINT data.

We modeled the DTR as a Markov decision process (MDP). Within such a system, a policy can be trained by an RL algorithm to return an optimal regime that maximizes the future reward in a given dynamic state. RL has been applied to many chronic diseases, including heart disease, cancer, diabetes, anemia, HIV, and mental diseases (Gottesman et al., 2019; Yu et al., 2019; Liu et al., 2020). However, RL has not been applied to long-term blood pressure control because of the unique challenges in the application.

4.3 DRT Recommendation using Reinforcement Learning

4.3.1 Preliminaries on Reinforcement Learning

In this research, the DTR is modeled as an MDP with a deterministic policy. The MDP is formed by $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state space and action space, respectively. At a given discrete time step t, an RL agent takes action $a \in \{0,1\}^K$ from $\mathcal{A}$ in state $s \in \mathcal{S}$, and receives a new state $s' \in \mathcal{S}$ and a reward $r(s, a, s')$ based on the transition dynamics $p(s', r|s, a)$. K is the number of action dimensions that will be introduced in the following subsection. The agent makes decisions by its policy $\pi : \mathcal{S} \to \mathcal{A}$. For a given policy

π, the state-action value function (or Q-function) is defined as the expected reward of an agent, i.e., $Q^\pi(s,a) = E_\pi[R_t|s,a]$, where $R_t = \sum_{i=t}^{\infty} \gamma^{(i-t)} r(s_i, a_i, s_{i+1})$ is the cumulative discounted future reward (or return) from time step t, and γ is the discount factor that determines the effective horizon by weighting future rewards. The goal of RL is to find an optimal policy that attains the maximum Q-value. Mathematically, the optimal value function is defined as $Q^*(s,a) = \max_\pi Q^\pi(s,a)$ for any π, s, and a. For $\gamma \in [0,1)$, the optimal Q-function is the unique solution to the Bellman optimality equation (Bellman, 1966; Bertsekas and Tsitsiklis, 1996),

$$Q^*(s,a) = \sum_{s' \in \mathcal{S}} p(s', r|s, a)[r(s, a, s') + \gamma \max_{a'} Q^*(s', a')].$$

In deep RL, a neural network Q_θ is used to approximate the Q-function. In the Deep Q-Network (DQN) (Mnih et al., 2015), the Q-function was updated by minimizing the temporal difference (TD) error through Q-learning (Watkins, 1989):

$$\mathcal{L}_{RL}(\theta) = \sum_{(s,a,s',r) \in \mathcal{M}} l_\delta(r + \gamma \max_{a'} Q_\theta(s', a') - Q_\theta(s, a)), \tag{4.1}$$

where l_δ is the Huber loss (Huber, 1992):

$$l_\delta(x) = \begin{cases} 0.5x^2 & \text{if } x \leq \delta, \\ \delta(|x| - 0.5\delta) & \text{otherwise.} \end{cases}$$

We used Huber loss to avoid large TD error gradients. The loss was minimized over mini-batches $\mathcal{M}_i, i = 1, ..., I$ of sampled transitions (s, a, s', r), where I is the number of iterations. To maintain a relatively fixed target, the double DQN (DDQN) was proposed by Van Hasselt et al. (2015). The two DQNs are a primary network Q_θ and a target network $Q_{\theta'}$ that are initialized with the same parameters, i.e., $\theta' = \theta$. In the DDQN, the objective function

(4.1) becomes

$$\mathcal{L}_{RL}(\theta) = \sum_{(s,a,s',r)\in\mathcal{M}} l_\delta(r + \gamma Q_{\theta'}(s', a') - Q_\theta(s, a)), \text{where } a' = argmax_{a'} Q_\theta(s', a'). \quad (4.2)$$

With the above loss function, a target network $Q_{\theta'}$ can be soft-updated (Lillicrap et al., 2015) by $\theta' = \eta\theta + (1-\eta)\theta'$, where η is a parameter with a value close to zero. An extension of the DDQN is the dueling double DQN (D3Q) (Wang et al., 2016). The D3Q explicitly separates the representation of state value and state-dependent action advantages into two separate streams while sharing a common learning module. The two streams are then combined into a special aggregating layer to produce an estimation of the Q-function,

$$Q(s, a) = V(s) + \left(A(s, a) - \frac{1}{|\mathcal{A}|} \sum_a A(s, a) \right).$$

The dueling network architecture has been shown to lead to better policy evaluation in the presence of many similar-valued actions, and thus, it achieves faster generalization over large action spaces (Wang et al., 2016).

As introduced above, an RL agent does not learn by mimicking the actions in the data set. Instead, an RL agent learns based on interactions with the environment, which distinguishes RL from supervised learning (SL), where the correct (or best) actions need to be provided. A clear limitation of SL is that the behavior cloning agent can only be as good as the human it is imitating. However, RL can outperform the human. Also, in RL, the environment does not need to be modeled. Model-free RL algorithms can directly develop a control policy based on interactions with the environment. RL also considers the delayed effect of current action and the effect of future actions, which uniquely suits the DTR recommendation task because treatments usually have delayed effects and the regimens do not change dramatically over time (see Figure 4.2). Moreover, RL can return a personalized

optimal treatment based on a patient's dynamic state, rather than on the average treatment effect from a randomized clinical trial. These features make RL an attractive solution to construct a DTR recommendation system for chronic diseases.

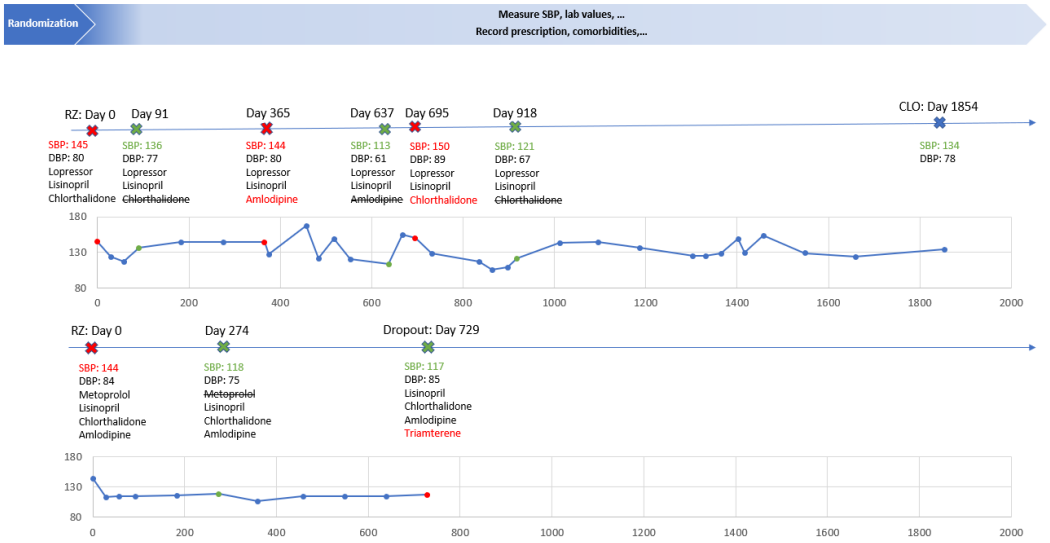


Figure 4.2: The SBP, dystolic blood pressure (DBP) measurements, and prescriptions of two patients in the standard group (top) and the intensive group (bottom) in the SPRINT study.

4.3.2 Challenges and related works

There is a growing literature on healthcare applications of RL. However, there are several unique challenges that made RL ill-fitted in DTR recommendation. First, when EHR or large study data are the main source of training, the data source is observational. The regimens are prescribed by experienced and knowledgeable physicians. We would not want to apply policies that are not well learned on real patients, for safety and ethical reasons. In other words, continuous data collection is not always possible. And many standard RL algorithms have been shown to fall short in offline settings (i.e., situations of learning using static data). For some diseases, this is not a big problem because some accurate simulators are available to simulate the complex interaction between treatments and the outcome. For example, the chemotherapy - tumor growth model of cancer (De Pillis and

Radunskaya, 2003) and the glucose - insulin dynamic system of diabetes (Daskalaki et al., 2010). However, for hypertension, many drugs can be used for blood pressure management; but the mechanisms of drug action can be quite different. So, there are no generally accepted antihypertensive drugs - SBP model for use. Second, the number of antihypertensive drugs is large, and they can be taken alone or in combination to achieve the desired effect. As a result, the action space will increase exponentially with the number of drug classes. It is difficult to efficiently explore the large action space with a fixed data (Lillicrap et al., 2015). Third, safety of the recommended regimen by RL policy cannot be guaranteed because the method learns by optimizing the Q-value at particular state. A less desirable situation is to have a regimen containing too many working drugs, a phenomenon known as polypharmacy.

In this section, we first review the related methods, and then we propose remedies to address these difficulties simultaneously.

Offline reinforcement learning.

In offline RL, the data set is static and has no additional accumulation because no further online interactions with the environment occur. Decision-making in healthcare is usually made in the offline setting (Levine et al., 2020). In offline learning, when selected actions $a' = argmax_{a'} Q_\theta(s', a')$ in (4.2) are barely seen in the static data, the Q-function $Q_{\theta'}(s', a')$ could lead to an extrapolation error (Fujimoto et al., 2019) or a distribution shift (Levine et al., 2020). The standard RL algorithms may diverge or otherwise yield poor performance in the offline setting (Fujimoto et al., 2019; Agarwal et al., 2019). Modern offline RL algorithms that have been proposed recently have been shown to work in the offline setting. They can be categorized into two categories: policy constraint algorithms (Fujimoto et al., 2019; Laroche et al., 2019; Kumar et al., 2019; Jaques et al., 2019) and some less conservative algorithms (Yu et al., 2020; Kidambi et al., 2020; Kumar et al., 2020; Agarwal et al., 2020). Policy constraint methods require behavior policy estimation and set

a constraint in various ways to avoid an extrapolation error. The less conservative methods tackle the extrapolation error by adding a regularization or a penalty of uncertainty to the reward or value function. Policy constraint methods often work well when the behavior policy distribution is easy to model and the static data is from human experts. Batch-constrained deep Q-learning (BCQ) (Fujimoto et al., 2019), a policy constraint algorithm, is one of the best offline RL algorithms that operates in a discrete action space. Basically, BCQ uses a generative model $G_\omega(a|s)$ to compute the probabilities of each action, given a state, and utilizes some threshold to eliminate actions that are unlikely to be contained in the fixed data. The action selection step in (4.2) becomes:

$$a' = \underset{a' \Big| \frac{G_\omega(a'|s')}{max_{\hat{a}} G_\omega(\hat{a}|s')} > \tau}{\arg\max} Q_\theta(s', a').$$

As shown in the experimental results (Fujimoto et al., 2019), offline RL can outperform SL. Unlike SL, which mimics the choice in static data, offline RL tries to find good choices in the data, and then recombines and applies the good choices to other subjects.

Large discrete combinatorial action space. Even though D3Q can handle a relatively large discrete action space, it is not efficient enough when the action space is combinatorial. Metz et al. (2017) developed an approach to sequentially predict the action value. This required manually ordering the action dimensions, which is hard in clinical decision-making. Independent DQN (IDQ) (Tampuu et al., 2017) combines DQN with independent Q-learning. In IDQ, each agent independently and simultaneously learns its own action value function. The Branching Dueling Q-Network (BDQ) (Tavakoli et al., 2017), a variant of D3Q, uses a common learning module for state and all action dimensions. Each semi-independent action branch returns the advantages of the sub-actions for that action dimension (see Figure 4.3). Formally, for an action dimension $d \in 1, ..., K$ with $|\mathcal{A}_d|$ discrete

sub-actions, the branch d's Q-value at state $s \in \mathcal{S}$ with sub-action $a_d \in \mathcal{A}_d$ is expressed in terms of the common state value $V(s)$ and the corresponding sub-action advantage $A^d(s, a_d)$ by:

$$Q^d_\theta(s, a_d) = V_\theta(s) + \left[A^d_\theta(s, a_d) - \frac{1}{|\mathcal{A}_d|} \sum_{a'_d \in \mathcal{A}_d} A^d_\theta(s, a'_d) \right]. \tag{4.3}$$

BDQ has been shown to scale robustly to environments with high-dimensional action spaces to solve the benchmark domains (Tavakoli et al., 2017).

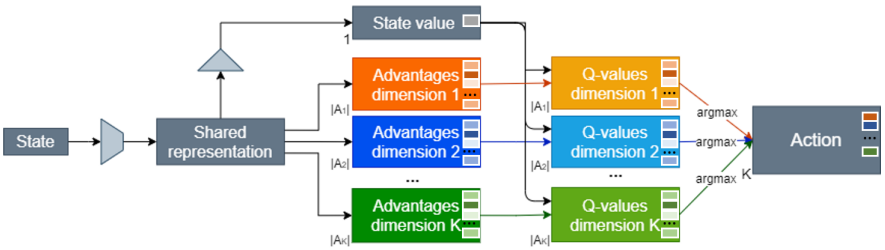


Figure 4.3: A visualization of the architecture of BDQ.

In the example of hypertension, the number of action dimensions is the number of different antihypertension drug classes K, and the sub-actions are taking the drug (1) or not taking the drug (0). For D3Q, the size of the output layer is 2^K. With the branching structure, the number of neurons in the output layer decreases from 2^K to $2K$.

Medication safety. Recently, algorithms that combine RL and SL to avoid the risks of actions and to improve learning efficiency have been proposed. Common examples are supervised actor-critic (Rosenstein et al., 2004) and RL from demonstrations (Vecerik et al., 2017; Hester et al., 2018). The supervised actor-critic uses expert behaviors to pre-train a "guardian" and sends low-risk actions to train the agent. However, SL and RL agents cannot learn from each other in the training process. The RL from demonstrations does pre-training solely using demonstration data (or expert behaviors) and by combining the TD error and SL error to get a reasonable policy as the start point. Then, both use demonstration data

and self-generated data for updating. While effective, RL from demonstrations is inadequate for offline learning, since it requires further data collection or access to an oracle (Fujimoto et al., 2019). These requirements are not in line with clinical reality. In addition, some works combine actor-critic and SL to optimize the parameter θ jointly in critical care applications (Wang et al., 2018; Yu et al., 2020). The objective function is:

$$\mathcal{L}(\theta) = (1 - \epsilon)\mathcal{L}_{RL}(\theta) + \epsilon\mathcal{L}_{SL}(\theta), \tag{4.4}$$

where $\mathcal{L}_{RL}(\theta)$ is the objective function of RL task, which tries to maximize the expected return, $\mathcal{L}_{SL}(\theta)$ is the objective function of SL task, which tries to minimize the difference between selected the action and the doctor's prescription, and ϵ is a weight parameter to trade off the RL and SL tasks.

4.4 Proposed Modification

In this section, we describe a new algorithm, called supervised batch-constrained branching dueling double deep Q-network (SBC-BDQ), for simultaneously handling all of the challenges in DTR recommendation for hypertension patients. The components in SBC-BDQ are generalizable to DTR recommendation for other chronic diseases.

First, for blood pressure control, no generally-accepted metrics are available to quantify the *long-term* effects of a recommended regimen. In critical care, mortality is generally used for evaluation. However, for hypertension and many other chronic diseases, a more appropriate evaluating metric is needed. We propose a weighted moving average-based metric that assigns larger weights to more recent blood pressure observations for evaluating long-term blood pressure maintenance. This idea can be easily generalized to other applications whose goal is to maintain biochemical indexes (e.g., blood glucose for diabetes). Second, SBC-BDQ applies modern offline RL to DTR recommendation. This is crucial because

continuous data accumulation is not available in most healthcare applications. Also, an accurate simulator is not available for many chronic diseases. To reduce extrapolation error, we used batch-constraint in offline learning because policy constraint algorithms generally perform well when static data is collected by an expert, and in healthcare applications, an aggressive constraint on action is needed. In the proposed algorithm, we extended the idea of batch-constraint to a combinatorial action space case. Third, to avoid learning inefficiently, we adopted the branching architecture proposed by Tavakoli et al. (2017) for DTR recommendation. If the dose level is available in the data source, the proposed algorithm for DTR recommendation can be easily extended to do a DTR dosage search. In addition to Q-value branches, we added supervised branches in the architecture to guarantee the medication safety. With the modified architecture, we combined deep Q-learning and SL in network updating. The results showed that supervision from experts can improve long-term blood pressure maintenance.

We present the architecture of SBC-BDQ in Figure 4.4. Each component of SBC-BDQ is introduced in detail in following paragraphs.

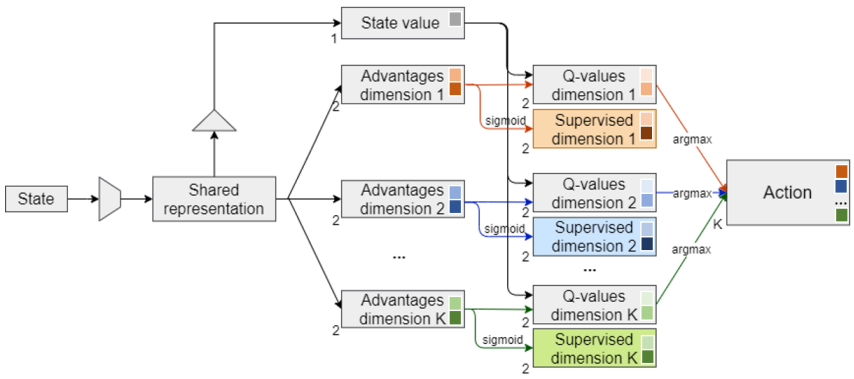


Figure 4.4: A visualization of the proposed SBC-BDQ agent. Each branch is made up of the Q-value dimension and the supervised dimension.

Common State-Value Estimator

SBC-BDQ uses a common learning module for state and all action dimensions (see Figure 4.4). The Q-value of branch d is represented by

$$Q_\theta^d(s, a_d) = V_\theta(s) + \left[A_\theta^d(s, a_d) - \frac{1}{2} \sum_{a'_d \in \mathcal{A}_d} A_\theta^d(s, a'_d) \right]. \tag{4.5}$$

Finally, with the well-learned policy, in each action branch d, the sub-action a_d^* that maximizes the $Q_\theta^d(s, a_d)$ in state s will be recommended. The final recommended action is $a^* = (a_1^*, ..., a_K^*)$. The supervised dimensions in Figure 4.4 are added for combining RL and SL, and they will be introduced later.

Batch-Constraint for Combinatorial Action Space

In BCQ (Fujimoto et al., 2019), the conditional probability of action is predicted by a multi-class classification neural network. A threshold is utilized to eliminate actions that are not possible for physicians to select at a state. To adapt the large combinatorial discrete action space, we need to use multi-label classification neural network as a generative model G_ω, where multiple labels can be assigned to each instance. Different from a multi-class classification network whose outputs sum up to one, the last activation function of a deep multi-label classification neural network is a sigmoid instead of a softmax function; therefore, the outputs do not add up to one. Formally, the outputs of a multi-label classification network are $p_\omega = (p_\omega^1, ..., p_\omega^K) \in (0, 1)^K$. The conditional probability of each action is calculated as $G_\omega(a|s) = \prod_{d=1}^K p_\omega^d(s)$. Then, a threshold τ is used to eliminate actions:

$$a'_d = \underset{a'_d : a' = [a'_1, ..., a'_K] \Big| \frac{G_\omega(a'|s')}{max_{\hat{a}} G_\omega(\hat{a}|s')} > \tau}{\arg\max} Q_\theta^d(s', a'_d).$$

To adaptively adjust this threshold, we scale it by the maximum probability from the generative model over all actions. This allows only actions whose relative probability is above some threshold τ. The generative model G_ω is trained by minimizing the binary cross-entropy:

$$H = \sum_{(a,s)\in\mathcal{M}} \frac{1}{K} \sum_{d=1}^{K} -a_d log p_\omega^d(s) - (1-a_d) log(1-p_\omega^d(s)). \tag{4.6}$$

The RL part in the objective function is defined as follows:

$$\mathcal{L}_{RL}(\theta) = \sum_{(s,a,s',r)\in\mathcal{M}} \frac{1}{K} \sum_{d=1}^{K} l_\delta \left(r + \gamma Q_{\theta'}^d(s', a'_d) - Q_\theta^d(s, a_d) \right).$$

Combining Reinforcement Learning and Supervised Learning

For each action dimension, the softmax function f is applied to adjusted sub-action advantages to values between zero and one, i.e.,

$$q_\theta^d(s) = f\left(A_\theta^d(s, a_d) - \frac{1}{|\mathcal{A}_d|} \sum_{a'_d \in \mathcal{A}_d} A_\theta^d(s, a'_d) \right) \in (0,1)^{|\mathcal{A}_d|}.$$

For prescriptions by the doctors, we transferred each drug assignment a_d to $\mathbf{a}_d^*$ by one-hot encoding. For example, when the sub-action space size $|\mathcal{A}_d| = 2$, we transferred $(1) \to (0,1)$ and $(0) \to (1,0)$. Then, we minimized the difference between $q_\theta(s)$ and the doctors' decisions $\mathbf{a}^*$ based on binary-cross entropy as the SL term in the objective function:

$$\mathcal{L}_{SL}(\theta) = \sum_{(s,a)\in\mathcal{M}} \frac{1}{K} \sum_{d=1}^{K} \frac{1}{|\mathcal{A}_d|} \sum_{j=1}^{|\mathcal{A}_d|} -a_{dj}^* log(q_{\theta j}^d(s)) - (1-a_{dj}^*) log(1-q_{\theta j}^d(s)).$$

The whole objective function was then the weighted sum of $\mathcal{L}_{RL}(\theta)$ and $\mathcal{L}_{SL}(\theta)$ as in (4.4). We summarize the SBC-BDQ in algorithm 5.

Algorithm 5: SBC-BDQ

Input: Batch $\mathcal{B}$, number of iterations I, target network soft update rate η, mini-batch size M, weight parameter ϵ, number of classes of drugs K, threshold τ.

Initialize primary Q-network Q_θ, target Q-network $Q_{\theta'}$ with $\theta' \leftarrow \theta$, and generative model G_ω;

for $i = 1$ *to* I **do**

  Sample mini-batch $\mathcal{M}$ contains M of N transitions (s, a, r, s') from $\mathcal{B}$;

  $a_{G_\omega} \leftarrow$ action given by G_ω, where

  $a_{G_\omega} = \big\{[a^1_{G_\omega}, ..., a^K_{G_\omega}] : a^d_{G_\omega} = I(p^d_\omega(s) > 0.5), d = 1, ..., K\big\}$;

  $a' = [a'_1, ..., a'_K]$, where $a'_d = \arg\max_{a'_d : a' = [a'_1, ..., a'_K] \big| \frac{G_\omega(a'|s')}{\max_{\hat{a}} G_\omega(\hat{a}|s')} > \tau} Q^d_\theta(s', a'_d)$;

  Perform an Adam[1]step on $(1 - \epsilon)\mathcal{L}_{RL}(\theta) + \epsilon\mathcal{L}_{SL}(\theta)$ with respect to θ ; `// update primary Q-network`

  Perform an Adam step on (4.6) with respect to ω ; `// update generative model`

  $\theta' \leftarrow \eta\theta + (1 - \eta)\theta'$; `// update target Q-network`

end

We used TensorFlow (Abadi et al., 2015) to implement SBC-BDQ.

4.5 Data analysis

4.5.1 Dataset and Cohort

Based on the result of the SPRINT study showing that keeping SBP below 120 mmHg is better than 120-140 mmHg, our target SBP was set to below 120 mmHg. Therefore, we only used transitions $\{s, a, r, s'\}$ from the intensive group for training. In the intensive

[1]Adaptive Moment Estimation (Adam)(Kingma and Ba, 2014) is an optimizer.

group, since more than half of the patients were on three or more classes of drugs and had a well-maintained SBP, it was difficult to further lower SBP without adding too many drugs. However, we could determine if the learned policy could help to recommend a better regimen for patients who were not doing well in the SPRINT study. As shown in Figure 4.2, even if a patient was on the same set of drugs, the SBP could change dramatically. Therefore, we identified a cohort of transitions whose SBP increased after drug change (i.e. $SBP_{t+1} > SBP_t$ and $a_{t+1} \neq a_t$) as a validation and testing set (see Figure 4.5). Finally, we obtained 53,753 transitions from 4,436 patients as a training set, and splitted 7,470 transitions from 4,705 patients 50:50 into the validation and testing sets.

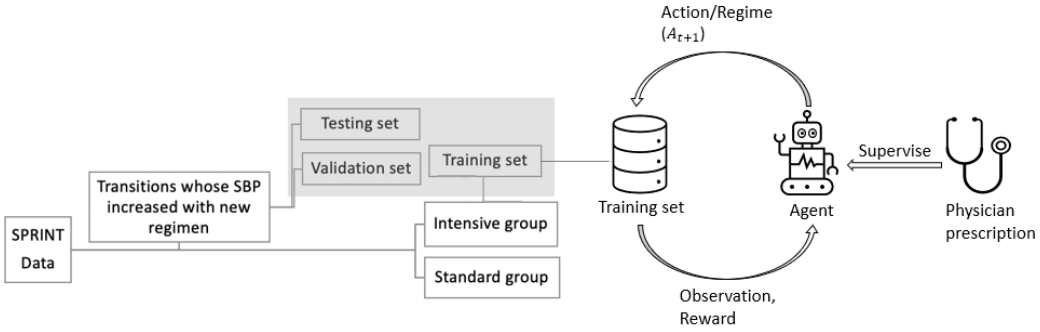


Figure 4.5: The data processing and architecture of SBC-BDQ.

4.5.2 Settings of Reinforcement Learning

For each patient, we extracted relevant parameters that included baseline and time-series variables. The baseline patient characteristics included gender, age, race, body mass index (BMI), SBP, smoking status, aspirin usage, statin usage, chronic kidney disease (CKD), and cardiovascular disease (CVD) conditions. The time-series variables included lab values, prescriptions, heart rate, adverse events, and the outcome of interest (i.e., SBP). The adverse events included myocardial infarction, acute coronary syndrome not resulting in myocardial infarction, stroke, acute decompensated heart failure, dialysis, and kidney transfer. At time step t, the features mentioned above corresponded to the state s_t in the MDP. We imputed

the missing lab values with multiple imputation, carried forward the dialysis and kidney transfer condition, and removed transitions with a missing value in other variables. The action was the following regimen prescribed, depending on state s_t, that contains the 5 most commonly used drug classes and their combinations, i.e., $a_t \in \{0,1\}^5$. Note that transitions whose current and following prescriptions contain other drug classes are not used in this experiment. The reward at time $t+1$ is based on following SBP_{t+1},

$$r_{t+1} = \begin{cases} 1 & \text{for} \quad SBP_{t+1} < 120 \\ -1 & \text{for} \quad SBP_{t+1} > 140 \\ 0 & \text{for} \quad otherwise. \end{cases}$$

The relationship between hospital visits and the settings of RL is shown in Figure 4.6. For simplicity, except for the outcome of interest (SBP), other variables included in state are not shown in the figure.

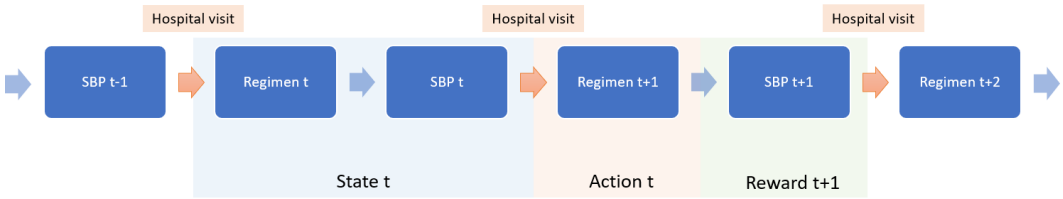


Figure 4.6: Demonstration of the relationship between hospital visits and state, action, and reward setting.

4.5.3 Evaluation Metrics

Evaluation methodology in DTR recommendation is a challenge, especially for chronic diseases, due to the lack of long-term evaluation metrics. Thus, we propose a new moving average-based metric to evaluate our algorithm. We used the long-term SBP maintenance ($LSPM$) indicator, defined below, to measure whether policies would be helpful to keep

the SBP below 120 mmHg for a long time period or not,

$$LBPM_t = I\left[\frac{1}{w_1+w_2+w_3}(w_1 SBP_{t+1} + w_2 SBP_{t+2} + w_3 SBP_{t+3}) < 120\right],$$

with $w_{t+x} = 1/(\text{number of months between } SBP_t \text{ and } SBP_{t+x})$, and the corresponding long-term SBP maintenance score (abbreviated as maintenance score in the rest of the chapter) is defined as $p(LBPM_t = 1)$. This metric considers the three SBP measurements following assignment of the current regimen. Only the SBP measured in next 3 to 9 months are covered by this metric because the effect of antihypertensive drugs is usually immediate and can show up in several weeks. The $LBPM$ is calculated for every transition in the validation set. Then we fitted a curve to model the maintenance score using the Q-value of the prescribed regimen as shown in Figure 4.7. Based on Figure 4.7, the $LBPM$ is positively correlated with the expected returns for all adopted methods. With these curves, we can calculate the corresponding maintenance score of the recommended regimen. The adopted methods and their implementation will be introduced in next subsection.

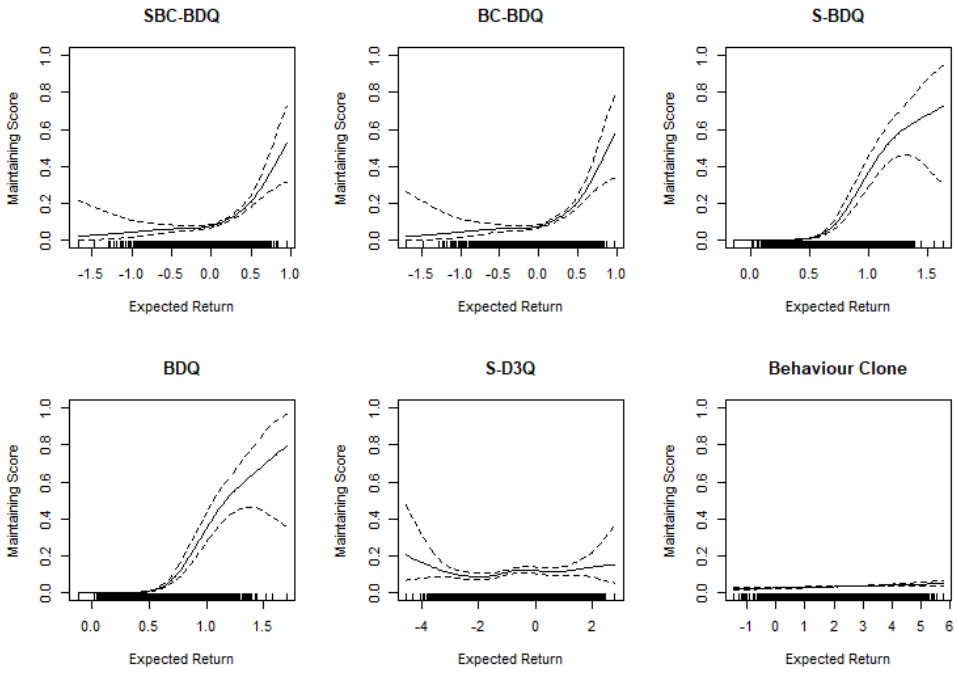


Figure 4.7: Curve of maintenance score vs expected return of the prescribed regimen on the validation set. The expected returns are from well-trained policies.

Inspired by Wang et al. (2018), we used the Jaccard distance to measure the distance between the physician's choice and the regimen recommended by different policies. The mean Jaccard distance is defined as $\frac{1}{N}\sum_{i=1}^{N}\frac{|A_i \bigcup B_i| - |A_i \bigcap B_i|}{|A_i \bigcup B_i|}$, where N is the number of transitions and A_i and B_i are the regimens from the physician and the policy, respectively.

In addition, we calculated the correlation between the maintenance score and the number of drug classes in the validation set. Figure 4.8 shows that when a drug class was added into the regimen, the maintenance score increased by about 0.025. To further guarantee the safety of the recommended regimen, the average number of drug classes recommended by policies was also calculated as an evaluation metric.

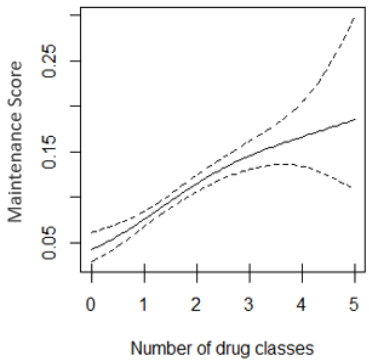


Figure 4.8: Correlation between maintenance score and number of drug classes in the validation set.

4.5.4 Methods Considered

All the methods we considered in the experiments are as follows.

Supervised Batch-constrained BDQ (SBC-BDQ): SBC-BDQ, the proposed method, is introduced in Section 4.4. The hyper-parameters are listed in Table 4.1.

Table 4.1: Hyper-parameters used in SBC-BDQ

Hyper-parameter	Value
Network optimizer	Adam
Learning rate of SBC-BDQ	0.001
Learning rate decay rate of SBC-BDQ	0.999
Delay of decay	100
Soft updating rate of primary network	0.001
Learning rate of generative model	0.05
Discount factor	0.1
Batch size	64
Number of episodes	4000
Supervised learning weight	0.5
Hidden layer size	10

Supervised BDQ (S-BDQ): S-BDQ is a simplified version of SBC-BDQ in which the batch-constraint is removed. The only difference between SBC-BDQ and S-BDQ is the a' selection step. The selection in S-BDQ is based on following criterion:

$$a = (a'_1, ..., a'_K), \text{where } a'_d = \arg\max Q^d_\theta(s', a'_d). \tag{4.7}$$

Batch-contrained BDQ (BC-BDQ): BC-BDQ is a simplified version of SBC-BDQ in which supervision from physicians is removed. BC-BDQ can be easily implemented by setting weight $\epsilon = 0$.

Branching Dueling Double DQN (BDQ) (Tavakoli et al., 2017): BDQ is a branching variant of D3N. Compared to S-BDQ, it has supervised learning weight $\epsilon = 0$, and the a' selection is based on (4.7).

Supervised Dueling Double DQN (S-D3Q): S-D3Q is a supervised RL method that combines SL and dueling DQN. There were 32 different drug combinations composed of 5 drug classes, so the number of neurals in the output layer was 32, and the number of neurals in the hidden layer was also 32. Note that the action space ($|\mathcal{A}| = 32$) was larger than the state space ($|\mathcal{S}| = 26$), so this method might be less efficient than S-BDQ.

Behavior Clone: The BC method was implemented as a multi-label classification model with 5 neurals in the output layer. Compared to the methods that combine SL and RL, the BC method has SL weight $\epsilon = 1$.

4.5.5 Results

Table 4.2 shows the estimated maintenance score, estimated following SBP, Jaccard distance, and number of drug classes in the recommended regimen for all the adopted methods in the DTR recommendation for the testing set. The testing set has an average maintenance score 0.11, an average following SBP of 138.27 mmHg, and the average number of drug classes is 2.02. The number of drug classes from the method that performed the best is 2.25. Even though the policy-recommended regimen has slightly more drug classes, the recommended regimen helps to notably improve the maintenance score.

Comparing the results of S-BDQ and S-D3Q, we can see that S-D3Q has a lower maintenance score. This is due to the lack of efficiency of S-D3Q, which is consistent with the

experimental results in (Tavakoli et al., 2017). We then compared SBC-BDQ with the simplified version BC-BDQ. The smaller Jaccard distance and the better performance of SBC-BDQ indicates that knowledgeable supervision guarantees better performance of the learned policy. Finally, our proposed method, SBC-BDQ, had better performance than the simplified S-BDQ. As shown in Figure 4.9(A), the batch-constrained method tends to choose a' much closer to a_{G_ω}, especially at the beginning of learning.

In conclusion, our proposed method outperforms all the adopted baselines. The reasons are: 1) SBC-BDQ uses batch-constraint to avoid extrapolation error (compared to S-BDQ); 2) S-BDQ uses a branching structure to reduce the dimension of the action space (compared to S-D3Q); 3) SBC-BDQ considers the prescriptions of physicians as supervision information to learn a safe and robust policy (compared to BC-BDQ); and 4) SBC-BDQ regards the treatment recommendation as a sequential decision process, reflecting the clinical practice and using RL to optimize the long-term reward (compared to BC).

Table 4.2: Performance comparison on testing set for regimen recommendation. Thiazide diuretic is the first-line drug, ACEI, ARB, and CCB are the second-line drugs, and alpha/beta-blocker is the third-line drug.

	Estimated Maintenance Score	Estimated Following SBP (mmHg)	Jaccard Distance	First-line Drugs	Second-line Drugs	Third-line Drugs	All Drugs
SBC-BDQ	0.360	128.90	0.479	0.57	1.23	0.45	2.25
S-BDQ	0.275	132.27	0.480	0.57	1.23	0.45	2.25
BC-BDQ	0.146	136.85	0.744	0.28	1.24	0.47	1.99
BDQ	0.146	136.89	0.755	0.28	1.22	0.47	1.97
S-D3Q	0.120	137.96	0.502	0.87	1.35	0.47	2.69
Behavior Clone	0.120	137.70	0.488	0.57	1.16	0.45	2.18

Figure 4.9(C) shows how the observed maintenance score changes with the difference between the learned policy from SBC-BDQ and the doctors' prescriptions from the validation set. We calculated the difference as the hamming distance, i.e. $|A_i - B_i|$, with A_i and B_i as the regimens from the physician and the algorithm, respectively. When the difference is minimal, we obtained the highest maintenance score (0.58). This shows that SBC-BDQ

could learn good policy, and that the doctors' prescription worked well, in general, for patients in the validation set. Figure 4.9(B) shows the effect of the weighting parameter ϵ. SBC-DQN achieved the highest maintenance score and a relatively low Jaccard distance when the value of the weighting parameter was 0.5. This verifies that SBC-BDQ can significantly reduce the estimated maintenance score and recommend safe regimen simultaneously. Figure 4.9(D) shows the effect of the discount factor γ. The evaluation metric (maintenance score) is optimized when $\gamma = 0.1$.

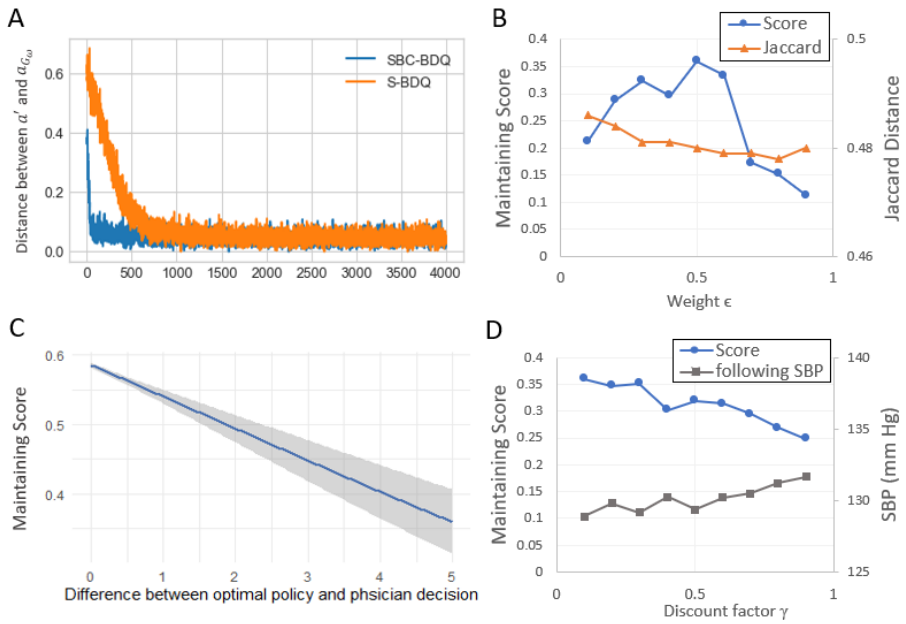


Figure 4.9: (A) Comparison of the distance between a' selected by SBC-BDQ and S-BDQ and a_{G_ω} from the generative model. (B) The effect of ϵ on the testing set. (C) Correlation of the observed maintenance score of the validation set and the difference between the optimal policy and the physicians' decision. (D) The effect of the discount factor γ on the testing set.

Additional simulation was conducted to compare the multi-class classification model and the multi-label classification model. All transitions in the SPRINT data were used. If we only considered the 5 most popular drug classes, there were 32 distinct regimens. When we considered all 10 drug classes, there were a total of 308 distinct regimens. The Jaccard scores from the multi-label classification model and the multi-class classification model were

cross-validated 10-fold, and the results are summarized in Table 4.3. The Jaccard score is defined as $\frac{1}{10}\sum_{j=1}^{10}\frac{1}{N_j}\sum_{i=1}^{N_j}\frac{|A_{ij}\bigcap B_{ij}|}{|A_{ij}\bigcup B_{ij}|}$, where N_j is the number of transitions in j-th fold, and A_{ij} and B_{ij} are the regimens from the physician and the classification models, respectively.

Table 4.3: Simulation result from the comparison of classification models (Jaccard score)

Setup	Multi-label classification	Multi-class classification
All 10 drug classes	0.917	0.903
5 popular drug classes	0.924	0.920

4.5.6 Case Study

Figure 4.10 summarizes the regimens recommended by SBC-DQN and compares the frequencies of the recommended and prescribed drug classes. As the current SBP, baseline CKD condition, baseline CVD condition, race, and gender are all important for regimen recommendation, we compared the regimens using all these factors. In general, we recommend more thiazide-type diuretics, ACEI, and CCB to patients in the testing set. Based on the overall change, for a patient without a baseline CKD condition, we recommend less ARB. For patients with a baseline CVD condition, we recommend more ACEI and CCB, and less ARB. For a patient whose current SBP is greater or equal to 180, we recommend more thiazide-type diuretics, CCB, ARBs, and alpha/beta-blockers, and less ACEIs.

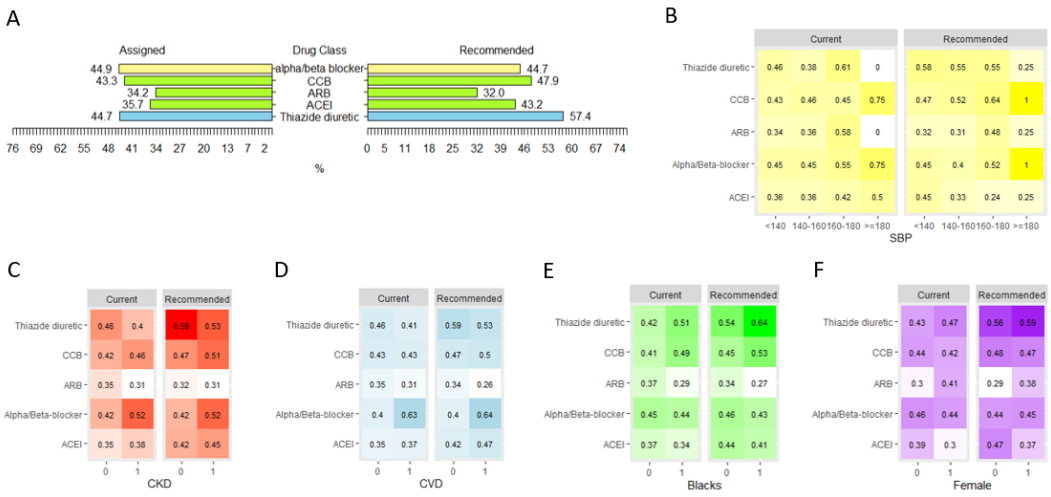


Figure 4.10: Comparison of recommended regimen and prescriptions.

4.6 Discussion

In this chapter, we propose a DTR recommendation system to assist physicians in identifying the personalized optimal DTR for hypertension treatment without resorting to trying all the drugs/drug combinations while guaranteeing medication safety at the same time. We performed comprehensive experiments on the SPRINT data and demonstrated that the proposed system can improve the long-term SBP maintenance score by 24% for patients who might be assigned inappropriate drugs, without drastically increasing the number of drug classes used.

The system is based on the RL algorithm SBC-BDQ. RL can help to identify the optimal action at a particular state by maximizing the expectation of the long-term reward. To solve the challenges in applying RL to hypertension treatment, SBC-BDQ combines SL and RL to ensure safety, uses a branching structure to improve the learning efficiency for a large combinatorial action space, and extends the batch-constraint to avoid extrapolation error in offline RL.

While our results in applying RL to blood pressure control are encouraging, they come with several limitations. First, the SPRINT study had an inclusion criterion of only patients more than 50 years of age. This limits the generalization of our findings to younger patients, even though 37% of hypertension patients are younger than 60 years of age (Fryar et al., 2017). This limitation can be alleviated by combining other data sources with the SPRINT data. Second, a few less common drug classes were not considered in this study for a lack of observations and the intended use of those drugs. These are considered second-line or third-line drugs in the guidelines, and they are usually used for patients with a specific conditions. For example, loop diuretics are more effective than thiazide diuretics in patients with impaired kidney functions. Even with these limitations, our system represents an initial step towards the development of a computer-assisted tool for hypertension drug recommendation. This study focuses on the personalized DTR recommendation of hypertension patients. However, the methods we used to deal with the specific challenges of hypertension can be extended to the treatment of other chronic diseases.

Appendices

Appendix A: Robust Estimation of HTE using Additive Model

A.1. Expressing the existing methods in the general formulation

In Section A.1, we specify the expressions of $c(\mathbf{X}, \mathbf{T})$, $w(\mathbf{X}, \mathbf{T})$, and $g(\mathbf{X})$ for MCM-EA, RL, IPW, and DR methods. We show they satisfy the constraints associated with the general formulation. For most of the methods, the derivations are similar for L_1 and L_2 loss functions. So we show the derivation under the L_2 loss.

(1) MCM-EA. The objective function of L_2-MCM-EA method is

$$L(\tau(\mathbf{x})) = E\left[\frac{(Y_i - \mu(\mathbf{X_i}) - \frac{T_i}{2}\tau(\mathbf{X_i}))^2}{T_i p(\mathbf{X_i}) + (1 - T_i)/2}\middle|\mathbf{X_i} = \mathbf{x}\right].$$

We write

$$w(\mathbf{X_i}, T_i) = \frac{1}{T_i p(\mathbf{X_i}) + (1 - T_i)/2}, \quad c(\mathbf{X_i}, T_i) = \frac{T_i}{2}, \quad g(\mathbf{X_i}) = \mu(\mathbf{X_i}).$$

Then

$$\begin{aligned} &p(\mathbf{x})w(\mathbf{x}, 1)c(\mathbf{x}, 1) + (1 - p(\mathbf{x}))w(\mathbf{x}, -1)c(\mathbf{x}, -1) \\ =&p(\mathbf{x})\frac{1}{p(\mathbf{x})}\frac{1}{2} + (1 - p(\mathbf{x}))\frac{1}{1 - p(\mathbf{x})}\left(-\frac{1}{2}\right) = 0 \\ &c(\mathbf{x}, 1) - c(\mathbf{x}, -1) = \frac{1}{2} - \left(-\frac{1}{2}\right) = 1, \end{aligned}$$

which shows the c and w functions satisfy Conditions C1 and C2. Condition C3 ($w > 0$ and $c \neq 0$) is clearly met. The same set of parameters can be used in L_1 loss. The verification is the same.

(2) R-Learning. The objective function of L_2-based R-learning method is

$$L(\tau(\mathbf{x})) = E\left[(Y_i - \mu(\mathbf{X_i}) - \frac{T_i - 2p(\mathbf{X_i}) + 1}{2}\tau(\mathbf{X_i}))^2 \middle| \mathbf{X_i} = \mathbf{x}\right].$$

We write

$$w(\mathbf{X_i}, T_i) = 1, \quad c(\mathbf{X_i}, T_i) = \frac{T_i - 2p(\mathbf{X_i}) + 1}{2}, \quad g(\mathbf{X_i}) = \mu(\mathbf{X_i}).$$

Then

$$p(\mathbf{x})w(\mathbf{x}, 1)c(\mathbf{x}, 1) + (1 - p(\mathbf{x}))w(\mathbf{x}, -1)c(\mathbf{x}, -1) = p(\mathbf{x})(1 - p(\mathbf{x})) + (1 - p(\mathbf{x}))(-p(\mathbf{x})) = 0$$

$$c(\mathbf{x}, 1) - c(\mathbf{x}, -1) = (1 - p(\mathbf{x})) - (-p(\mathbf{x})) = 1.$$

Therefore, Conditions C1-C3 are met. The same specification works for L_1 loss. The verification of A-learning remains the same.

(3) IPW. The objective function of L_2-based IPW method is

$$L(\tau(\mathbf{x})) = E\left[\left(\frac{T_i + 1}{2p(\mathbf{X_i})} - \frac{1 - T_i}{2(1 - p(\mathbf{X_i}))}Y_i - \tau(\mathbf{X_i})\right)^2 \middle| \mathbf{X_i} = \mathbf{x}\right].$$

We write

$$w(\mathbf{X_i}, T_i) = \left(\frac{T_i + 1}{2p(\mathbf{X_i})} - \frac{1 - T_i}{2(1 - p(\mathbf{X_i}))}\right)^2, \quad c(\mathbf{X_i}, T_i) = \frac{2p(\mathbf{X_i})(1 - p(\mathbf{X_i}))}{T_i - 2p(\mathbf{X_i}) + 1}, \quad g(\mathbf{X_i}) = 0.$$

Then

$$\begin{aligned} &p(\mathbf{x})w(\mathbf{x}, 1)c(\mathbf{x}, 1) + (1 - p(\mathbf{x}))w(\mathbf{x}, -1)c(\mathbf{x}, -1) \\ =&p(\mathbf{x})\frac{1}{p(\mathbf{x})^2}p(\mathbf{x}) + (1 - p(\mathbf{x}))\frac{1}{(1 - p(\mathbf{x}))^2}(p(\mathbf{x}) - 1) = 0 \\ &c(\mathbf{x}, 1) - c(\mathbf{x}, -1) = p(\mathbf{x}) - (p(\mathbf{x}) - 1) = 1. \end{aligned}$$

Therefore, Conditions C1-C3 are met. The same specification works for the L_1 loss function.

(4) DR. The verification of doubly robust method is the same.

A.2. Basic properties of the general formulation

Property 1. Under conditions C1-C3,

$$\tau_0(\mathbf{x}) = argmin_{\tau(\mathbf{x})} E[w(\mathbf{X_i}, T_i)(y - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{x}))^2 | \mathbf{X_i} = \mathbf{x}, T_i = t].$$

Proof of Property 1.

$$\begin{aligned}
L(\tau(\mathbf{x})) =& E[w(\mathbf{X_i}, T_i)(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i}))^2 | \mathbf{X_i} = \mathbf{x}, T_i = t] \\
=& p(\mathbf{x}) E[w(\mathbf{X_i}, T_i)(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i}))^2 | \mathbf{X_i} = \mathbf{x}, T_i = 1] \\
&+ (1 - p(\mathbf{x})) E[w(\mathbf{X_i}, T_i)(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i}))^2 | \mathbf{X_i} = \mathbf{x}, T_i = -1] \\
=& p(\mathbf{x}) w(\mathbf{x}, 1) E[(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i}))^2 | \mathbf{X_i} = \mathbf{x}, T_i = 1] \\
&+ (1 - p(\mathbf{x})) w(\mathbf{x}, -1) E[(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i}))^2 | \mathbf{X_i} = \mathbf{x}, T_i = -1] \\
\frac{\partial L(\tau(\mathbf{x}))}{\partial \tau(\mathbf{x})} =& -2p(\mathbf{x}) w(\mathbf{x}, 1) c(\mathbf{x}, 1)(E[Y_i^{(1)} | \mathbf{X_i} = \mathbf{x}] - g(\mathbf{X_i}) - c(\mathbf{x}, 1)\tau(\mathbf{x})) \\
&- 2(1 - p(\mathbf{x})) w(\mathbf{x}, -1) c(\mathbf{x}, -1)(E[Y_i^{(-1)} | X_i = \mathbf{x}] - g(\mathbf{x}) - c(\mathbf{x}, -1)\tau(\mathbf{x})) \\
=& -2p(\mathbf{x}) w(\mathbf{x}, 1) c(\mathbf{x}, 1)(b_0(\mathbf{x}) + \frac{\tau_0(\mathbf{x})}{2} + E[\varepsilon_i^{(1)} | X_i = \mathbf{x}] - g(\mathbf{X_i}) - c(\mathbf{x}, 1)\tau(\mathbf{x})) \\
&- 2(1 - p(\mathbf{x})) w(\mathbf{x}, -1) c(\mathbf{x}, -1)(b_0(\mathbf{x}) - \frac{\tau_0(\mathbf{x})}{2} + E[\varepsilon_i^{(-1)} | X_i = \mathbf{x}] \\
&- g(\mathbf{x}) - c(\mathbf{x}, -1)\tau(\mathbf{x}))
\end{aligned}$$

Conditions C1-C3 and the conditional independence assumption lead us to

$\tau_0(\mathbf{x}) = argmin_{\tau(x)} L(\tau(\mathbf{x}))$. ■

Property 2. When $c(\mathbf{x}, 1) = 1 - p(\mathbf{x})$, the optimal augmentation function is mean outcome function, i.e., $g_0(\mathbf{x}) = \mu(\mathbf{x})$.

Proof of Property 2. We provide the optimal $g(\cdot)$ in this section, here "optimal" $g(\cdot)$ means the one minimizing the variance of estimator. Let $S(Y_i, \mathbf{X_i}, T_i; \tau(\mathbf{X_i}))$ be the derivative of the objective function $w(\mathbf{X_i}, T_i)(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i}))^2$, with respect to τ. Then the estimating equation is

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} S(Y_i, \mathbf{X_i}, T_i; \tau(\mathbf{X_i})) =& \frac{1}{n}\sum_{i=1}^{n} -2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i})) \\ =& \frac{1}{n}\sum_{i=1}^{n} S_0(Y_i, \mathbf{X_i}, T_i; \tau(\mathbf{X_i})) + 2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)g(\mathbf{X_i}) = 0,\end{aligned}$$

where $S_0(Y_i, \mathbf{X_i}, T_i; \tau(\mathbf{X_i})) = -2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)[Y_i - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i})]$ is the score function without augmentation. By Condition C1, $E[2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)g(\mathbf{X_i})] = 0$, the solution of the augmented score equation always converges to $\tau_0(\cdot)$ in probability. Following Tian et al. (2014); Chen et al. (2017), selecting the optimal $g(\cdot)$ is equivalent to minimizing the conditional variance of

$$S_0(Y_i, \mathbf{X_i}, T_i; \tau_0(\mathbf{X_i})) + 2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)g(\mathbf{X_i})|\mathbf{X_i} = \mathbf{x},$$

where $\tau_0(\mathbf{x})$ is the minimizer of $E[w(\mathbf{X_i}, T_i)(Y_i - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i}))^2|\mathbf{X_i} = \mathbf{x}]$. Noting that

$$\begin{aligned}&E[\{S_0(Y_i, \mathbf{X_i}, T_i; \tau_0(\mathbf{X_i})) + 2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)g(\mathbf{X_i})\}^2|\mathbf{X_i} = \mathbf{x}] \\ =& E[\{S_0(Y_i, \mathbf{X_i}, T_i; \tau_0(\mathbf{X_i})) + 2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)g_0(\mathbf{X_i})\}^2|\mathbf{X_i} = \mathbf{x}] \\ &+ E[\{2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)(g_0(\mathbf{X_i}) - g(\mathbf{X_i}))\}^2|\mathbf{X_i} = \mathbf{x}] \\ \geq& E[\{S_0(Y_i, \mathbf{X_i}, T_i; \tau_0(\mathbf{X_i})) + 2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)g_0(\mathbf{X_i})\}^2|\mathbf{X_i} = \mathbf{x}],\end{aligned}$$

where $g_0(\mathbf{x}) = (1 - p(\mathbf{x}))E[Y_i^{(1)} - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i})|\mathbf{X_i} = \mathbf{x}, T_i = 1]$
$+ p(\mathbf{x})E[Y_i^{(-1)} - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i})|\mathbf{X_i} = \mathbf{x}, T_i = -1]$, which satisfies the equation
$E[\{S_0(Y_i, \mathbf{X_i}, T_i; \tau_0(\mathbf{X_i})) + 2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)g_0(\mathbf{X_i})\}2w(\mathbf{X_i}, T_i)c(\mathbf{X_i}, T_i)\eta(\mathbf{X_i})|\mathbf{X_i} = \mathbf{x}] = 0$

for any function $\eta(\cdot)$. By interaction model (1) and Condition C2, the expression of $g_0(x)$ can be further simplified to $g_0(\mathbf{x}) = \mu(\mathbf{x}) + [1 - p(\mathbf{x}) - c(\mathbf{x}, 1)]\tau_0(\mathbf{x})$. As $\tau_0(\cdot)$ is the unknown target, when $c(\mathbf{x}, 1) = 1 - p(\mathbf{x})$, the optimal augmentation function is mean outcome function, i.e., $g_0(\mathbf{x}) = \mu(\mathbf{x})$. ■

Property 3. Under Conditions C1-C3,

$$\tau_0(\mathbf{x}) = argmin_{\tau(\mathbf{x})} E[w(\mathbf{X_i}, T_i)|y - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{x}))||\mathbf{X_i} = \mathbf{x}, T_i = t].$$

Proof of Property 3.

$$\begin{aligned}
L(\tau(\mathbf{x})) =& E[w(\mathbf{X_i}, T_i)|Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i})||\mathbf{X_i} = \mathbf{x}, T_i = t] \\
=& p(\mathbf{x})E[w(\mathbf{X_i}, T_i)|Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i})||\mathbf{X_i} = \mathbf{x}, T_i = 1] \\
&+ (1 - p(\mathbf{x}))E[w(\mathbf{X_i}, T_i)|Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i})||\mathbf{X_i} = \mathbf{x}, T_i = -1] \\
=& p(\mathbf{x})w(\mathbf{x}, 1)E[|Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i})||\mathbf{X_i} = \mathbf{x}, T_i = 1] \\
&+ (1 - p(\mathbf{x}))w(\mathbf{x}, -1)E[|Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i}, T_i)\tau(\mathbf{X_i})||\mathbf{X_i} = \mathbf{x}, T_i = -1]
\end{aligned}$$

$$
\begin{aligned}
&\frac{\partial L(\tau(\mathbf{x}))}{\partial \tau(\mathbf{x})} \\
=& -p(\mathbf{x})w(\mathbf{x},1)c(\mathbf{x},1)E[sgn(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i},T_i)\tau(\mathbf{X_i}))|\mathbf{X_i}=\mathbf{x},T_i=1] \\
&-(1-p(\mathbf{x}))w(\mathbf{x},-1)c(\mathbf{x},-1)E[sgn(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i},T_i)\tau(\mathbf{X_i}))|\mathbf{X_i}=\mathbf{x},T_i=-1] \\
=& -p(\mathbf{x})w(\mathbf{x},1)c(\mathbf{x},1)E[1-2I(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i},T_i)\tau(\mathbf{X_i}))|\mathbf{X_i}=\mathbf{x},T_i=1] \\
&-(1-p(\mathbf{x}))w(\mathbf{x},-1)c(\mathbf{x},-1)E[1-2I(Y_i - g(\mathbf{X_i}) - c(\mathbf{X_i},T_i)\tau(\mathbf{X_i}))|\mathbf{X_i}=\mathbf{x},T_i=-1] \\
=& -p(\mathbf{x})w(\mathbf{x},1)c(\mathbf{x},1)(1-2P(Y_i^{(1)} < g(\mathbf{X_i}) + c(\mathbf{X_i},T_i)\tau(\mathbf{X_i})|\mathbf{X_i}=\mathbf{x},T_i=1)) \\
&-(1-p(\mathbf{x}))w(\mathbf{x},-1)c(\mathbf{x},-1)(1-2P(Y_i^{(-1)} < g(\mathbf{X_i}) + c(\mathbf{X_i},T_i)\tau(\mathbf{X_i})|\mathbf{X_i}=\mathbf{x},T_i=-1)) \\
=& -p(\mathbf{x})w(\mathbf{x},1)c(\mathbf{x},1)(1-2F_{Y_i^{(1)}}(g(\mathbf{X_i}) + c(\mathbf{X_i},T_i)\tau(\mathbf{X_i})|\mathbf{X_i}=\mathbf{x},T_i=1)) \\
&-(1-p(\mathbf{x}))w(\mathbf{x},-1)c(\mathbf{x},-1)(1-2F_{Y_i^{(-1)}}(g(\mathbf{X_i}) + c(\mathbf{X_i},T_i)\tau(\mathbf{X_i})|\mathbf{X_i}=\mathbf{x},T_i=-1))
\end{aligned}
$$

By Condition C1, the score equation can be written to

$$F_{Y_i^{(1)}}(g(\mathbf{x}) + c(\mathbf{x},1)\tau(\mathbf{x})) - F_{Y_i^{(-1)}}(g(\mathbf{x}) + c(\mathbf{x},-1)\tau(\mathbf{x})) = 0.$$

Let $F_{Y_i^{(1)}}(g(\mathbf{x}) + c(\mathbf{x},1)\hat{\tau}(\mathbf{x})) = F_{Y_i^{(-1)}}(g(\mathbf{x}) + c(\mathbf{x},-1)\hat{\tau}(\mathbf{x})) = q$, where $q \in (0,1)$, then

$$g(\mathbf{x}) + c(\mathbf{x},1)\hat{\tau}(\mathbf{x}) = Q_q(Y_i^{(1)}|\mathbf{X_i}=\mathbf{x})$$

$$g(\mathbf{x}) + c(\mathbf{x},-1)\hat{\tau}(\mathbf{x}) = Q_q(Y_i^{(-1)}|\mathbf{X_i}=\mathbf{x}).$$

By Condition C2 ($c(\mathbf{x},1) - c(\mathbf{x},-1) = 1$), we have

$$\hat{\tau}(\mathbf{x}) = Q_q(Y_i^{(1)}|\mathbf{X_i}=\mathbf{x}) - Q_q(Y_i^{(-1)}|\mathbf{X_i}=\mathbf{x}).$$

As

$$
\begin{aligned}
&Q_q(Y_i^{(1)}|\mathbf{X_i}=\mathbf{x}) - Q_q(Y_i^{(-1)}|\mathbf{X_i}=\mathbf{x}) \\
=&Q_q(Y_i|\mathbf{X_i}=\mathbf{x}, T_i=1) - Q_q(Y_i|\mathbf{X_i}=\mathbf{x}, T_i=-1) \\
=&Q_q(b_0(\mathbf{X_i}) + \frac{\tau_0(\mathbf{X_i})}{2} + \varepsilon_i|\mathbf{X_i}=\mathbf{x}, T_i=1) - Q_q(b_0(\mathbf{X_i}) - \frac{\tau_0(\mathbf{X_i})}{2} + \varepsilon_i|\mathbf{X_i}=\mathbf{x}, T_i=-1) \\
=&b_0(x) + \frac{\tau_0(x)}{2} + Q_q(\varepsilon_i|\mathbf{X_i}=\mathbf{x}, T_i=1) - b_0(x) + \frac{\tau_0(x)}{2} - Q_q(\varepsilon_i|\mathbf{X_i}=\mathbf{x}, T_i=-1) \\
=&\tau_0(x) + Q_q(\varepsilon_i|\mathbf{X_i}=\mathbf{x}, T_i=1) - Q_q(\varepsilon_i|\mathbf{X_i}=\mathbf{x}, T_i=-1).
\end{aligned}
$$

By Assumption 3, $\tau_0(\mathbf{x}) = argmin_{\tau(x)} L(\tau(\mathbf{x}))$. ■

A.3. Asymptotic Properties

To prove Theorem 1, we introduce two lemma.

Lemma 1. Under the same assumptions as Theorem 1, W_n is asymptotically equivalent to the $(K+p)$ dimensional normal with mean 0 and variance G.

Proof of Lemma 1. Let $Z_n = -\sqrt{\frac{K_n}{n}} \sum_{i=1}^{n} w(X_i, T_i) c(X_i, T_i) B(X_i)^T \delta \rho'(U_i)$, the conditional expectation of $w(X_i, T_i) c(X_i, T_i) \rho'(U_i)$ with respect to X_i is as follows. First, we calculate the conditional expectation with respect of X_i and T_i,

$$
\begin{aligned}
&E[w(X_i, T_i) c(X_i, T_i) \rho'(U_i)|X_i = x_i] \\
=&E[w(X_i, T_i) c(X_i, T_i) \rho'(Y_i - g(X_i) - c(X_i, T_i) B(X_i)^T \beta^*)|X_i = x_i] \\
=&p(x_i) w(x_i, 1) c(x_i, 1) E[\rho'(Y_i^{(1)} - g(X_i) - c(X_i, 1) B(X_i)^T \beta^*)|X_i = x_i, T_i = 1] \\
&+ (1 - p(x_i)) w(x_i, -1) c(x_i, -1) \times \\
&E[\rho'(Y_i^{(-1)} - g(X_i) - c(X_i, -1) B(X_i)^T \beta^*)|X_i = x_i, T_i = -1] \qquad (4.1) \\
=&p(x_i) w(x_i, 1) c(x_i, 1) \Big(E[\rho'(Y_i^{(1)} - g(X_i) - c(X_i, 1) B(X_i)^T \beta^*) \\
&- \rho'(Y_i^{(-1)} - g(X_i) - c(X_i, -1) B(X_i)^T \beta^*)|X_i = x_i] \Big). \qquad (4.2)
\end{aligned}
$$

From (6.1) to (6.2) is based on Condition C1. Then, based on the interaction model and the distance between $\tau_0(x)$ and $B(x)^T\beta^*$, we have

$$
\begin{aligned}
&E[w(X_i,T_i)c(X_i,T_i)\rho'(U_i)|X_i=x_i]\\
=&p(x_i)w(x_i,1)c(x_i,1)\times\\
&\Big\{E[\rho'(b(X_i)+\frac{1}{2}\tau_0(X_i)+\varepsilon_i^{(1)}-g(X_i)\\
&-c(X_i,1)\tau_0(X_i)-c(X_i,1)b^a(X_i)[1+\tilde{o}(1)])\\
&-\rho'(b(X_i)-\frac{1}{2}\tau_0(X_i)+\varepsilon_i^{(-1)}-g(X_i)-c(X_i,-1)\tau_0(X_i)-c(X_i,-1)b^a(X_i)[1+\tilde{o}(1)])\\
&|X_i=x_i]\Big\} \qquad (4.3)\\
=&p(x_i)w(x_i,1)c(x_i,1)\times\\
&\Big\{E[\rho'(b(X_i)-g(X_i)-[c(X_i,1)-0.5]\tau_0(X_i)+\varepsilon_i^{(1)}-c(X_i,1)b_a(X_i)[1+\tilde{o}(1)])\\
&-\rho'(b(X_i)-g(X_i)-[c(X_i,1)-0.5]\tau_0(X_i)+\varepsilon_i^{(-1)}\\
&-c(X_i,-1)b_a(X_i)[1+\tilde{o}(1)])|X_i=x_i]\Big\}, \qquad (4.4)
\end{aligned}
$$

where $\tilde{o}(1)$ uniformly holds for all x by the distance between $\tau_0(x)$ and $B(x)^T\beta^*$. From (6.3) to (6.4) is based on Condition C2. Let $\varphi(X_i,T_i)=b(X_i)-g(X_i)-[c(X_i,T_i)-0.5]\tau_0(X_i)+\varepsilon_i^{(T_i)}$, the expectation condition of $w(X_i,T_i)c(X_i,T_i)\rho'(U_i)$ on X_i is

$$
\begin{aligned}
& p(x_i)w(x_i,1)c(x_i,1) \\
& \times \Big\{ E[\rho'(\varphi(X_i,T_i))|X_i=x_i,T_i=1] - E[\rho'(\varphi(X_i,T_i))|X_i=x_i,T_i=-1] \\
& - E[\rho''(\varphi(X_i,T_i) - \alpha^{(1)}c(X_i,1)b_a(X_i)[1+\tilde{o}(1)])c(X_i,1)b_a(X_i)[1+\tilde{o}(1)]|X_i=x_i] \\
& + E[\rho''(\varphi(X_i,T_i) - \alpha^{(-1)}c(X_i,-1)b_a(X_i)[1+\tilde{o}(1)])c(X_i,-1)b_a(X_i)[1+\tilde{o}(1)]|X_i=x_i] \Big\}
\end{aligned} \tag{4.5}
$$

$$
\begin{aligned}
=& p(x_i)w(x_i,1)c(x_i,1)\times \\
& \Big\{ - E[\rho''(\varphi(X_i,T_i) - \alpha^{(1)}c(X_i,1)b_a(X_i)[1+\tilde{o}(1)])c(X_i,1)b_a(X_i)[1+\tilde{o}(1)]|X_i=x_i] \\
& + E[\rho''(\varphi(X_i,T_i) - \alpha^{(-1)}c(X_i,-1)b_a(X_i)[1+\tilde{o}(1)])c(X_i,-1)b_a(X_i)[1+\tilde{o}(1)]|X_i=x_i] \Big\},
\end{aligned}
$$

where $\alpha^{(1)}, \alpha^{(-1)} \in (0,1)$ are from Taylor expansion. The first two terms in (6.5) inside the brace are cancelled out based on the conditional independence error assumption (Assumption 3). Finally, by the definition of Φ, we have the conditional expectation equals

$$
\begin{aligned}
& p(x_i)w(x_i,1)c(x_i,1)b_a(x_i)\times \\
& \Big\{ - \Phi''([1-\alpha^{(1)}]c(X_i,1)b_a(X_i)[1+\tilde{o}(1)]|X_i,T_i=1)c(x_i,1)) \\
& + \Phi''([1-\alpha^{(-1)}]c(X_i,-1)b_a(X_i)[1+\tilde{o}(1)]|X_i,T_i=-1)c(x_i,-1)) \Big\}[1+\tilde{o}(1)] \\
=& o(1).
\end{aligned} \tag{4.6}
$$

As the order of $b^a(x)$ is $o(K_n^{-(q+1)})$, Assumption 2 (positivity assumption), and Conditions C4 and C9, the conditional expectation is of $o(1)$. This means the conditional expectation of the score function with loss functions satisfy Conditions C7-C11 goes to zero.

Therefore, let $\psi(X_i, T_i, U_i) = w(X_i, T_i)c(X_i, T_i)\rho'(U_i)$, we obtain

$$
\begin{aligned}
& E\left[\left|\sqrt{\frac{K_n}{n}}B(X_i)^T\delta\left[\psi(X_i,T_i,U_i) - E[\psi(X_i,T_i,U_i)|X_i = x_i]\right]\right|^{2+\gamma}\Big|X_i = x_i\right] \\
=& \left(\frac{K_n}{n}\right)^{\frac{2+\gamma}{2}}|B(x_i)^T\delta|^{2+\gamma}E\left[|\psi(X_i,T_i,U_i)|^{2+\gamma} + o(1)\Big|X_i = x_i\right] \\
=& \left(\frac{K_n}{n}\right)^{\frac{2+\gamma}{2}}|B(x_i)^T\delta|^{2+\gamma}\{p(x_i)E[|w(x_i,1)c(x_i,1)\rho'(U_i)|^{2+\gamma} + o(1)|X_i = x_i, T_i = 1] \\
& + (1-p(x_i))E[|w(x_i,-1)c(x_i,-1)\rho'(U_i)|^{2+\gamma} + o(1)|X_i = x_i, T_i = -1]\} \\
\leq& O\left(\left(\frac{K_n}{n}\right)^{\frac{2+\gamma}{2}}\right),
\end{aligned}
$$

where the last two steps are derived by Condition C11. The conditional variance of Z_n respect to X_n can be calculated as following.

$$
\begin{aligned}
V[Z_n|X^{(n)}] &= \frac{K_n}{n}\sum_{i=1}^{n}\{B(x_i)^T\delta\}^2 V\left[w(X_i,T_i)c(X_i,T_i)\rho'(U_i)\Big|X_i = x_i\right] \\
&= \frac{K_n}{n}\sum_{i=1}^{n}\{B(x_i)^T\delta\}^2\Bigg\{E\left[\left(w(X_i,T_i)c(X_i,T_i)\rho'(U_i)\right)^2\Big|X_i = x_i\right] \\
&\quad - E\left[w(X_i,T_i)c(X_i,T_i)\rho'(U_i)\Big|X_i = x_i\right]^2\Bigg\} \qquad (4.7) \\
&= K_n\delta^T G\delta(1 + o_P(1)) \qquad (4.8) \\
&= O(K_n),
\end{aligned}
$$

where G is the variance of W_n. Here the derivation from (6.7) to (6.8) uses the Condition C8. Because the matrix G is positive definite and has a finite maximum eigenvalue for any bounded function (Lemma 6.2 of Zhou et al. (1998)), there exists the constants d_1 and d_2 such that

$$d_1 \leq \delta^T G\delta \leq d_2.$$

So it follows that

$$\begin{aligned}\frac{1}{V[Z_n|X_n]^{(2+\gamma)/2}}\sum_{i=1}^{n}E\left[\left|\sqrt{\frac{K_n}{n}}B(X_i)^T\delta\{\psi(X_i,T_i,U_i)-E[\psi(X_i,T_i,U_i)|X_i]\}\right|^{2+\gamma}\Big|X_i\right]\\ \leq O(K_n^{-(2+\gamma)/2})O\left(n\left(\frac{K_n}{n}\right)^{(2+\gamma)/2}\right)\\ =o(1)\end{aligned}$$

since $\gamma \geq 0$. This leads to

$$\frac{Z_n - E[Z_n|X^{(n)}]}{\sqrt{V[Z_n|X^{(n)}]}} \xrightarrow{D} N(0,1)$$

from Lyapunov's Theorem. The conditional expectation of Z_n respect to $X^{(n)}$ can be calculated as

$$\begin{aligned}E[Z_n|X^{(n)}] &= -\sqrt{\frac{K_n}{n}}\sum_{i=1}^{n}B(x_i)^T\delta E[\psi(X_i,T_i,U_i)|X_i=x_i]\\ &= -\sqrt{nK_n}\frac{1}{n}\sum_{i=1}^{n}p(x_i)w(x_i,1)c(x_i,1)b_a(x_i)B(x_i)^T\delta\times\\ &\Big\{-\Phi''\big([1-\alpha^{(T_i)}]c(X_i,T_i)b_a(X_i)[1+\tilde{o}(1)]|X_i=x_i,T_i=1\big)c(x_i,1)\\ &+\Phi''\big([1-\alpha^{(T_i)}]c(X_i,T_i)b_a(X_i)\\ &\times[1+\tilde{o}(1)]|X_i=x_i,T_i=-1\big)c(x_i,-1)\Big\}[1+\tilde{o}(1)]\\ &= -\sqrt{nK_n}\int_0^1 p(x)w(x,1)c(x,1)b_a(x)B(x)^T\delta\times\\ &\Big\{-\Phi''\big([1-\alpha^{(T_i)}]c(X_i,T_i)b_a(X_i)[1+\tilde{o}(1)]|X_i=x,T_i=1\big)c(x,1)\\ &+\Phi''\big([1-\alpha^{(T_i)}]c(X_i,T_i)b_a(X_i)[1+\tilde{o}(1)]|X_i=x,T_i=-1\big)c(x,-1)\Big\}dQ(x)\\ &\times[1+\tilde{o}(1)]\\ &= o\left(\sqrt{nK_n}K_n^{-(q+2)}\right).\end{aligned}$$

The last step is from the proof of Lemma 6.10 of Agarwal and Studden (1980) and equation (6), for $j = -p+1, ..., K_n$, we have

$$\begin{aligned}
&\int_0^1 p(x)w(x,1)c(x,1)c(x,t)b_a(x)B_j(x)^T\delta\times \\
&\Phi''([1-\alpha^{(T)}]c(X,T)b_a(X)[1+\tilde{o}(1)]|X=x,T=t)dQ(x)[1+\tilde{o}(1)] \\
=&o(K_n^{-(q+2)}),
\end{aligned}$$

by which $\sqrt{nK_n}o(K_n^{-(q+2)}) = o(1)$ from the order of K_n in Theorem 1. Consequently, we have $E[Z_n|X^{(n)}]/\sqrt{V[Z_n|X^{(n)}]} = o_P(1)$ and Lemma 1 holds. ■

Lemma 2. Let ν be a continuous function on the interval $[0,1]$, then $D(\nu) = O(K_n^{-1})$. Furthermore, $D(\nu)^{-1} = O(K_n)$.

Proof of Lemma 2. The (i,j)-component of $D(\nu)$ is

$$d_{ij} = \int_0^1 \nu(x)B_i(x)B_j(x)dQ(x).$$

From the fundamental property of B-spline function (Lemma 6.1 in Zhou et al. (1998)), we have

$$|g_{ij}(\nu)| \leq sup_{x\in[0,1]}\{|\nu(x)|\}sup_{x\in[0,1]}\{|Q(x)|\}max_{i,j}\int_0^1 B_i(x)B_j(x)dx = O(K_n^{-1}).$$

From the property of B-spline function (De Boor et al., 1978), $D(\nu)$ is positive definite matrix. Therefore, $G(\nu)^{-1} = O(K_n)$ is satisfied. ■

Now, we are ready to prove Theorem 1. For simplicity we write $a_n \overset{as}{\sim} b_n$, where random sequence $\{a_n\}$ and $\{b_n\}$, if $a_n/b_n = O_P(1)$.

Proof of Theorem 1. The objective function of proposed method is

$$\begin{aligned} L_n(\beta) &= \sum_{i=1}^{n} w(X_i, T_i)\rho\left(Y_i - g(X_i) - c(X_i, T_i)B(X_i)^T\beta\right) \\ &= \sum_{i=1}^{n} w(X_i, T_i)\rho\left(Y_i - g(X_i) - c(X_i, T_i)B(X_i)^T[\beta - \beta^* + \beta^*]\right) \\ &= \sum_{i=1}^{n} w(X_i, T_i)\rho\left(Y_i - g(X_i) - c(X_i, T_i)B(X_i)^T\beta^* - c(X_i, T_i)B(X_i)^T[\beta - \beta^*]\right). \end{aligned}$$

As this minimization problem doesn't have explicit solution, for the convergence of $\sqrt{a_n}(\hat{\beta} - \beta^*)$, we modify the objective function $L_n(\beta)$ as follows:

$$U_n(\delta) = \sum_{i=1}^{n}\left[w(X_i, T_i)\left(\rho\left(U_i - \sqrt{\frac{K_n}{n}}c(X_i, T_i)B(X_i)^T\delta\right) - \rho\left(U_i\right)\right)\right],$$

where $U_i = Y_i - g(X_i) - c(X_i, T_i)B(X_i)^T\beta^*$. Then the minimizer $\hat{\delta}_n$ of $U_n(\delta)$ can be obtained as

$$\hat{\delta}_n = \sqrt{\frac{n}{K_n}}(\hat{\beta} - \beta^*).$$

Define $R_n(\delta) =$

$U_n(\delta) - E[U_n(\delta|X^{(n)}, T_n)] - \sum_{i=1}^{n} w(X_i, T_i)\{\rho'(U_i) - E[\rho'(U_i)|X_i, T_i]\}\{\alpha_n c(X_i, T_i)B(X_i)^T\delta\}$, where $X^{(n)}$ represents all the observed X. We have $E[R_n(\delta)|X^{(n)}, T_n] = 0$ from the straight calculation. Let

$$r_i = w(X_i, T_i)\left[\rho\left(U_i - \alpha_n c(X_i, T_i)B(X_i)^T\delta\right) - \rho(U_i) - \rho'(U_i)\{\alpha_n c(X_i, T_i)B(X_i)^T\delta\}\right].$$

Then by Condition C9-10 with $s = \alpha_n c(X_i, T_i) B(X_i)^T \delta$, the variance of r_i is

$$\begin{aligned} V[r_i] =& E\left[(w(X_i, T_i) \left\{ \rho\left(U_i - \alpha_n c(X_i, T_i) B(X_i)^T \delta\right) - \rho(U_i) \right. \right. \\ & - \rho'(U_i) \{\alpha_n c(X_i, T_i) B(X_i)^T \delta\}^2 \\ & - \left[w(X_i, T_i) \{ \Phi(\alpha_n c(X_i, T_i) B(X_i)^T \delta | X_i, T_i) \right. \\ & \left. \left. \left. - \Phi(0|X_i, T_i) - \Phi'(0|X_i, T_i) \alpha_n c(X_i, T_i) B(X_i)^T \delta \} \right]^2 \right] \\ =& o(\alpha_n^2). \end{aligned}$$

Therefore, we have from $K_n = O(n^{1/(2q+3)})$, $E[R_n(\delta)^2] = \frac{1}{n} V[r_1] = o(1)$ and $R_n(\delta) = o_P(1)$. By the definition of $\Phi(t|X, T)$, the Taylor expansion of

$$\Phi(\alpha_n c(X_i, T_i) B(X_i)^T \delta | X_i, T_i)$$

around $\alpha_n = 0$, we have $E[\rho(U_i - \alpha_n c(X_i, T_i) B(X_i)^T \delta) | X_i, T_i] =$ $\Phi(\alpha_n c(X_i, T_i) B(X_i)^T \delta | X_i, T_i)$ and

$$\begin{aligned} & \Phi(\alpha_n c(X_i, T_i) B(X_i)^T \delta | X_i, T_i) \\ =& \Phi(0|X_i, T_i) + \Phi'(0|X_i, T_i) \alpha_n c(X_i, T_i) B(X_i)^T \delta \\ & + \frac{1}{2} \Phi''(0|X_i, T_i) \{\alpha_n c(X_i, T_i) B(X_i)^T \delta\}^2 + o(\alpha_n^2). \end{aligned}$$

Therefore, the conditional expectation of $U_n(\delta)$ given X_n can be written as

$$\begin{aligned} & E[U_n(\delta) | X^{(n)}, T_n] \\ =& \sum_{i=1}^n w(X_i, T_i) \left[\Phi'(0|X_i, T_i) \alpha_n c(X_i, T_i) B(X_i)^T \delta \right. \\ & \left. + \frac{1}{2} \sum_{i=1}^n \Phi''(0|X_i, T_i) \{\alpha_n c(X_i, T_i) B(X_i)^T \delta\}^2 \right] + o(\alpha_n^2). \end{aligned}$$

Thus, we have $U_n(\delta)$ as

$$\begin{aligned} U_n(\delta) &= E[U_n(\delta|X^{(n)}, T_n)] \\ &\quad + \sum_{i=1}^{n} w(X_i, T_i)\{\rho'(U_i) - E[\rho'(U_i)|X_i, T_i]\}\{\alpha_n c(X_i, T_i) B(X_i)^T \delta\} + o_P(1) \\ &= -\sqrt{K_n} W_n^T \delta + \frac{K_n}{2} \delta^T G_n \delta + o_P(1), \end{aligned}$$

where

$$\begin{aligned} W_n &= -\sqrt{\frac{1}{n}} \sum_{i=1}^{n} \rho'(U_i) w(X_i, T_i) c(X_i, T_i) B(X_i) \\ G_n &= \frac{1}{n} \sum_{i=1}^{n} \Phi''(0|X_i, T_i) w(X_i, T_i) c(X_i, T_i)^2 B(X_i)^T B(X_i). \end{aligned}$$

The minimizer of $U_n(\delta)$ is

$$\hat{\delta} = argmin_\delta\{U_n(\delta)\} = G_n^{-1} \frac{W_n}{\sqrt{K_n}} + o_P(1),$$

which is the solution of $\partial Q_n(\delta)/\partial\delta = 0$. Hence, because $\hat{\delta} = \frac{1}{\alpha_n}(\hat{\beta} - \beta^*)$, we have

$$\sqrt{\frac{n}{K_n}}(\hat{\tau}(x) - \tau^*(x)) = \sqrt{\frac{n}{K_n}} B(x)^T G_n^{-1} \frac{W_n}{\sqrt{K_n}} + o_P(1).$$

The asymptotic variance of $\hat{\tau}(x)$ is similar to that of $\hat{\tau}(x) - \tau^*(x)$ because W_n is the only random vector in the asymptotic form of $\hat{\tau}(x)$, it is easy to show that

$$V[\hat{\tau}(x)] = \frac{1}{K_n} B(x)^T G_n V[W_n] G_n B(x)(1 + o(1)),$$

where $G_n = D(\nu) + o(K_n^{-1})$ and $\nu(x) = p(x)w(x,1)c(x,1)^2\rho''(y^{(1)} - g(x) - c(x,1)B(x)^T\beta^*) + (1-p(x))w(x,-1)c(x,-1)^2\rho''(y^{(-1)} - g(x) - c(x,-1)B(x)^T\beta^*)$ due to the Riemann integral, fundamental asymptotic property of B-spline basis, and Lemma 2. Under the condition

$K_n = O(n^{1/(2q+3)})$, we have

$$\sqrt{\frac{n}{K_n}}\{\hat{\tau}(x) - \tau_0(x)\} = \sqrt{\frac{n}{K_n}}\{\hat{\tau}(x) - \tau^*(x) + b^a(x) + o(K_n^{-(q+1)})\}$$

and $\sqrt{\frac{n}{K_n}}b^a(x) = O\left(\sqrt{\frac{n}{K_n}}K_n^{-(q+1)}\right) = O(1)$. Thus, we have

$$\sqrt{\frac{n}{K_n}}(\hat{\tau}(x) - \tau_0(x) - b^a(x)) \xrightarrow{D} N(0, \Psi(x)),$$

where $\Psi(x) = lim_{n\to\infty}\frac{1}{K_n}B(x)^T D(\nu)^{-1}GD(\nu)^{-1}B(x)$. This completes the proof. ■

To prove Theorem 2, we introduce two lemmas as well.

Lemma 3. Let $U_i = w(X_i, T_i)(Y_i - g(X_i) - c(X_i, T_i)B(X_i)^T\beta^*)$. Under the same assumptions as Theorem 2,

$$-\sqrt{\frac{K_n}{n}}\sum_{i=1}^{n} w(X_i, T_i)c(X_i, T_i)B(X_i)^T\delta[1 - 2I(U_i < 0)] \overset{as}{\approx} -\sqrt{K_n}W^T\delta,$$

where $W \sim N(0, G)$.

Proof of Lemma 3. Let $Z_n = -\sqrt{\frac{K_n}{n}}\sum_{i=1}^{n} w(X_i, T_i)c(X_i, T_i)B(X_i)^T\delta[1 - 2I(U_i < 0)]$, the conditional expectation of $w(X_i, T_i)c(X_i, T_i)[1-2I(U_i < 0)]$ respect to X_i can be calculated

as following.

$$
\begin{aligned}
&E[w(X_i,T_i)c(X_i,T_i)[1-2I(U_i<0)]|X_i=x_i] \\
=&p(x_i)w(x_i,1)c(x_i,1)E[1-2I(U_i<0)|X_i=x_i,T_i=1] \\
&+(1-p(x_i))w(x_i,-1)c(x_i,-1)E[1-2I(U_i<0)|X_i=x_i,T_i=-1] \\
=&p(x_i)w(x_i,1)c(x_i,1)\{1-2E[I(U_i<0)|X_i=x_i,T_i=1] \\
&-1+2E[I(U_i<0)|X_i=x_i,T_i=-1]\} \\
=&2p(x_i)w(x_i,1)c(x_i,1)[P(U_i<0|X_i=x_i,T_i=-1)-P(U_i<0|X_i=x_i,T_i=1)] \\
=&2p(x_i)w(x_i,1)c(x_i,1) \\
&\times[P(Y_i^{(-1)}<g(x_i)+c(x_i,-1)B(x_i)^T\beta^*)-P(Y_i^{(1)}<g(x_i)+c(x_i,1)B(x_i)^T\beta^*)] \\
=&2p(x_i)w(x_i,1)c(x_i,1) \\
&\times[P(\varepsilon_i^{(-1)}<g(x_i)-b(x_i)+[c(x_i,1)-0.5]\tau_0(x_i)+[c(x_i,1)-1]b^a(x_i)(1+o(1))|x_i) \\
&-P(\varepsilon_i^{(1)}<g(x_i)-b(x_i)+[c(x_i,1)-0.5]\tau_0(x_i)+c(x_i,1)b^a(x_i)(1+o(1))|x_i)] \\
=&-2p(x_i)w(x_i,1)c(x_i,1)b^a(x_i)f_{\varepsilon_i}(g(x_i)-b(x_i)+[c(x_i,1)-0.5]\tau_0(x_i)|x_i)(1+o(1)) \\
=&o(1).
\end{aligned}
$$

The derivation from the first equation to the second equation is by Condition C1, that from the third equation to the fourth equation is by Model (1), that to the fifth equation is by proposed Condition C2, the last two steps are by Taylor expansion and the order of $b^a(x_i)$.

Therefore, let $\psi(X_i, T_i, U_i) = w(X_i, T_i)c(X_i, T_i)[1 - 2I(U_i < 0)]$, we obtain

$$
\begin{aligned}
& E\left[\left|\sqrt{\frac{K_n}{n}} B(X_i)^T \delta \left[\psi(X_i, T_i, U_i) - E[\psi(X_i, T_i, U_i)|X_i = x_i]\right]\right|^{2+\gamma} \Big| X_i = x_i\right] \\
=& \left(\frac{K_n}{n}\right)^{\frac{2+\gamma}{2}} |B(x_i)^T \delta|^{2+\gamma} E\left[|\psi(X_i, T_i, U_i)|^{2+\gamma} + o(1)\Big| X_i = x_i\right] \\
=& \left(\frac{K_n}{n}\right)^{\frac{2+\gamma}{2}} \\
& \times |B(x_i)^T \delta|^{2+\gamma} \{p(x_i) E[|w(x_i, 1)c(x_i, 1)[1 - 2I(U_i < 0)]|^{2+\gamma} + o(1)|X_i = x_i, T_i = 1] \\
& + (1 - p(x_i)) E[|w(x_i, -1)c(x_i, -1)[1 - 2I(U_i < 0)]|^{2+\gamma} + o(1)|X_i = x_i, T_i = -1]\} \\
\leq & O\left(\left(\frac{K_n}{n}\right)^{\frac{2+\gamma}{2}}\right),
\end{aligned}
$$

where the last two steps are derived by Condition C12. The conditional variance of Z_n respect to X_n can be calculated as following.

$$
\begin{aligned}
V[Z_n|X_n] &= \frac{K_n}{n} \sum_{i=1}^{n} \{B(x_i)^T \delta\}^2 V\left[w(X_i, T_i)c(X_i, T_i)[1 - 2I(U_i < 0)]\Big| X_i = x_i\right] \\
&= \frac{K_n}{n} \sum_{i=1}^{n} \{B(x_i)^T \delta\}^2 \Bigg\{ E\left[\left(w(X_i, T_i)c(X_i, T_i)[1 - 2I(U_i < 0)]\right)^2 \Big| X_i = x_i\right] \\
&\quad - E\left[w(X_i, T_i)c(X_i, T_i)[1 - 2I(U_i < 0)]\Big| X_i = x_i\right]^2 \Bigg\} \\
&= \frac{K_n}{n} \sum_{i=1}^{n} \{B(x_i)^T \delta\}^2 \Big\{ p(x_i) w(x_i, 1)^2 c(x_i, 1)^2 + (1 - p(x_i)) w(x_i, -1)^2 c(x_i, -1)^2 \\
&\quad - E\left[w(X_i, T_i)c(X_i, T_i)[1 - 2I(U_i < 0)]\Big| X_i = x_i\right]^2 \Big\} \\
&= \frac{K_n}{n} \sum_{i=1}^{n} \{B(x_i)^T \delta\}^2 \left\{ \frac{p(x_i)}{(1 - p(x_i))} w(x_i, 1)^2 c(x_i, 1)^2 \right\} (1 + o_P(1)) \\
&= K_n \delta^T G \delta (1 + o_P(1)) \\
&= O(K_n).
\end{aligned}
$$

So it follows that

$$\begin{aligned}\frac{1}{V[Z_n|X_n]^{(2+\gamma)/2}}\sum_{i=1}^{n}E\left[\left|\sqrt{\frac{K_n}{n}}B(X_i)^T\delta\{\psi(X_i,T_i,U_i)-E[\psi(X_i,T_i,U_i)|X_i]\}\right|^{2+\gamma}\Big|X_i\right]\\ \leq O(K_n^{-(2+\gamma)/2})O\left(n\left(\frac{K_n}{n}\right)^{(2+\gamma)/2}\right)\\ =o(1)\end{aligned}$$

since $\gamma \geq 0$. This leads to

$$\frac{Z_n - E[Z_n|X_n]}{\sqrt{V[Z_n|X_n]}} \xrightarrow{D} N(0,1)$$

from Lyapunov's Theorem. The conditional expectation of Z_n respect to X_n can be calculated as

$$\begin{aligned}E[Z_n|X_n] &= -\sqrt{\frac{K_n}{n}}\sum_{i=1}^{n}B(X_i)^T\delta E[\psi(X_i,T_i,U_i)|X_i]\\ &= 2\sqrt{\frac{K_n}{n}}\sum_{i=1}^{n}p(x_i)w(x_i,1)c(x_i,1)b^a(x_i)\\ &\times f_{\varepsilon_i}(g(x_i)-b(x_i)+[c(x_i,1)-0.5]\tau_0(x_i))(1+o(1))\\ &= 2\sqrt{nK_n}\int_0^1 p(u)w(u,1)c(u,1)b^a(u)\\ &\times f_{\varepsilon_i}(g(u)-b(u)+[c(u,1)-0.5]\tau_0(u)|u)dQ(u)(1+o(1)).\end{aligned}$$

From the proof of Lemma 6.10 of Agarwal and Studden (1980), for $j = -p+1, ..., K_n$, we have

$$\begin{aligned}&\int_0^1 p(u)w(u,1)c(u,1)b^a(u)f_{\varepsilon_i}(g(u)-b(u)+[c(u,1)-0.5]\tau_0(u)|u)dQ(u)(1+o(1))\\ =&o(K_n^{-(p+2)}),\end{aligned}$$

by which $\sqrt{nK_n}o(K_n^{-(p+2)}) = o(1)$. Consequently, we have $E[Z_n|X_n]/\sqrt{V[Z_n|X_n]} = o_P(1)$ and Lemma 3 holds. ■

Lemma 4. Let $w_{in} = \sqrt{\frac{K_n}{n}}w(x_i,t_i)c(x_i,t_i)B(x_i)^T\delta (i = 1,...,n)$ for $\delta \in \mathbb{R}^{K_n+q}$. Then, under the assumptions of Theorem 2,

$$2\sum_{i=1}^{n}\int_0^{w_{in}}[I(U_i \le s) - I(U_i \le 0)]ds \overset{as}{\approx} K_n\delta^T D\delta.$$

Proof of Lemma 4. Let

$$R_n = 2\sum_{i=1}^{n}\int_0^{w_{in}}[I(U_i \le s) - I(U_i \le 0)]ds.$$

Since

$$\begin{aligned}
&E[\int_0^{w_{in}}[I(U_i \le s) - I(U_i \le 0)]ds|X_i = x_i, T_i = t_i] \\
=&\int_0^{w_{in}} E[I(U_i \le s) - I(U_i \le 0)|X_i = x_i, T_i = t_i]ds \\
=&\int_0^{w_{in}} E\big[I\left(Y_i \le g(X_i) + c(X_i,T_i)\tau^*(X_i) + \frac{s}{w(X_i,T_i)}\right) \\
&- I\left(Y_i \le g(X_i) + c(X_i,T_i)\tau^*(X_i)\right)\Big|X_i = x_i, T_i = t_i\big]ds \\
=&\int_0^{w_{in}} P\left(Y_i \le g(X_i) + c(X_i,T_i)\tau^*(X_i) + \frac{s}{w(X_i,T_i)}\Big|X_i = x_i, T_i = t_i\right) \\
&- P\left(Y_i \le g(X_i) + c(X_i,T_i)\tau^*(X_i)\Big|X_i = x_i, T_i = t_i\right)ds \\
=&\sqrt{\frac{K_n}{n}}w(x_i,t_i)c(x_i,t_i) \\
&\times\int_0^{B(x_i)^T\delta} P\left(Y_i \le g(X_i) + c(X_i,T_i)\tau^*(X_i) + \sqrt{\frac{K_n}{n}}c(X_i,T_i)t\Big|X_i = x_i, T_i = t_i\right) \\
&- P\left(Y_i \le g(X_i) + c(X_i,T_i)\tau^*(X_i)\Big|X_i = x_i, T_i = t_i\right)dt \\
=&\frac{K_n}{n}w(x_i,t_i)c(x_i,t_i)^2\int_0^{B(x_i)^T\delta} f\left(g(X_i) + c(X_i,T_i)\tau^*(X_i)\Big|X_i = x_i, T_i = t_i\right)tdt(1+o(1)) \\
=&\frac{K_n}{2n}w(x_i,t_i)c(x_i,t_i)^2 f\left(g(X_i) + c(X_i,T_i)\tau^*(X_i)\Big|X_i = x_i, T_i = t_i\right)\{B(x_i)^T\delta\}^2(1+o(1)).
\end{aligned}$$

Therefore we obtain

$$
\begin{aligned}
E[R_n|X_n] = 2\frac{K_n}{2n}\sum_{i=1}^{n}\Bigg\{ & p(x_i)w(x_i,1)c(x_i,1)^2 f_1\left(g(X_i)+c(X_i,T_i)\tau^*(X_i)\middle| X_i=x_i, T_i=1\right) \\
& + (1-p(x_i))w(x_i,-1)c(x_i,-1)^2 \\
& \times f_{-1}\left(g(X_i)+c(X_i,T_i)\tau^*(X_i)\middle| X_i=x_i, T_i=-1\right)\Bigg\} \\
& \times \delta^T B(x_i)B(x_i)^T\delta(1+o(1)) \\
& = K_n\delta^T\Bigg\{\frac{1}{n}\sum_{i=1}^{n}[p(x_i)w(x_i,1)c(x_i,1)^2 f_1(g(x_i)+c(x_i,1)\tau_0(x_i)|x_i) \\
& + (1-p(x_i))w(x_i,-1)c(x_i,-1)^2 f_{-1}(g(x_i)+c(x_i,-1)\tau_0(x_i)|x_i)]B(x_i)B(x_i)^T\Bigg\} \\
& \times \delta(1+o_P(1)) \\
& = K_n\delta^T D\delta(1+o_P(1)).
\end{aligned}
$$

For $i=1,...,n$, we have

$$
\int_0^{w_{in}}[I(u_i\le s)-I(u_i\le 0)]ds \le \sqrt{\frac{K_n}{n}}w(x_i,t_i)c(x_i,t_i)B(x_i)^T\delta.
$$

Therefore the variance of R_n can be evaluated as

$$
\begin{aligned}
V[R_n|X_n] &\le \sum_{i=1}^{n} E\left[\left(\int_0^{w_{in}}[I(u_i\le s)-I(u_i\le 0)]ds\right)^2\middle| X_i=x_i\right] \\
&\le \sqrt{\frac{K_n}{n}}max_{i=1,...,n}\left\{w(x_i,t_i)c(x_i,t_i)\right\}E[R_n|X_n].
\end{aligned}
$$

Since $E[R_n|X_n]=O(K_n)$, we obtain $\sqrt{V[R_n|X_n]}/E[R_n|X_n]=o_P(1)$ and hence, Lemma 4 holds.■

Now we are ready to prove Theorem 2.

Proof of Theorem 2. The objective function of proposed method is

$$\begin{aligned}
L(\beta) &= \sum_{i=1}^{n} w(X_i, T_i) \left| Y_i - g(X_i) - c(X_i, T_i) B(X_i)^T \beta \right| \\
&= \sum_{i=1}^{n} w(X_i, T_i) \left| Y_i - g(X_i) - c(X_i, T_i) B(X_i)^T (\beta - \beta^* + \beta^*) \right| \\
&= \sum_{i=1}^{n} w(X_i, T_i) \left| Y_i - g(X_i) - c(X_i, T_i) B(X_i)^T \beta^* - c(X_i, T_i) B(X_i)^T (\beta - \beta^*) \right|.
\end{aligned}$$

Let

$$U_n(\delta) = \sum_{i=1}^{n} \left[\left| U_i - \sqrt{\frac{K_n}{n}} w(X_i, T_i) c(X_i, T_i) B(X_i)^T \delta \right| - \left| U_i \right| \right] = U_{1n}(\delta) + U_{2n}(\delta),$$

where $U_i = w(X_i, T_i) \left(Y_i - g(X_i) - c(X_i, T_i) B(X_i)^T \beta^* \right)$. Then the minimizer $\hat{\delta}_n$ of $U_n(\delta)$ can be obtained as

$$\hat{\delta}_n = \sqrt{\frac{n}{K_n}} (\hat{\beta} - \beta^*).$$

Following the Knight's identity, we can write $U_n(\delta)$ as

$$U_n(\delta) = U_{1n}(\delta) + U_{2n}(\delta),$$

where

$$\begin{aligned}
U_{1n}(\delta) &= -\sqrt{\frac{K_n}{n}} \sum_{i=1}^{n} w(X_i, T_i) c(X_i, T_i) B(X_i)^T \delta [1 - 2I(U_i < 0)] \\
U_{2n}(\delta) &= 2 \sum_{i=1}^{n} \int_0^{w_{in}} I(U_i \leq s) - I(U_i \leq 0) ds,
\end{aligned}$$

where $w_{in} = \sqrt{\frac{K_n}{n}} w(x_i, t_i) c(x_i, t_i) B(x_i)^T \delta$. From Lemma 3,

$$U_{1n}(\delta) \overset{as}{\sim} -\sqrt{K_n} W^T \delta,$$

where $W \sim N(0, G)$. Furthermore, Lemma 4 yield

$$U_{2n}(\delta) \overset{as}{\approx} K_n \delta^T D \delta.$$

Therefore, for both methods we obtain

$$U_n(\delta) \overset{as}{\approx} U_{0n}(\delta) = -\sqrt{K_n} W^T \delta + K_n \delta^T D \delta.$$

Because $U_{0n}(\delta)$ is convex with respect to δ and has unique minimizer, the minimizer of $U_n(\delta)$ converges to $\delta_0 = argmin_\delta\{U_{0n}(\delta)\}$. This fact is detailed in Knight (1998). Hence, we have

$$\sqrt{\frac{n}{K_n}}\{\hat{\beta} - \beta^*\} \overset{as}{\approx} \delta_0 = D^{-1}\left(\frac{1}{2\sqrt{K_n}} W\right).$$

Since $\hat{\tau}(x) - \tau^*(x) = B(x)^T(\hat{\beta} - \beta^*)$, we obtain for $x \in (0, 1)$, as $n \to \infty$,

$$\sqrt{\frac{n}{K_n}}\{\hat{\tau}(x) - \tau^*(x)\} \overset{D}{\to} N(0, \Psi(x)),$$

where $\Psi(x) = lim_{n\to\infty} \frac{1}{4K_n} B(x)^T D^{-1} G D^{-1} B(x)$ by the definition of W.

Under the condition $K_n = O(n^{1/(2q+3)})$, we have

$$\sqrt{\frac{n}{K_n}}\{\hat{\tau}(x) - \tau_0(x)\} = \sqrt{\frac{n}{K_n}}\{\hat{\tau}(x) - \tau^*(x) - b^a(x) + o(K_n^{-(q+1)})\}$$

and $\sqrt{\frac{n}{K_n}} b^a(x) = O\left(\sqrt{\frac{n}{K_n}} K_n^{-(q+1)}\right) = O(1)$. This completes the proof. ■

A.4. Simulation Results of Settings 1-3

Table A.1: Simulation Results for Setting 1.

ξ_o	Measurement	Bias.sq	Var	MSE	MAE	Recall	Specificity	$Q(\hat{\eta})$
0	MCMEA	0.34	0.08	0.42	0.48	1.00	0.49	1.18
	L_1-MCMEA	0.27	0.23	0.50	0.52	1.00	0.88	1.18
	RL	0.18	0.24	0.42	0.47	1.00	0.60	1.20
	L_1-RL	0.14	0.32	0.46	0.50	1.00	0.93	1.19
	AL	0.35	0.19	0.54	0.47	1.00	0.53	1.18
	L_1-AL	0.28	0.37	0.64	0.52	1.00	0.86	1.18
	QL	0.35	0.16	0.51	0.43	1.00	0.30	1.17
	L_1-QL	0.16	0.28	0.44	0.46	1.00	0.41	1.19
0.05	MCMEA	0.92	0.59	1.51	0.95	0.90	0.50	1.04
	L_1-MCMEA	0.27	0.33	0.61	0.53	1.00	0.86	1.17
	RL	0.55	0.79	1.34	0.88	0.94	0.49	1.07
	L_1-RL	0.15	0.41	0.55	0.55	1.00	0.91	1.18
	AL	0.91	0.69	1.60	0.97	0.88	0.54	1.03
	L_1-AL	0.32	0.47	0.79	0.55	1.00	0.85	1.17
	QL	0.76	0.54	1.30	0.88	0.96	0.46	1.07
	L_1-QL	0.17	0.31	0.48	0.48	1.00	0.43	1.18
0.1	MCMEA	1.41	0.75	2.16	1.17	0.72	0.62	0.91
	L_1-MCMEA	0.28	0.42	0.70	0.56	1.00	0.85	1.16
	RL	1.08	0.99	2.08	1.13	0.77	0.59	0.94
	L_1-RL	0.15	0.45	0.61	0.57	1.00	0.92	1.18
	AL	1.39	0.88	2.27	1.19	0.70	0.64	0.90
	L_1-AL	0.34	0.60	0.94	0.60	0.99	0.81	1.16
	QL	1.20	0.66	1.86	1.08	0.86	0.54	0.98
	L_1-QL	0.17	0.33	0.50	0.50	1.00	0.48	1.18
0.15	MCMEA	1.72	0.88	2.60	1.29	0.58	0.69	0.81
	L_1-MCMEA	0.30	0.51	0.81	0.61	0.99	0.85	1.15
	RL	1.48	1.12	2.60	1.28	0.62	0.69	0.83
	L_1-RL	0.15	0.52	0.67	0.60	0.99	0.91	1.17
	AL	1.71	0.99	2.69	1.31	0.56	0.71	0.80
	L_1-AL	0.39	0.76	1.15	0.67	0.97	0.83	1.15
	QL	1.15	0.78	2.23	1.19	0.79	0.56	0.91
	L_1-QL	0.18	0.38	0.56	0.53	0.99	0.47	1.17
0.2	MCMEA	1.98	0.98	2.96	1.39	0.46	0.76	0.73
	L_1-MCMEA	0.36	0.67	1.03	0.67	0.98	0.83	1.14
	RL	1.79	1.20	2.99	1.39	0.49	0.75	0.75
	L_1-RL	0.16	0.59	0.75	0.63	0.99	0.89	1.17
	AL	1.95	1.10	3.06	1.40	0.44	0.77	0.73
	L_1-AL	0.50	1.00	1.50	0.75	0.96	0.81	1.13
	QL	1.70	0.88	2.59	1.29	0.70	0.61	0.84
	L_1-QL	0.19	0.42	0.61	0.54	0.99	0.52	1.17

Note: In the presence of outliers, L_1-MCMEA and L_1-RL outperformed their L_2-based counterparts. The MSE and MAE decreased and sensitivity, specificity, and $Q(\hat{\eta})$ increased with sample size.

Table A.2: Simulation results of Setting 2.

ξ_o	Sample Size	Method	Bias.sq	Var	MSE	MAE	Recall	Specificity	$Q(\hat{\eta})$
0.05	200	MCMEA	2.29	0.92	3.21	1.43	0.32	0.83	0.66
		L_1-MCMEA	1.32	1.34	2.65	1.20	0.72	0.62	0.92
		RL	1.76	1.85	3.61	1.47	0.39	0.82	0.71
		L_1-RL	0.72	2.58	3.30	1.34	0.84	0.52	0.97
		AL	2.28	1.49	3.77	1.52	0.26	0.88	0.64
		L_1-AL	1.31	2.77	4.08	1.40	0.60	0.67	0.86
		QL	2.14	0.79	2.92	1.37	0.69	0.57	0.78
		L_1-QL	0.64	1.19	1.83	0.99	0.89	0.37	1.02
	500	MCMEA	1.49	0.75	2.24	1.18	0.69	0.65	0.88
		L_1-MCMEA	0.50	0.65	1.14	0.74	0.97	0.71	1.11
		RL	0.98	1.17	2.15	1.14	0.73	0.64	0.91
		L_1-RL	0.26	0.87	1.14	0.79	0.98	0.77	1.14
		AL	1.48	0.97	2.44	1.23	0.63	0.70	1.10
		L_1-AL	0.53	1.04	0.82	0.82	0.96	0.67	0.91
		QL	1.26	0.72	1.97	1.11	0.87	0.50	0.95
		L_1-QL	0.25	0.50	0.75	0.62	0.99	0.46	1.15
	1000	MCMEA	0.92	0.59	1.51	0.95	0.90	0.50	1.04
		L_1-MCMEA	0.27	0.33	0.61	0.53	1.00	0.86	1.17
		RL	0.55	0.79	1.34	0.88	0.94	0.49	1.07
		L_1-RL	0.15	0.41	0.55	0.55	1.00	0.91	1.18
		AL	0.91	0.69	1.60	0.97	0.88	0.54	1.03
		L_1-AL	0.32	0.47	0.79	0.55	1.00	0.85	1.17
		QL	0.76	0.54	1.30	0.88	0.96	0.46	1.07
		L_1-QL	0.17	0.31	0.48	0.48	1.00	0.43	1.18
0	200	MCMEA	1.22	0.73	1.95	1.04	0.84	0.61	1.00
		L_1-MCMEA	1.20	1.10	2.31	1.11	0.83	0.57	0.98
		RL	0.45	0.96	1.41	0.85	0.98	0.62	1.10
		L_1-RL	0.73	2.23	2.96	1.26	0.88	0.53	1.01
		AL	1.22	1.62	2.84	1.17	0.74	0.70	0.94
		L_1-AL	1.19	2.45	3.64	1.29	0.75	0.57	0.93
		QL	1.34	0.33	1.67	1.01	0.93	0.70	1.03
		L_1-QL	0.56	1.03	1.59	0.93	0.92	0.34	1.05
	500	MCMEA	0.57	0.26	0.83	0.65	1.00	0.51	1.15
		L_1-MCMEA	0.44	0.50	0.94	0.68	0.99	0.74	1.14
		RL	0.26	0.25	0.51	0.48	1.00	0.62	1.18
		L_1-RL	0.28	0.76	1.04	0.74	0.99	0.81	1.14
		AL	0.55	0.42	0.96	0.64	0.99	0.55	1.15
		L_1-AL	0.47	0.80	1.27	0.73	0.97	0.67	1.13
		QL	0.48	0.29	0.76	0.55	1.00	0.31	1.14
		L_1-QL	0.24	0.45	0.69	0.59	0.99	0.42	1.16
	1000	MCMEA	0.38	0.14	0.52	0.50	1.00	0.49	1.18
		L_1-MCMEA	0.27	0.27	0.54	0.52	1.00	0.88	1.18
		RL	0.18	0.13	0.31	0.37	1.00	0.60	1.20
		L_1-RL	0.14	0.36	0.50	0.52	1.00	0.93	1.19
		AL	0.35	0.19	0.54	0.47	1.00	0.53	1.18
		L_1-AL	0.28	0.37	0.64	0.52	1.00	0.86	1.18
		QL	0.35	0.16	0.51	0.43	1.00	0.30	1.17
		L_1-QL	0.16	0.28	0.44	0.46	1.00	0.41	1.19

Note: In the presence of outliers, L_1-MCMEA and L_1-RL outperformed their L_2-based counterparts. The MSE and MAE decreased and sensitivity and specificity increased with sample size.

Table A.3: Simulation results of Setting 3.

ξ_o	Dimension	Method	Bias.sq	Var	MSE	MAE	Recall	Specificity	$Q(\hat{\eta})$
0.05	10	MCMEA	0.92	0.59	1.51	0.95	0.90	0.50	1.04
		L_1-MCMEA	0.27	0.33	0.61	0.53	1.00	0.86	1.17
		RL	0.55	0.79	1.34	0.88	0.94	0.49	1.07
		L_1-RL	0.15	0.41	0.55	0.55	1.00	0.91	1.18
		AL	0.91	0.69	1.60	0.97	0.88	0.54	1.03
		L_1-AL	0.32	0.47	0.79	0.55	1.00	0.85	1.17
		QL	0.76	0.54	1.30	0.88	0.96	0.46	1.07
		L_1-QL	0.17	0.31	0.48	0.48	1.00	0.43	1.18
	30	MCMEA	1.06	0.61	1.67	1.05	0.87	0.73	1.01
		L_1-MCMEA	0.31	0.37	0.68	0.56	0.99	0.99	1.17
		RL	0.62	0.89	1.51	0.98	0.85	0.75	0.99
		L_1-RL	0.18	0.42	0.60	0.59	1.00	0.98	1.18
		AL	1.06	0.72	1.78	1.12	0.78	0.76	0.98
		L_1-AL	0.33	0.50	0.83	0.58	0.99	0.98	1.16
		QL	0.87	0.61	1.48	0.96	0.87	0.72	1.00
		L_1-QL	0.25	0.33	0.58	0.50	0.99	0.99	1.17
	50	MCMEA	1.14	0.63	1.77	1.12	0.80	0.82	1.01
		L_1-MCMEA	0.32	0.43	0.75	0.59	0.99	0.99	1.17
		RL	0.69	0.92	1.61	1.05	0.79	0.81	0.98
		L_1-RL	0.21	0.44	0.65	0.61	0.99	0.98	1.18
		AL	1.10	0.75	1.85	1.17	0.73	0.84	0.96
		L_1-AL	0.33	0.55	0.88	0.62	0.98	0.99	1.15
		QL	0.96	0.62	1.58	1.04	0.83	0.75	0.98
		L_1-QL	0.29	0.34	0.63	0.52	0.98	0.99	1.17
0	10	MCMEA	0.34	0.08	0.42	0.48	1.00	0.49	1.18
		L_1-MCMEA	0.27	0.23	0.50	0.52	1.00	0.88	1.18
		RL	0.18	0.24	0.42	0.47	1.00	0.60	1.20
		L_1-RL	0.14	0.32	0.46	0.50	1.00	0.93	1.19
		AL	0.35	0.19	0.54	0.47	1.00	0.53	1.18
		L_1-AL	0.28	0.37	0.64	0.52	1.00	0.86	1.18
		QL	0.35	0.16	0.41	0.43	1.00	0.30	1.17
		L_1-QL	0.16	0.28	0.44	0.46	1.00	0.41	1.19
	30	MCMEA	0.35	0.12	0.47	0.51	1.00	0.54	1.18
		L_1-MCMEA	0.28	0.26	0.54	0.55	1.00	0.99	1.18
		RL	0.20	0.26	0.46	0.50	1.00	0.60	1.19
		L_1-RL	0.19	0.32	0.51	0.53	1.00	0.98	1.19
		AL	0.39	0.19	0.58	0.50	1.00	0.58	1.18
		L_1-AL	0.30	0.38	0.68	0.55	0.99	0.99	1.18
		QL	0.30	0.15	0.45	0.46	0.99	0.97	1.13
		L_1-QL	0.16	0.32	0.48	0.49	0.99	0.99	1.19
	50	MCMEA	0.37	0.15	0.52	0.54	1.00	0.60	1.17
		L_1-MCMEA	0.29	0.29	0.58	0.58	1.00	0.99	1.18
		RL	0.21	0.27	0.48	0.53	1.00	0.61	1.17
		L_1-RL	0.21	0.35	0.56	0.56	1.00	0.99	1.19
		AL	0.42	0.20	0.62	0.53	1.00	0.63	1.16
		L_1-AL	0.32	0.40	0.72	0.58	1.00	0.99	1.18
		QL	0.32	0.17	0.49	0.49	1.00	0.97	1.09
		L_1-QL	0.20	0.32	0.52	0.52	0.98	1.00	1.18

Note: Effects of the covariate dimension. With the presence of outliers, the L_1-based methods outperformed the L_2-based methods in MSE and MAE.

A.5. Additional simulation settings and results

A.5.1. High-dimensional training sample with nonparametric independence screening

We designed and conducted simulation on an additional parameter setting, to assess the performance of the proposed methods in high-dimension situations. As described in Section 2.3, we used NIS to screen the covariates in the first step.

We generated data as follows. the dimension of the covariates was indexed by p:

$$\mathbf{X_i} \sim N_p(0, \mathbf{\Sigma}), \quad diag(\mathbf{\Sigma}) = \mathbf{1}, \quad Corr(X_{ij}, X_{ik}) = 0.5^{|j-k|}, i = 1, ..., n,$$

$$D_i|\mathbf{X_i} \sim Bernoulli(p(\mathbf{X_i})), \quad T_i = 2D_i - 1, \quad logit(p(\mathbf{X_i})) = X_{i1} - X_{i2},$$

$$Y_i = b_0(\mathbf{X_i}) + \frac{T_i}{2}\tau_0(\mathbf{X_i}) + \varepsilon_i, \quad \varepsilon_i \sim (1 - p_o)N(0, 1) + p_o logNormal(0, 4),$$

$$b_0(\mathbf{X_i}) = 0.5 + 4X_{i1} + X_{i2} - 3X_{i3}, \tau_0(\mathbf{X_i}) = 2sin(2X_{i1}) - X_{i2} + 3tanh(0.5X_{i3}),$$

where p_o is the proportion of outliers, $n = 1000$, $p_o \in \{0, 0.05\}$, and $p \in \{1000, 3000, 5000\}$

Figure A.1 shows that the NIS performed well in variable selection, especially when there were outliers.

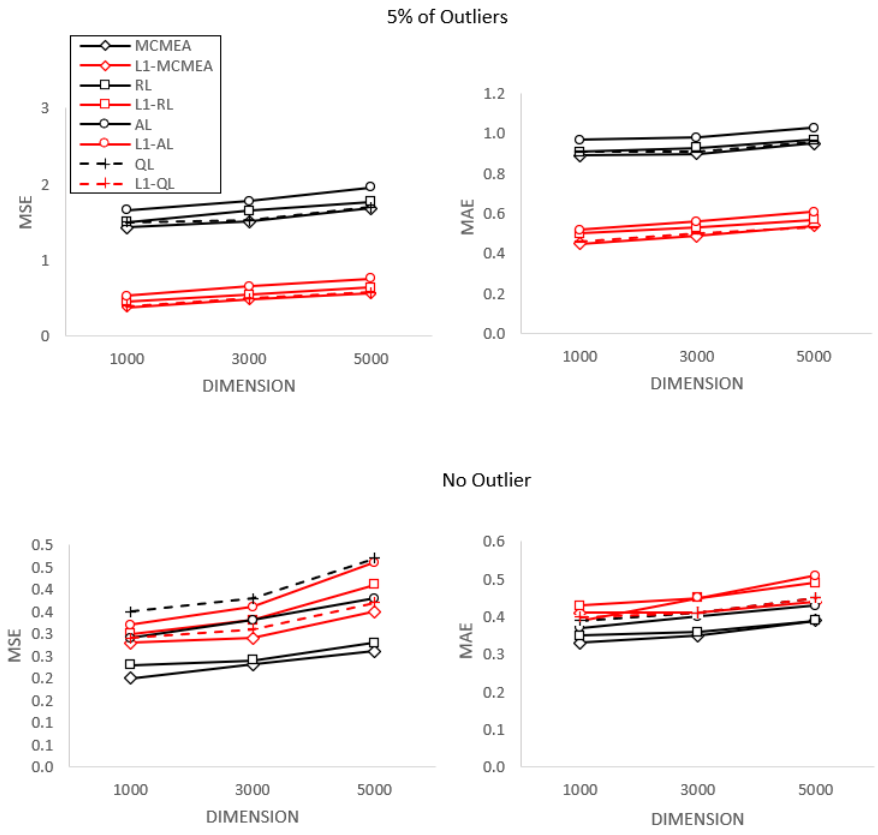


Figure A.1: Influence of dimension.

Note: MSE and MAE tended to increase with dimension. But when there were outliers, the L_1-based methods (red line) performed markedly better than the L_2-based methods (black line).

Table A.4: Simulation results of Setting 4.

ξ_o	Dimension	Method	Bias.sq	Var	MSE	MAE	Recall	Specificity	$Q(\hat{\eta})$
0.05	1000	MCMEA	0.65	0.78	1.43	0.89	0.91	1.00	1.07
		L_1-MCMEA	0.07	0.32	0.38	0.45	0.98	1.00	1.18
		RL	0.61	0.88	1.50	0.91	0.89	1.00	1.06
		L_1-RL	0.06	0.40	0.46	0.50	0.98	1.00	1.18
		AL	0.88	0.78	1.66	0.97	0.86	1.00	1.03
		L_1-AL	0.12	0.41	0.53	0.52	0.98	1.00	1.17
		QL	0.78	0.71	1.50	0.91	0.94	1.00	1.07
		L_1-QL	0.09	0.32	0.40	0.46	0.98	1.00	1.18
	3000	MCMEA	0.56	0.94	1.51	0.90	0.88	1.00	1.05
		L_1-MCMEA	0.07	0.41	0.48	0.49	0.95	1.00	1.17
		RL	0.52	1.13	1.65	0.93	0.87	1.00	1.04
		L_1-RL	0.07	0.48	0.55	0.53	0.95	1.00	1.17
		AL	0.77	1.01	1.78	0.98	0.84	1.00	1.02
		L_1-AL	0.15	0.51	0.66	0.56	0.95	1.00	1.16
		QL	0.67	0.85	1.52	0.91	0.92	1.00	1.05
		L_1-QL	0.11	0.39	0.50	0.50	0.95	1.00	1.17
	5000	MCMEA	0.72	0.96	1.68	0.95	0.86	1.00	1.04
		L_1-MCMEA	0.13	0.43	0.56	0.54	0.93	1.00	1.16
		RL	0.67	1.10	1.77	0.97	0.85	1.00	1.04
		L_1-RL	0.12	0.52	0.64	0.57	0.93	1.00	1.16
		AL	0.98	0.98	1.96	1.03	0.82	1.00	1.01
		L_1-AL	0.24	0.52	0.76	0.61	0.93	1.00	1.15
		QL	0.86	0.84	1.70	0.96	0.91	1.00	1.05
		L_1-QL	0.17	0.42	0.58	0.53	0.93	1.00	1.16
0	1000	MCMEA	0.10	0.10	0.20	0.33	1.00	1.00	1.21
		L_1-MCMEA	0.06	0.22	0.28	0.41	1.00	1.00	1.20
		RL	0.10	0.13	0.23	0.35	1.00	1.00	1.20
		L_1-RL	0.08	0.28	0.30	0.43	1.00	1.00	1.20
		AL	0.17	0.12	0.29	0.37	1.00	1.00	1.19
		L_1-AL	0.11	0.21	0.32	0.39	1.00	1.00	1.19
		QL	0.22	0.14	0.35	0.39	1.00	1.00	1.18
		L_1-QL	0.09	0.20	0.29	0.40	1.00	1.00	1.20
	3000	MCMEA	0.12	0.10	0.23	0.35	1.00	1.00	1.21
		L_1-MCMEA	0.06	0.23	0.29	0.41	1.00	1.00	1.20
		RL	0.11	0.13	0.24	0.36	1.00	1.00	1.20
		L_1-RL	0.06	0.28	0.33	0.45	1.00	1.00	1.20
		AL	0.20	0.13	0.33	0.40	1.00	1.00	1.19
		L_1-AL	0.10	0.26	0.36	0.45	1.00	1.00	1.19
		QL	0.22	0.15	0.38	0.41	1.00	1.00	1.18
		L_1-QL	0.08	0.23	0.31	0.41	1.00	1.00	1.20
	5000	MCMEA	0.15	0.11	0.26	0.39	1.00	1.00	1.20
		L_1-MCMEA	0.11	0.24	0.35	0.28	1.00	1.00	1.20
		RL	0.14	0.13	0.28	0.39	1.00	1.00	1.20
		L_1-RL	0.11	0.30	0.41	0.49	1.00	1.00	1.20
		AL	0.25	0.13	0.38	0.43	1.00	1.00	1.19
		L_1-AL	0.18	0.28	0.46	0.51	1.00	1.00	1.19
		QL	0.30	0.17	0.47	0.45	1.00	1.00	1.18
		L_1-QL	0.13	0.24	0.37	0.45	1.00	1.00	1.19

Note: In the presence of outliers, the L_1-based methods performed markedly better than the L_2-based methods.

A.5.2. Effects of Smoothness Penalty

Finally, we investigated the effects of an added smoothness penalty. We considered a situation involving a univariate covariate x. We visualized the performance differences of the L_1 and L_2 methods, with and without the smoothness penalty.

We generated the data as follows: $X_i \sim Unif(0,1), \tau(X_i) = 3sin(9(X_i - 0.5)), p(X_i) = 1/(1+e^{-X_i}), Y_i = 1 + \frac{T_i}{2}\tau_0(X_i) + \varepsilon_i, \varepsilon_i \sim 0.9N(0,1) + 0.1logNormal(0,4)$, the sample size $n = 1000$, and the validation set with a size of 200. From Figures A.2 and A.3, it is clear that the L_1 methods outperform the L_2 methods, and the L_1 with smoothness penalty greatly improved the performance of the estimation while reducing the variance. The 95% Bootstrap C.I. coverage rate at selected point $x \in \{0.2, 0.5, 0.8\}$ were listed in Table A.5. The asymptotic variances of the L_1-MCM-EA methods without penalty at selected point $x \in \{0.2, 0.5, 0.8\}$ are 0.130, 0.110, and 0.137, corresponding 95% asymptotic C.I. coverage rates are 0.957, 0.973, and 0.959. The asymptotic variances of the L_1-RL methods without penalty at selected points are 0.129, 0.110, and 0.130, corresponding 95% asymptotic C.I. coverage rates are 0.953, 0.969, and 0.957.

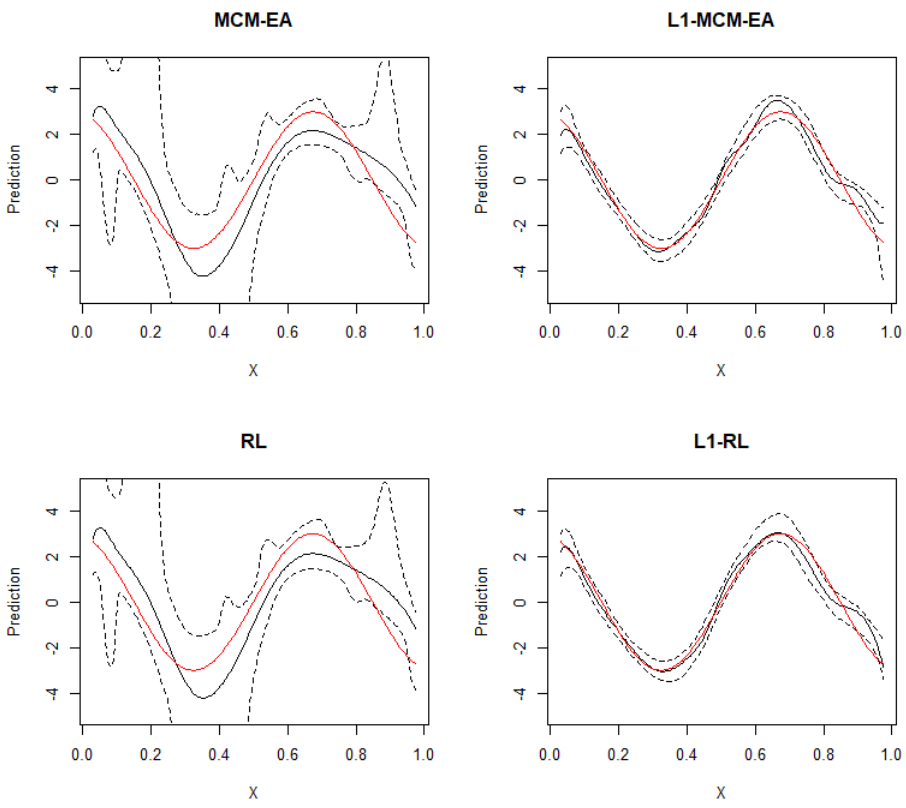


Figure A.2: Effect of smoothness penalty.

Note: Panels on the left are L_2-based methods with smoothness penalty. Panels on the right are L_1-based methods, also with smoothness penalties. The black solid line is the estimate from one replication, the black dashed lines represent quantile 95% bootstrap confidence interval from the same replication, and the red solid line represents the true treatment effect function.

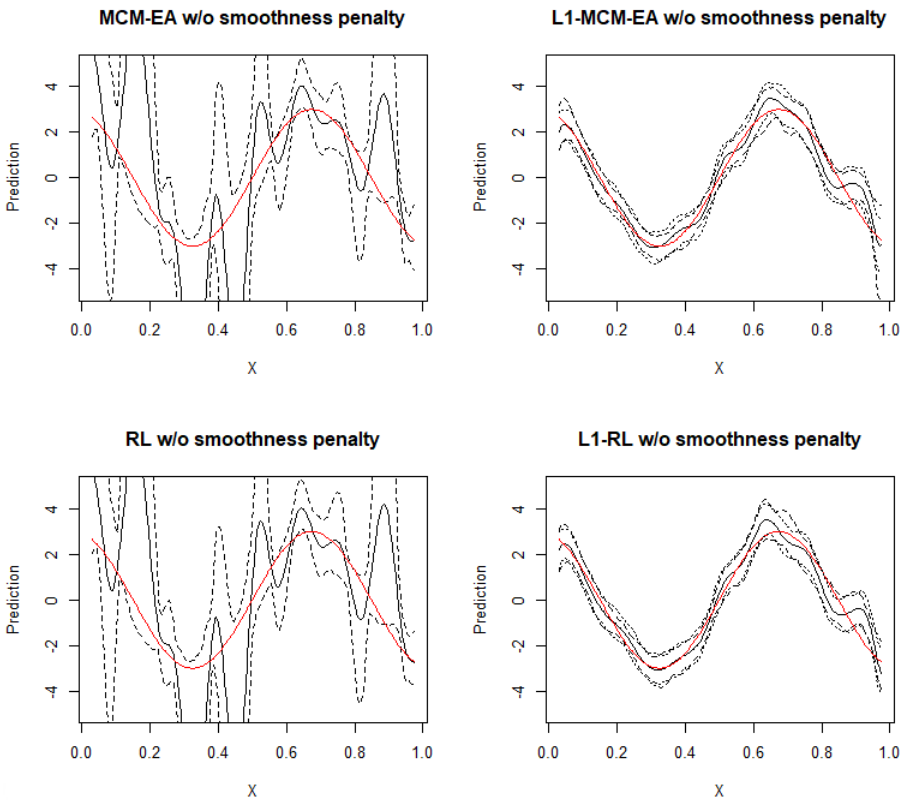


Figure A.3: Effect of smoothness penalty.

Note: Panels on the left are L_2 loss based methods without smoothness penalties, panels on the right are L_1- based methods without smoothness penalties. The black solid line is the estimate from one replication, the black dashed lines represent quantile 95% bootstrap C.I., the black dotted lines represent the 95% asymptotic C.I. from the same replication, and the red solid line represents the true treatment effect function.

Table A.5: Simulation results of Setting 5

Method	X	$\tau(X)$	95% Bootstrap C.I. Coverage Rate
MCMEA	0.2	-1.28	0.773
	0.5	0	0.955
	0.8	1.28	0.791
L_1-MCMEA	0.2	-1.28	0.954
	0.5	0	0.956
	0.8	1.28	0.945
RL	0.2	-1.28	0.784
	0.5	0	0.962
	0.8	1.28	0.798
L_1-RL	0.2	-1.28	0.950
	0.5	0	0.951
	0.8	1.28	0.951
MCMEA w/o penalty	0.2	-1.28	0.927
	0.5	0	0.928
	0.8	1.28	0.931
L_1-MCMEA w/o penalty	0.2	-1.28	0.955
	0.5	0	0.968
	0.8	1.28	0.963
RL w/o penalty	0.2	-1.28	0.929
	0.5	0	0.926
	0.8	1.28	0.931
L_1-RL w/o penalty	0.2	-1.28	0.959
	0.5	0	0.969
	0.8	1.28	0.962

Note: The 95% asymptotic C.I. coverage rates of L_1-MCM-EA and L_1-RL methods with penalty are close to 0.95, but that of L_2-based methods could be far from 0.95.

A.5.3. Comparison of Q-learning and A-learning when model is miss-specified

We investigated the performance of Q-learning and proposed methods when model is miss-specified. We summarize the MSE and MAE of the Q-learning and proposed methods with combination of L_1 and L_2 loss when there is a small amount of outliers.

We generated the data as follows:

$$\mathbf{X_i} \sim N_p(0, \mathbf{\Sigma}), diag(\mathbf{\Sigma}) = \mathbf{1}, Corr(X_{ij}, X_{ik}) = 0.5^{|j-k|}, i = 1, ..., n,$$

$$D_i|\mathbf{X_i} \sim Bernoulli(p(\mathbf{X_i})), T_i = 2D_i - 1, logit(p(\mathbf{X_i})) = X_{i1} - X_{i2},$$

$$Y_i = b_0(\mathbf{X_i}) + \frac{T_i}{2}\tau_0(\mathbf{X_i}) + \varepsilon_i, \varepsilon_i \sim (1 - \xi_o)N(0, 1) + \xi_o Laplace(0, 10),$$

$$b_0(\mathbf{X_i}) = 0.5 + X_{i1} + X_{i2}^2 - 6X_{i3}, \tau_0(\mathbf{X_i}) = 2sin(2X_{i1}) - X_{i2} + 3tanh(0.5X_{i3}),$$

where $n = 1000$, $q = 10$, $\xi_o = 0.1$, and $Q(\eta^{opt}) = 2.18$. For Q-learning, the objective function is

$$L_n(\beta) = \frac{1}{n}\sum_{i=1}^{n} \rho\left(Y_i - X_i^T\gamma - \frac{T_i}{2}B(X_i)^T\beta\right) + \Lambda_n(\beta),$$

where we used L_1 or L_2 loss functions for ρ.

Table A.6: Simulation results of Setting 6

Method	Bias.sq	Var	MSE	MAE	Recall	Specificity	$Q(\hat{\eta})$
MCMEA	1.50	0.82	2.32	1.21	0.68	0.63	1.85
L_1-MCMEA	0.18	0.64	0.82	0.62	0.99	0.74	2.14
RL	1.57	0.73	2.30	1.21	0.68	0.65	1.86
L_1-RL	0.19	0.63	0.83	0.61	0.99	0.73	2.15
AL	1.43	0.89	2.32	1.20	0.69	0.65	1.87
L_1-AL	0.35	2.25	2.61	0.84	0.94	0.75	2.12
QL	1.66	1.11	2.76	1.31	0.74	0.63	1.81
L_1-QL	3.57	1.01	4.58	1.15	1.00	0.52	2.08

Note: When model is mis-specified, L_1-QL and L_2-QL have the largest MSE compared with other methods.

A.6. Existence of outliers

The following figure shows that the outcome observations in both treatment groups are beyond normally distributed.

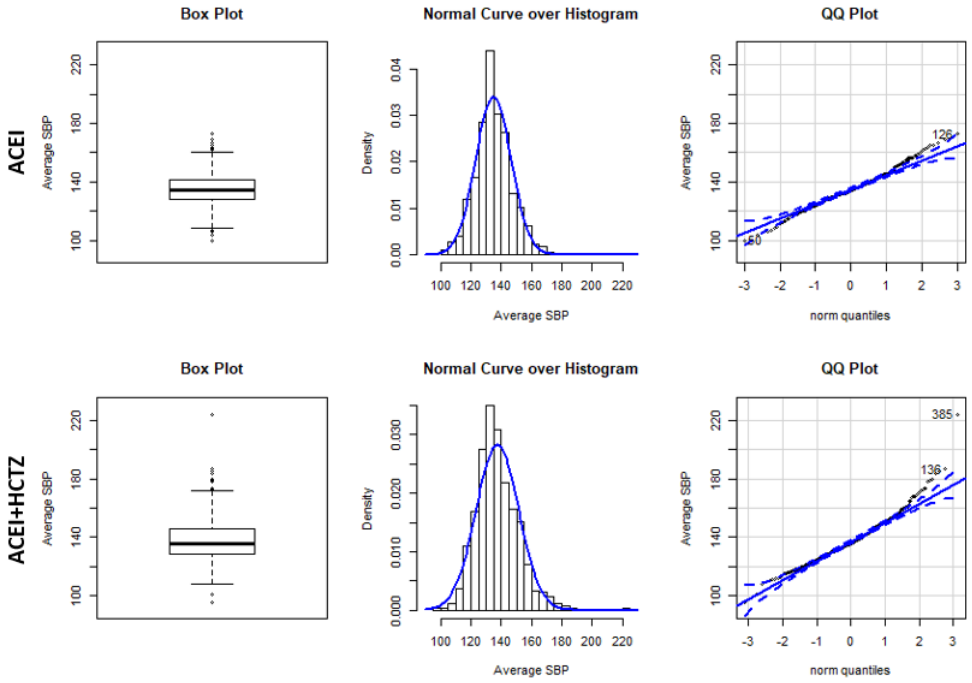


Figure A.4: Heavy-tailed Systolic Blood Pressure Distribution

A.7: Nuisance parameter estimation

The GBM is used to estimate mean outcome and propensity score. The estimation of two groups are as following figure.

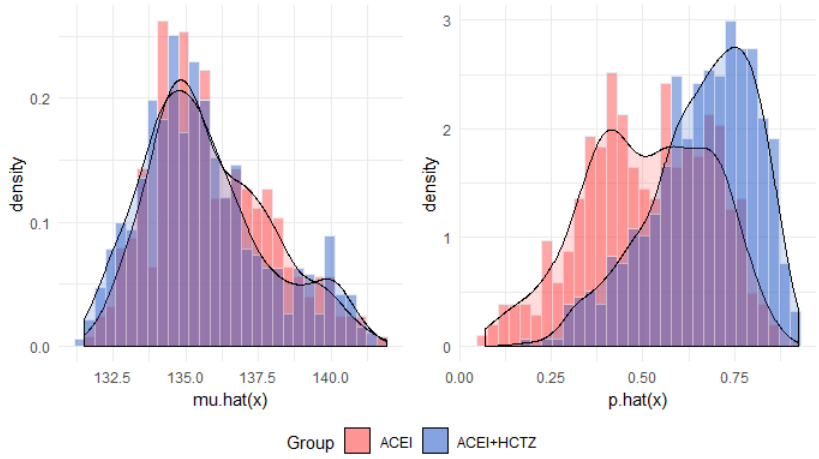


Figure A.5: Histograms of the mean outcomes and the estimated propensity score in the two treatment groups. The mean functions had similar shapes whereas the propensity distributions were clearly different.

In the application of proposed method, the importance levels from GBM are consistent with the result from regression. The importance levels from GBM and the linear and logistic regression results are summarized in the following two tables.

Table A.7: Importance levels from the GBM analysis vs coefficients and p-values from regression analysis.

Variable	Importance (scaled)	Linear regression coefficient	Linear regression p-value
Average PDC	100.0000	-8.3352	<0.001*
BMI	71.4317	0.1217	0.022*
Pulse	57.9355	0.0829	0.052
Male	51.0437	5.3345	<0.001*
Age	40.9329	0.1421	<0.001*
Depression	21.7349	-2.7787	0.007*
CAD	9.6647	3.9908	0.181
Diabetes	8.3385	-1.4464	<0.001*
Stroke	5.7007	3.9820	0.242
Hyperlipidemia	3.3518	-0.6630	0.581
Black	2.9593	0.8341	0.351
CKD	1.6875	-4.6852	0.101
COPD	0.7970	-1.6243	0.272
CHF	0.3214	1.4062	0.668
Atrial fibrillation	0	0.2215	0.968
MI	0	-5.3773	0.405

Table A.8: Propensity models based on GBM and logistic regression

Variable	Importance (scaled)	Logistic regression coefficient	Logistic regression p-value
BMI	100.0000	0.0342	<0.001*
Pulse	75.3721	-0.0162	0.028*
Age	65.0922	0.0205	0.002*
Diabetes	61.9452	-1.2126	<0.001*
Black	27.7566	0.7032	<0.001*
Male	8.5503	-0.2410	0.123
Hyperlipidemia	5.2229	0.3572	0.091
Depression	3.3688	0.2294	0.200
COPD	2.6379	-0.2325	0.353
CAD	2.0312	0.2867	0.577
Stroke	1.9077	-0.9878	0.086
CKD	1.1372	0.1413	0.772
CHF	0.5027	0.1182	0.834
MI	0	0.4516	0.678
Atrial fibrillation	0	1.5788	0.176

Appendix B

Appendix B.1: Implementation

Title: RCATE package

R-package for robust estimation of CATE: R package **RCATE** contains code of 9 robust estimation algorithms of CATE described in the Chapter 3 and also the methods based on additive B-spline LAD regression in Chapter 2. The package also contains the dataset used as example in this dissertation.

Hypertension dataset: Data set used in the illustration of robust estimation of CATE algorithms in Section 3.2.

Example of usage: A simple simulated example.

```
## Install package
require(devtools)
devtools::install_github("rhli-Hannah/RCATE")
library(RCATE)


## Data generation
n <- 1000; p <- 3; set.seed(2223)
X <- as.data.frame(matrix(runif(n*p,-3,3),nrow=n,ncol=p))
tau = 6*sin(2*X[,1])+3*(X[,2]+3)*X[,3]
p = 1/(1+exp(-X[,1]+X[,2]))
d = rbinom(n,1,p)
t = 2*d-1
y = 100+4*X[,1]+X[,2]-3*X[,3]+tau*t/2 + rnorm(n,0,1)
```

```
set.seed(2223)
x_val = as.data.frame(matrix(rnorm(200*3,0,1),
nrow=200,ncol=3))
tau_val = 6*sin(2*x_val[,1])+3*(x_val[,2]+3)*x_val[,3]

## Use robust GBM + R-learning to estimate CATE
fit <- rcate.ml(X,y,d,method='RL',algorithm='GBM')
y_pred <- predict(fit,x_val)$predict
plot(tau_val,y_pred);abline(0,1)
```

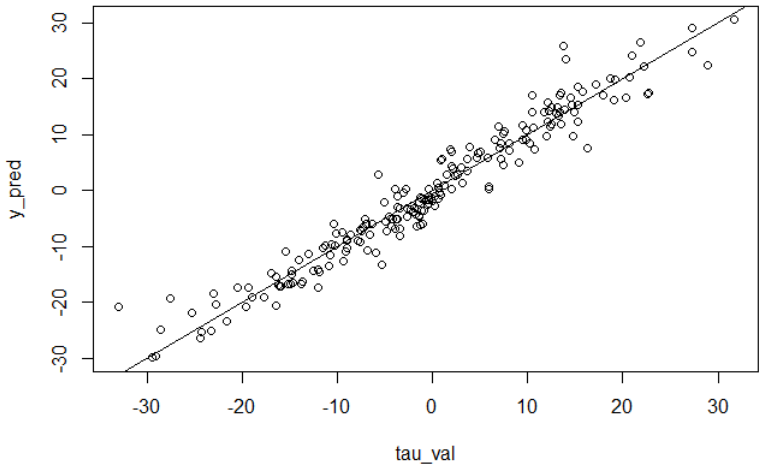


Figure B.1: Comparison of τ_0 and $\hat{\tau}$ from the example of the **RCATE** package

```
## Variable importance level
importance <- importance.rcate(fit)
```

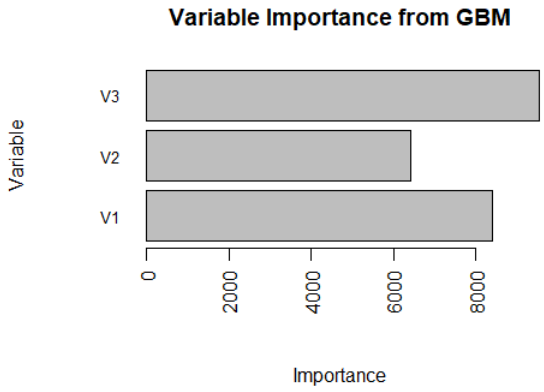
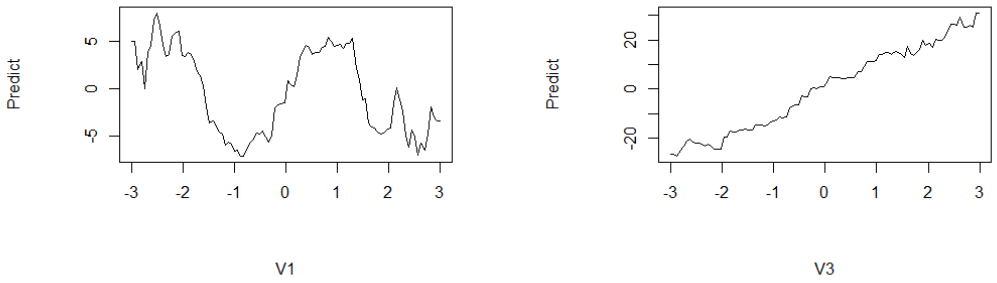


Figure B.2: Variable importance from the example of the **RCATE** package

```
## Marginal treatment effect plot
marginal.rcate(fit,'V1')
marginal.rcate(fit,'V3')
```



(a) Marginal effect of $V1$

(b) Marginal effect of $V2$

Figure B.3: Marginal treatment effects of selected variables from the example of the **RCATE** package

Appendix B.2: Simulation Results

Table B.1: Simulation Results (Coverage Probabilities (%)) of Simulation 1

	MCM-EA													
	$Laplace(0, \sqrt{50})$							$N(0, 100)$						
p_o	0.00	0.05	0.10	0.15	0.20	0.30	0.50	0.00	0.05	0.10	0.15	0.20	0.30	0.50
robust GBM	95	94	91	93	93	95	97	95	94	95	94	94	98	97
GBM	95	93	96	96	94	97	96	95	96	97	99	99	91	95
robust NN	95	94	96	92	94	99	98	95	92	98	92	95	98	93
NN	96	76	87	47	83	79	89	96	39	43	27	32	35	29
robust RF	94	96	97	94	92	93	98	94	95	94	93	92	96	98
RF	96	96	98	97	97	89	96	96	90	92	92	93	89	87
robust GAM	23	38	49	54	57	66	74	23	39	48	54	59	66	76
robust QL	47	49	49	52	53	53	0	47	49	50	51	53	33	0
	RL													
	$Laplace(0, \sqrt{50})$							$N(0, 100)$						
p_o	0.00	0.05	0.10	0.15	0.20	0.30	0.50	0.00	0.05	0.10	0.15	0.20	0.30	0.50
robust GBM	95	94	95	96	95	97	97	95	95	96	94	96	91	98
GBM	93	97	98	98	97	96	95	93	99	98	99	100	100	99
robust NN	94	95	93	96	92	93	96	94	95	96	93	97	91	98
NN	96	82	94	95	88	85	91	96	94	93	94	85	82	36
robust RF	95	96	97	94	98	95	94	95	92	91	90	87	82	72
RF	94	89	98	97	87	93	94	94	82	93	89	95	89	86
robust GAM	27	48	57	61	65	71	80	27	49	58	64	65	72	81
robust QL	47	49	49	52	53	53	0	47	49	50	51	53	33	0
	AIPW													
	$Laplace(0, \sqrt{50})$							$N(0, 100)$						
p_o	0.00	0.05	0.10	0.15	0.20	0.30	0.50	0.00	0.05	0.10	0.15	0.20	0.30	0.50
robust GBM	96	93	92	91	90	89	90	96	94	93	92	90	98	93
GBM	96	92	98	97	95	95	98	96	97	96	99	97	90	91
robust NN	94	92	95	93	92	96	98	94	98	96	93	95	93	98
NN	95	74	89	96	80	73	88	95	44	88	84	87	83	78
robust RF	93	94	98	91	97	96	98	93	96	97	94	95	96	95
RF	94	98	97	93	93	82	96	94	94	91	87	95	86	82
robust GAM	54	57	59	61	62	64	70	54	58	59	61	64	66	73
robust QL	47	49	49	52	53	53	0	47	49	50	51	53	33	0

Table B.2: Simulation Results of Simulation 3 ($n_0 = 200$)

		MCM-EA									
		$Laplace(0, \sqrt{50})$					$N(0, 100)$				
	n	200	400	600	800	1000	200	400	600	800	1000
MSE	robust GBM	3.43	2.28	1.66	1.39	1.14	4.08	2.43	1.80	1.55	1.37
	GBM	19.96	13.87	10.95	8.99	7.69	21.64	14.18	11.33	9.36	7.33
	robust NN	5.70	2.25	1.54	1.12	1.00	5.84	2.45	1.74	1.21	1.14
	NN	6.50	3.95	3.01	2.60	2.17	6.79	4.09	3.01	2.69	2.23
	robust RF	2.96	2.10	1.53	1.37	1.24	3.21	2.05	1.55	1.54	1.25
	RF	11.12	9.94	8.77	7.94	7.75	10.05	9.37	8.33	8.38	7.23
	robust GAM	1.54					2.54				
	n	200	400	600	800	1000	200	400	600	800	1000
MAE	robust GBM	1.44	1.32	0.99	0.90	0.80	1.57	1.21	1.03	0.94	0.87
	GBM	3.40	3.56	2.49	2.23	2.05	3.62	2.91	2.58	2.34	2.06
	robust NN	1.86	1.14	0.93	0.80	0.75	1.89	1.19	0.98	0.83	0.79
	NN	1.99	1.53	1.33	1.23	1.14	2.04	1.57	1.34	1.26	1.14
	robust RF	1.27	0.99	0.90	0.84	0.79	1.32	1.03	0.89	0.89	0.80
	RF	2.02	3.56	1.80	1.73	1.70	2.08	4.08	1.89	1.83	1.75
	robust GAM	0.83					1.13				

		RL									
		$Laplace(0, \sqrt{50})$					$N(0, 100)$				
	n	200	400	600	800	1000	200	400	600	800	1000
MSE	robust GBM	4.87	2.88	2.19	1.77	1.52	6.46	3.19	2.36	2.05	1.77
	GBM	36.93	22.98	17.28	14.28	11.88	38.67	24.42	17.34	16.77	11.55
	robust NN	6.19	2.48	1.48	1.23	1.04	6.28	2.67	1.44	1.33	1.07
	NN	6.78	4.28	3.45	2.87	2.60	7.30	4.39	3.51	3.10	2.62
	robust RF	3.32	2.24	1.76	1.59	1.39	4.14	2.92	2.20	1.92	1.34
	RF	49.57	49.48	9.36	43.16	47.60	133.48	92.12	40.54	68.37	50.74
	robust GAM	1.87					3.03				
	n	200	400	600	800	1000	200	400	600	800	1000
MAE	robust GBM	1.71	1.32	1.07	1.01	0.93	1.95	1.38	1.18	1.08	1.00
	GBM	4.56	3.56	2.50	2.73	2.48	4.78	3.71	3.16	2.97	2.53
	robust NN	1.94	1.19	0.91	0.83	0.77	1.96	1.23	0.92	0.87	0.78
	NN	2.03	1.60	1.43	1.31	1.24	2.12	1.63	1.44	1.36	1.26
	robust RF	1.27	0.99	0.86	0.81	0.72	1.36	1.03	0.87	0.85	0.77
	RF	3.61	3.56	1.81	3.57	3.67	4.61	4.08	3.64	4.35	3.89
	robust GAM	0.79					1.10				

		AIPW									
		$Laplace(0, \sqrt{50})$					$N(0, 100)$				
	n	200	400	600	800	1000	200	400	600	800	1000
MSE	robust GBM	4.55	3.13	2.09	1.70	1.41	5.23	3.15	2.36	1.87	1.73
	GBM	19.10	14.01	11.75	8.61	7.32	20.36	13.40	12.07	9.90	7.52
	robust NN	5.12	2.17	1.57	1.18	0.94	5.88	2.48	1.58	1.24	0.99
	NN	1.97	3.77	2.91	2.46	2.23	6.86	4.24	3.14	2.57	2.34
	robust RF	2.80	1.98	1.36	1.29	1.18	3.25	1.98	1.56	1.37	1.18
	RF	10.57	9.66	9.36	8.58	8.33	9.98	8.81	9.21	9.86	8.66
	robust GAM	2.52					3.04				
	n	200	400	600	800	1000	200	400	600	800	1000
MAE	robust GBM	1.65	1.33	1.07	0.94	0.84	1.77	1.33	1.13	0.99	0.92
	GBM	3.35	2.78	2.50	2.13	1.98	3.50	2.81	2.59	2.31	2.05
	robust NN	1.76	1.11	0.93	0.81	0.73	1.89	1.19	0.96	0.84	0.76
	NN	1.97	1.50	1.31	1.20	1.15	2.05	1.59	1.37	1.23	1.17
	robust RF	1.21	0.99	0.86	0.82	0.78	1.32	1.01	0.89	0.84	0.77
	RF	1.98	1.89	1.81	1.73	1.68	2.09	1.94	1.92	1.87	1.76
	robust GAM	0.99					1.11				

Table B.3: Simulation Results of Simulation 3 ($n_0 = 1000$)

		MCM-EA							
		$Laplace(0, \sqrt{50})$				$N(0, 100)$			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MSE	robust GBM	1.85	1.01	0.80	0.80	1.83	0.97	0.83	0.77
	GBM	11.22	2.71	1.99	1.55	11.10	2.83	2.03	1.59
	robust NN	1.48	1.04	0.84	0.78	1.46	1.03	0.86	0.84
	NN	2.76	2.61	2.18	1.88	2.74	2.76	2.20	1.95
	robust RF	1.63	1.31	1.01	0.84	1.62	1.17	0.96	0.88
	RF	5.12	5.00	4.28	4.13	5.26	4.85	4.11	3.81
	robust GAM	0.87				0.88			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MAE	robust GBM	1.06	0.75	0.67	0.61	1.06	0.74	0.68	0.68
	GBM	2.56	1.23	1.05	0.93	2.58	1.28	1.07	0.94
	robust NN	0.94	0.77	0.69	0.62	0.93	0.77	0.70	0.72
	NN	1.28	1.23	1.13	1.04	1.28	1.27	1.13	1.07
	robust RF	0.94	0.84	0.72	0.67	0.92	0.79	0.71	0.68
	RF	1.51	1.48	1.40	1.36	1.58	1.53	1.44	1.37
	robust GAM	0.60				0.63			

		RL							
		$Laplace(0, \sqrt{50})$				$N(0, 100)$			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MSE	robust GBM	2.21	0.72	0.45	0.30	2.27	0.72	0.45	0.30
	GBM	29.31	4.55	3.01	2.14	27.49	5.02	3.04	1.80
	robust NN	1.29	0.77	0.53	0.36	1.36	0.80	0.55	0.37
	NN	3.58	3.26	2.58	1.90	3.66	3.51	2.55	1.93
	robust RF	0.98	0.73	0.60	0.50	1.09	0.85	0.70	0.62
	RF	107.18	91.18	91.12	72.19	127.63	91.68	81.65	70.37
	robust GAM	0.64				0.61			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MAE	robust GBM	1.14	0.64	0.50	0.40	1.15	0.63	0.49	0.40
	GBM	3.80	1.51	1.20	0.97	3.75	1.59	1.25	0.96
	robust NN	0.88	0.67	0.55	0.46	0.90	0.68	0.56	0.46
	NN	1.45	1.37	1.22	1.04	1.47	1.42	1.22	1.06
	robust RF	0.71	0.62	0.54	0.46	0.70	0.60	0.51	0.47
	RF	5.44	4.89	4.61	4.04	5.73	4.96	4.65	4.19
	robust GAM	0.56				0.53			

		AIPW							
		$Laplace(0, \sqrt{50})$				$N(0, 100)$			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MSE	robust GBM	2.46	1.55	0.82	0.60	2.99	1.29	1.07	0.98
	GBM	14.43	3.98	2.47	1.54	15.25	3.51	4.17	1.88
	robust NN	1.43	0.94	0.66	0.43	1.46	0.89	0.63	0.43
	NN	3.83	3.27	2.54	2.04	3.61	3.15	2.67	2.06
	robust RF	1.28	1.00	0.73	0.57	1.35	0.93	0.75	0.63
	RF	7.80	7.76	7.51	6.20	7.97	7.39	7.09	6.72
	robust GAM	0.63				0.75			
	n	1000	3000	5000	7000	1000	3000	5000	7000
MAE	robust GBM	1.48	0.77	0.59	0.48	1.15	0.76	0.61	0.48
	GBM	2.82	1.37	1.10	0.85	2.87	1.37	1.11	0.85
	robust NN	0.92	0.73	0.61	0.49	0.93	0.72	0.60	0.49
	NN	1.46	1.35	1.20	1.07	1.46	1.34	1.23	1.07
	robust RF	0.81	0.71	0.59	0.47	0.83	0.69	0.60	0.53
	RF	1.65	1.64	1.56	1.45	1.69	1.69	1.59	1.55
	robust GAM	0.52				0.53			

Table B.4: Tuning parameters of considered methods in experiments

Method	Parameter	Value
RF-based algorithms	Number of trees	50
	Fraction of feathers used in splitting	0.8
	Minimum node size	3
Boosting-based algorithms	Number of trees	1000
	Depth of trees	2
	Learning rate	0.1
Robust ANN	Number of hidden layers	2
	Number of neurons in hidden layers	p and $p/2$
	Adam optimization	$\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$
	L_1 regularization ($p = 100, 2000$)	0.1, if $p = 100$; 0.02, if $p = 2000$.
	Number of neurons in hidden layers ($p = 2000$)	$p/10$ and $p/40$
Robust GAM and QL	Number of knots	$\sqrt{n}/2$
	Number of degree	3
	γ in SCAD	3.7

References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Agarwal, G. G. and W. Studden (1980). Asymptotic integrated mean square error using least squares and bias minimizing splines. *The Annals of Statistics*, 1307–1325.

Agarwal, R., D. Schuurmans, and M. Norouzi (2019). Striving for simplicity in off-policy deep reinforcement learning. *arXiv preprint arXiv:1907.04543*.

Agarwal, R., D. Schuurmans, and M. Norouzi (2020). An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pp. 104–114. PMLR.

Allaire, J., F. Chollet, et al. (2019). keras: R interface to'keras'. *R package version 2*(4).

Allaire, J., D. Eddelbuettel, N. Golding, and Y. Tang (2016). *tensorflow: R Interface to TensorFlow*.

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

Athey, S., J. Tibshirani, S. Wager, et al. (2019). Generalized random forests. *The Annals of Statistics 47*(2), 1148–1178.

Bellman, R. (1966). Dynamic programming. *Science 153*(3731), 34–37.

Bertsekas, D. P. and J. N. Tsitsiklis (1996). *Neuro-dynamic programming*. Athena Scientific.

Biau, G. (2012). Analysis of a random forests model. *The Journal of Machine Learning Research 13*(1), 1063–1095.

Breiman, L. (2001). Random forests. *Machine learning 45*(1), 5–32.

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984). *Classification and regression trees*. CRC press.

Caron, A., I. Manolopoulou, and G. Baio (2020). Estimating individual treatment effects using non-parametric regression models: a review. *arXiv preprint arXiv:2009.06472*.

Centers for Disease Control and Prevention (2019). Hypertension cascade: Hypertension prevalence, treatment and control estimates among us adults aged 18 years and older applying the criteria from the american college of cardiology and american heart association's 2017 hypertension guideline-nhanes 2013-2016. *Centers for Disease Control and Prevention*.

Chakraborty, B. (2013). *Statistical methods for dynamic treatment regimes*. Springer.

Chen, S., L. Tian, T. Cai, and M. Yu (2017). A general statistical framework for subgroup identification and comparative treatment scoring. *Biometrics 73*(4), 1199–1209.

Cox, D. R. (1958). The interpretation of the effects of non-additivity in the latin square. *Biometrika 45*(1/2), 69–73.

Daskalaki, E., L. Scarnato, P. Diem, and S. G. Mougiakakou (2010). Preliminary results of a novel approach for glucose regulation using an actor-critic learning based controller.

De Boor, C., C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor (1978). *A practical guide to splines*, Volume 27. springer-verlag New York.

De Pillis, L. G. and A. Radunskaya (2003). The dynamics of an optimally controlled tumor model: A case study. *Mathematical and computer modelling 37*(11), 1221–1244.

Donoho, D. L. et al. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture 1*(2000), 32.

El-Melegy, M. T., M. H. Essai, and A. A. Ali (2009). Robust training of artificial feedforward neural networks. In *Foundations of Computational, Intelligence Volume 1*, pp. 217–242. Springer.

Fan, J., Y. Feng, and R. Song (2011). Nonparametric independence screening in sparse ultra-high-dimensional additive models. *Journal of the American Statistical Association 106*(494), 544–557.

Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(5), 849–911.

Feng, J. and N. Simon (2017). Sparse-input neural networks for high-dimensional nonparametric regression and classification. *arXiv preprint arXiv:1711.07592*.

Friedman, J., T. Hastie, R. Tibshirani, et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics 28*(2), 337–407.

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

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis 38*(4), 367–378.

Fryar, C. D., Y. Ostchega, C. M. Hales, G. Zhang, and D. Kruszon-Moran (2017). Hypertension prevalence and control among adults: United states, 2015-2016.

Fujimoto, S., E. Conti, M. Ghavamzadeh, and J. Pineau (2019). Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*.

Fujimoto, S., D. Meger, and D. Precup (2019). Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062. PMLR.

Gabriel, S. E. (2012). Getting the methods right–the foundation of patient-centered outcomes research. *The New England journal of medicine 367*(9), 787.

Geyer, C. J. (1996). On the asymptotics of convex stochastic optimization. *Unpublished manuscript 37*.

Girosi, F., M. Jones, and T. Poggio (1995). Regularization theory and neural networks architectures. *Neural computation 7*(2), 219–269.

Goodfellow, I., Y. Bengio, A. Courville, and Y. Bengio (2016). *Deep learning*, Volume 1. MIT press Cambridge.

Gottesman, O., F. Johansson, M. Komorowski, A. Faisal, D. Sontag, F. Doshi-Velez, and L. A. Celi (2019). Guidelines for reinforcement learning in healthcare. *Nat Med 25*(1), 16–18.

Gunter, T. D. and N. P. Terry (2005). The emergence of national electronic health record architectures in the united states and australia: models, costs, and questions. *Journal of medical Internet research 7*(1), e3.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Heinze-Deml, C., J. Peters, and N. Meinshausen (2018). Invariant causal prediction for nonlinear models. *Journal of Causal Inference 6*(2).

Hester, T., M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, et al. (2018). Deep q-learning from demonstrations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 32.

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

Hjort, N. and D. Pollard (1993). Asymptotics for minimisers of convex processes technical report. *Yale University*.

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

Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics*, pp. 492–518. Springer.

Imai, K., G. King, and E. A. Stuart (2008). Misunderstandings between experimentalists and observationalists about causal inference. *Journal of the royal statistical society: series A (statistics in society) 171*(2), 481–502.

James, P. A., S. Oparil, B. L. Carter, W. C. Cushman, C. Dennison-Himmelfarb, J. Handler, D. T. Lackland, M. L. LeFevre, T. D. MacKenzie, O. Ogedegbe, et al. (2014). 2014 evidence-based guideline for the management of high blood pressure in adults: report from

the panel members appointed to the eighth joint national committee (jnc 8). *Jama 311*(5), 507–520.

Jaques, N., A. Ghandeharioun, J. H. Shen, C. Ferguson, A. Lapedriza, N. Jones, S. Gu, and R. Picard (2019). Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.

Kidambi, R., A. Rajeswaran, P. Netrapalli, and T. Joachims (2020). Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Knight, K. (1998). Limiting distributions for l1 regression estimators under general conditions. *Annals of statistics*, 755–770.

Kuhn, M. (2012). The caret package. *R Foundation for Statistical Computing, Vienna, Austria. URL https://cran. r-project. org/package= caret*.

Kumar, A., J. Fu, M. Soh, G. Tucker, and S. Levine (2019). Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, pp. 11784–11794.

Kumar, A., A. Zhou, G. Tucker, and S. Levine (2020). Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*.

Künzel, S. R., J. S. Sekhon, P. J. Bickel, and B. Yu (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences 116*(10), 4156–4165.

Laroche, R., P. Trichelair, and R. T. Des Combes (2019). Safe policy improvement with baseline bootstrapping. In *International Conference on Machine Learning*, pp. 3652–3661. PMLR.

Levine, S., A. Kumar, G. Tucker, and J. Fu (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Li, Wang, T. (2021). Robust estimation of heterogeneous treatment effects using electronic health record data. *Statistics in Medicine*.

Liano, K. (1996). Robust error measure for supervised neural network learning with outliers. *IEEE Transactions on Neural Networks 7*(1), 246–250.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, S., K. C. See, K. Y. Ngiam, L. A. Celi, X. Sun, and M. Feng (2020). Reinforcement learning for clinical decision support in critical care: comprehensive review. *Journal of medical Internet research 22*(7), e18477.

Lu, W., H. H. Zhang, and D. Zeng (2013). Variable selection for optimal treatment decision. *Statistical methods in medical research 22*(5), 493–504.

McCaffrey, D. F., G. Ridgeway, and A. R. Morral (2004). Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological methods 9*(4), 403.

Meier, L., S. Van de Geer, P. Bühlmann, et al. (2009). High-dimensional additive modeling. *The Annals of Statistics 37*(6B), 3779–3821.

Meinshausen, N. and G. Ridgeway (2006). Quantile regression forests. *Journal of Machine Learning Research 7*(6).

Metz, L., J. Ibarz, N. Jaitly, and J. Davidson (2017). Discrete sequential prediction of continuous actions for deep rl. *arXiv preprint arXiv:1705.05035*.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. (2015). Human-level control through deep reinforcement learning. *nature 518*(7540), 529–533.

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

Nair, V. and G. E. Hinton (2010). Rectified linear units improve restricted boltzmann machines. In *Icml*.

Nie, X. and S. Wager (2020, 09). Quasi-Oracle Estimation of Heterogeneous Treatment Effects. *Biometrika*. asaa076.

Osterberg, L. and T. Blaschke (2005). Adherence to medication. *New England journal of medicine 353*(5), 487–497.

Oza, N. C. and K. Tumer (2008). Classifier ensembles: Select real-world applications. *Information fusion 9*(1), 4–20.

Peng, B. and L. Wang (2015). An iterative coordinate descent algorithm for high-dimensional nonconvex penalized quantile regression. *Journal of Computational and Graphical Statistics 24*(3), 676–694.

Powers, S., J. Qian, K. Jung, A. Schuler, N. H. Shah, T. Hastie, and R. Tibshirani (2018). Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in medicine 37*(11), 1767–1787.

RColorBrewer, S. and M. A. Liaw. Package 'randomforest'.

Ridgeway, G. (2007). Generalized boosted models: A guide to the gbm package. *Update 1*(1), 2007.

Ridgeway, G. and M. G. Ridgeway (2004). The gbm package. *R Foundation for Statistical Computing, Vienna, Austria 5*(3).

Ridgeway, G., M. H. Southworth, and S. RUnit (2013). Package 'gbm'. *Viitattu 10*(2013), 40.

Robins, J. M. (2004). Optimal structural nested models for optimal sequential decisions. In *Proceedings of the second seattle Symposium in Biostatistics*, pp. 189–326. Springer.

Robins, J. M. and A. Rotnitzky (1995). Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association 90*(429), 122–129.

Robinson, P. M. (1988). Root-n-consistent semiparametric regression. *Econometrica: Journal of the Econometric Society*, 931–954.

Rosenbaum, P. R. and D. B. Rubin (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika 70*(1), 41–55.

Rosenstein, M. T., A. G. Barto, J. Si, A. Barto, and W. Powell (2004). Supervised actor-critic reinforcement learning. *Learning and Approximate Dynamic Programming: Scaling Up to the Real World*, 359–380.

Rosenthal, R. and L. Jacobson (1968). Pygmalion in the classroom. *The urban review 3*(1), 16–20.

Roy, M.-H. and D. Larocque (2012). Robustness of random forests for regression. *Journal of Nonparametric Statistics 24*(4), 993–1006.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology 66*(5), 688.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *nature 323*(6088), 533–536.

Schulte, P. J., A. A. Tsiatis, E. B. Laber, and M. Davidian (2014). Q-and a-learning methods for estimating optimal dynamic treatment regimes. *Statistical science: a review journal of the Institute of Mathematical Statistics 29*(4), 640.

Sekhon, J. S. (2008). The neyman-rubin model of causal inference and estimation via matching methods. *The Oxford handbook of political methodology 2*, 1–32.

Shah, R. D. and P. Bühlmann (2018). Goodness-of-fit tests for high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 80*(1), 113–135.

Sherwood, B. and A. Maidman (2016). rqpen: Penalizedquantile regression. *R package version 1*.

SPRINT Research Group (2015). A randomized trial of intensive versus standard blood-pressure control. *New England Journal of Medicine 373*(22), 2103–2116.

Stuart, E. A., E. DuGoff, M. Abrams, D. Salkever, and D. Steinwachs (2013). Estimating causal effects in observational studies using electronic health data: challenges and (some) solutions. *Egems 1*(3).

Tampuu, A., T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente (2017). Multiagent cooperation and competition with deep reinforcement learning. *PloS one 12*(4), e0172395.

Tavakoli, A., F. Pardo, and P. Kormushev (2017). Action branching architectures for deep reinforcement learning. *arXiv preprint arXiv:1711.08946*.

Team, R. C. (2013). R core team. r: A language and environment for statistical computing. *Foundation for Statistical Computing*.

Tian, L., A. A. Alizadeh, A. J. Gentles, and R. Tibshirani (2014). A simple method for estimating interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association 109*(508), 1517–1532.

Tsiatis, A. A. (2019). *Dynamic Treatment Regimes: Statistical Methods for Precision Medicine*. CRC press.

Tu, W., B. S. Decker, Z. He, B. L. Erdel, G. J. Eckert, R. N. Hellman, M. D. Murray, J. A. Oates, and J. H. Pratt (2016). Triamterene enhances the blood pressure lowering effect of hydrochlorothiazide in patients with hypertension. *Journal of general internal medicine 31*(1), 30–36.

Tu, W., G. J. Eckert, B. S. Decker, and J. Howard Pratt (2017). Varying influences of aldosterone on the plasma potassium concentration in blacks and whites. *American journal of hypertension 30*(5), 490–494.

Van Hasselt, H., A. Guez, and D. Silver (2015). Deep reinforcement learning with double q-learning. *arXiv preprint arXiv:1509.06461*.

Vecerik, M., T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller (2017). Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*.

Wang, L., W. Zhang, X. He, and H. Zha (2018). Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the*

24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2447–2456.

Wang, Z., T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas (2016). Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pp. 1995–2003. PMLR.

Watkins, C. J. and P. Dayan (1992). Q-learning. *Machine learning 8*(3-4), 279–292.

Watkins, C. J. C. H. (1989). Learning from delayed rewards.

Xiao, W., H. H. Zhang, and W. Lu (2019). Robust regression for optimal individualized treatment rules. *Statistics in medicine 38*(11), 2059–2073.

Xiong, S., B. Dai, J. Huling, and P. Z. Qian (2016). Orthogonalizing em: A design-based least squares algorithm. *Technometrics 58*(3), 285–293.

Yu, C., J. Liu, and S. Nemati (2019). Reinforcement learning in healthcare: A survey. *arXiv preprint arXiv:1908.08796*.

Yu, C., G. Ren, and Y. Dong (2020). Supervised-actor-critic reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units. *BMC medical informatics and decision making 20*(3), 1–8.

Yu, T., G. Thomas, L. Yu, S. Ermon, J. Zou, S. Levine, C. Finn, and T. Ma (2020). Mopo: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*.

Yuan, M. and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(1), 49–67.

Zhang, B., A. A. Tsiatis, E. B. Laber, and M. Davidian (2012). A robust method for estimating optimal treatment regimes. *Biometrics 68*(4), 1010–1018.

Zhang, W., J. Li, and L. Liu (2020). A unified survey on treatment effect heterogeneity modeling and uplift modeling. *arXiv preprint arXiv:2007.12769*.

Zhou, S., X. Shen, D. Wolfe, et al. (1998). Local asymptotics for regression splines and confidence regions. *The annals of statistics 26*(5), 1760–1782.

Curriculum Vitae

Ruohong Li

Education

- Ph.D. in Biostatistics, Indiana University-Purdue University Indianapolis, Indianapolis, IN, May 2021 (minor in Computer & Information Science)
- B.S. in Information and Computing Science, Beijing Institute of Technology, Beijing, China, May 2016

Working experiences

- Summer Intern, Microsoft, Redmond, WA, 2020
- Summer Intern, Amazon, Palo Alto, CA, 2019