Variable and Model Selection for Propensity Score Estimators in Causal Inference

by

Cheng Ju

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Biostatistics

and the Designated Emphasis

in

Computational and Data Science and Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Mark van der Laan, Chair
Professor Antoine Chambaz
Professor Alan Hubbard
Professor Haiyan Huang

Spring 2018

Variable and Model Selection for Propensity Score Estimators in Causal Inference

Abstract

Variable and Model Selection for Propensity Score Estimators in Causal Inference

by

Cheng Ju

Doctor of Philosophy in Biostatistics

University of California, Berkeley

Professor Mark van der Laan, Chair

Robust inference of a low-dimensional parameter in a large semi-parametric model relies on external estimators of infinite-dimensional features of the distribution of the data. Typically, only one of the latter is optimized for the sake of constructing a well behaved estimator of the low-dimensional parameter of interest. Optimizing more than one of them for the sake of achieving a better bias-variance trade-off in the estimation of the parameter of interest is the core idea driving the general template of the collaborative targeted minimum loss-based estimation (C-TMLE) procedure. In this dissertation, we first resolves the computational issue in the widely-used greedy variable selection C-TMLE. Then we further investigate how to extend the discrete, variable selection C-TMLE for a more general model selection purpose.

Chapter 1 begins by introducing the framework of causal inference in observational studies. We introduce the non-parametric structural equation model for modeling the data generating distribution. We briefly review the targeted minimum loss-based estimation (TMLE). We also introduce the general template of C-TMLE and its greedy-search variable selection version.

In chapter 2, we propose the template for scalable variable selection C-TMLEs to overcome the computational burden in the greedy variable selection C-TMLE. The original instantiation of the C-TMLE template can be presented as a greedy forward stepwise C-TMLE algorithm. It does not scale well when the number p of covariates increases drastically. This motivates the introduction of a novel instantiation of the C-TMLE template where the covariates are pre-ordered. Its time complexity is $\mathcal{O}(p)$ as opposed to the original $\mathcal{O}(p^2)$, a remarkable gain. We propose two pre-ordering strategies and suggest a rule of thumb to develop other meaningful strategies. Because it is usually unclear a priori which pre-ordering strategy to choose, we also introduce another instantiation called SL-C-TMLE algorithm that enables the data-driven choice of the better pre-ordering strategy given the problem at hand. Its time complexity is $\mathcal{O}(p)$ as well. The computational burden and relative performance of these algorithms were compared in simulation studies involving fully synthetic data or partially synthetic data based on a real world large electronic health database; and in

analyses of three real, large electronic health databases. In all analyses involving electronic health databases, the greedy C-TMLE algorithm is unacceptably slow. Simulation studies seem to indicate that our scalable C-TMLE and SL-C-TMLE algorithms work well.

In chapter 3, we extend C-TMLE to a more general model selection problem: we apply C-TMLE to select from a set of continuously-indexed nuisance parameter (the propensity score, PS) estimators. The propensity score models have traditionally been selected based on the goodness-of-fit for the treatment mechanism itself, without consideration of the causal parameter of interest. In contrast, the C-TMLE takes into account information on the causal parameter of interest when selecting a PS model. This "collaborative learning" considers variable associations with both treatment and outcome when selecting a PS model in order to minimize a bias-variance trade off in the estimated treatment effect. In this study, we introduce a novel approach for collaborative model selection when using the LASSO estimator for PS estimation in high-dimensional covariate settings. To demonstrate the importance of selecting the PS model collaboratively, we designed quasi-experiments based on a real electronic healthcare database, where only the potential outcomes were manually generated, and the treatment and baseline covariates remained unchanged. Results showed that the C-TMLE algorithm outperformed other competing estimators for both point estimation and confidence interval coverage. In addition, the PS model selected by C-TMLE could be applied to other PS-based estimators, which also resulted in substantive improvement for both point estimation and confidence interval coverage. We illustrate the discussed concepts through an empirical example comparing the effects of non-selective Nonsteroidal anti-inflammatory drugs with selective COX-2 inhibitors on gastrointestinal complications in a population of Medicare beneficiaries.

In chapter 4, we propose using C-TMLE to adaptively truncated the propensity score when there exist practical positivity violations. The positivity assumption, or the experimental treatment assignment (ETA) assumption, is important for identifiability in causal inference. Even if the positivity assumption holds, practical violations of this assumption may jeopardize the finite sample performance of the causal estimator. One of the consequences of practical violations of the positivity assumption is extreme values in the estimated propensity score. A common practice to address this issue is truncating the PS estimate when constructing PS-based estimators. In this study, we propose a novel adaptive truncation method, Positivity-C-TMLE, based on the C-TMLE methodology. We further show how to construct a robust confidence interval by a targeted variance estimator. We demonstrate the outstanding performance of our novel approach in a variety of simulations by comparing it with other commonly studied estimators, for both point estimation and confidence interval coverage. Results show that by adaptively truncating the estimated PS with a more targeted objective function, the Positivity-C-TMLE estimator achieves the best performance for both point estimation and confidence interval coverage among all estimators considered.

The code for all the variations of C-TMLE in this dissertation are publicly available in the *ctmle* R package.

To my family and friends.

Contents

List of Figures

List of Tables

Acknowledgments

It is my fortune to have Professor Mark van der Laan as my advisor. I will always be grateful to him for all he has taught me, and all the opportunities he made available to me. Because of him, my past four years have been the most intellectually stimulating and rewarding period of my life.

I am also grateful for the chance to learn from so many amazing professors and colleagues at Berkeley. I have learned so much from conversations, discussions, and collaborations with them. I am grateful to my collaborators in my past and on-going projects. I am also grateful to David Benkeser, Susan Gruber, Sam Lendle, and Eric Polley, and for their reliable software, which played important role in my research. I especially appreciate my committee members, Antoine Chambaz, Alan Hubbard, and Haiyan Huang, for their insightful guidance on my dissertation.

Part of the material presented here have been published elsewhere. Chapters 1 and 2 appeared in *Statistical Methods in Medical Research* as “Scalable Collaborative Targeted Learning for High-Dimensional Data”, co-authored with Susan Gruber, Samuel D Lendle, Antoine Chambaz, Jessica M Franklin, Richard Wyss, Sebastian Schneeweiss, and Mark van der Laan. Material from chapter 3, co-authored with Richard Wyss, Jessica M Franklin, Sebastian Schneeweiss, Jenny Haggstrom, and Mark van der Laan, is included in *Statistical Methods in Medical Research* as “Collaborative-controlled LASSO for Constructing Propensity Score-based Estimators in High-Dimensional Data”. The chapter 4 describing using collaborative targeted maximum likelihood estimation for the adaptive propensity score truncation, co-authored with Joshua Schwab and Mark van der Laan, is currently under review at *Statistical Methods in Medical Research.* I sincerely thank each of these individuals and publishers for their contributions to the work and permission to include it here.

Chapter 1

Background

1.1 Estimating the Average Treatment Effect in Observational Studies

We mainly consider the problem of estimating the ATE in an observational study where we observe on each experimental unit: a collection of p baseline covariates, W; a binary treatment indicator, A; a binary or continuous $(0,1)$-valued outcome of interest, Y. We use $O_i = (W_i, A_i, Y_i)$ to represent the i-th observation from the unknown observed data distribution P_0, and assume that $O_1, \ldots, O_n$ are independent. The parameter of interest is defined as

$$\Psi(P_0) = \mathbb{E}_0[\mathbb{E}_0(Y|A=1,W) - \mathbb{E}_0(Y|A=0,W)].$$

The ATE enjoys a causal interpretation under the non-parametric structural equation model (NPSEM) given by:

$$\begin{cases} W = f_W(U_W) \\ A = f_A(W, U_A) \\ Y = f_Y(A, W, U_Y) \end{cases},$$

where f_W, f_A and f_Y are deterministic functions and U_W, U_A, U_Y are background (exogenous) variables. The potential outcome under exposure level $a \in \{0,1\}$ can be obtained by substituting a for A in the third equality: $Y_a = f_Y(a, W, U_Y)$. Note that $Y = Y_A$ (this is known as the "consistency" assumption). If we are willing to assume that *(i)* A is conditionally independent of (Y_1, Y_0) given W (this is known as the "no unmeasured confounders" assumption) and *(ii)* $0 < P(A=1|W) < 1$ almost everywhere (this is known as the "positivity" assumption), then $\Psi(P_0)$ satisfies $\Psi(P_0) = \mathbb{E}_0(Y_1 - Y_0)$.

For future use, we introduce the propensity score (PS), defined as the conditional probability of receiving treatment, and define $g_0(a, W) \equiv P_0(A=a|W)$ for both $a = 0, 1$. We also introduce the conditional mean of the outcome: $\bar{Q}_0(A, W) = \mathbb{E}_0(Y|A, W)$. In the remainder of this article, $g_n(a, W)$ and $\bar{Q}_n(A, W)$ denote estimators of $g_0(a, W)$ and $\bar{Q}_0(A, W)$.

1.2 Targeted Maximum Likelihood Estimation for the ATE

We are primarily interested in double robust (DR, which also stands for double robustness) estimators of $\Psi(P_0)$. An estimator of $\Psi(P_0)$ is said to be DR if it is consistent if either $\bar{Q}_0$ or g_0 is consistently estimated. In addition, an estimator of $\Psi(P_0)$ is said to be efficient if it satisfies a central limit theorem with a limit variance which equals the second moment under P_0 of the so called efficient influence curve (EIC) at P_0. The EIC for the ATE parameter is given by

$$D^*(\bar{Q}_0, g_0)(O) = H_0(A,W)(Y - \bar{Q}_0(A,W)) + \bar{Q}_0(1,W) - \bar{Q}_0(0,W) - \Psi(P_0),$$

where $H_0(A,W) = A/g_0(1,W) - (1-A)/g_0(0,W)$. The notation $D^*(\bar{Q}_0, g_0)$ is slightly misleading: it suggests that $\bar{Q}_0$ and g_0 fully characterize $D^*(\bar{Q}_0, g_0)$ whereas the marginal distribution $P_{0,W}$ of W under P_0, which appears in $\Psi(P_0)$, is also needed. We nevertheless keep the notation as is for brevity. We refer the reader to [3] for details about efficient influence curves.

More generally, for every valid distribution P of $O = (W, A, Y)$ such that *(i)* the conditional expectation of Y given (A, W) equals $\bar{Q}(A, W)$ and the conditional probability that $A = a$ given W equals $g(a, W)$, and *(ii)* $0 < g(1, W) < 1$ almost surely, we denote

$$D^*(\bar{Q}, g)(O) = H_g(A,W)(Y - \bar{Q}(A,W)) + \bar{Q}(1,W) - \bar{Q}(0,W) - \Psi(P)$$

where $H_g(A,W) = A/g(1,W) - (1-A)/g(0,W)$. The augmented inverse probability of treatment weighted estimator (A-IPTW, or so called "DR IPTW"; [60, 58, 37]) and TMLE [39, 38] are two well studied DR estimators. Taking the estimation of the ATE as an example, A-IPTW estimates $\Psi(P_0)$ by solving the EIC equation directly. Given two estimators $\bar{Q}_n$ and g_n of $\bar{Q}_0$ and g_0, setting

$$H_{g_n}(A,W) = A/g_n(1,W) - (1-A)/g_n(0,W), \tag{1.1}$$

and solving (in ψ)

$$0 = \sum_{i=1}^{n} \big(H_{g_n}(A_i,W_i)(Y_i - \bar{Q}_n(A_i,W_i)) + \bar{Q}_n(1,W_i) - \bar{Q}_n(0,W_i) - \psi\big)$$

yield the A-IPTW estimator

$$\psi_n^{\text{A-IPTW}} = \frac{1}{n}\sum_{i=1}^{n} \big(H_{g_n}(A_i,W_i)(Y_i - \bar{Q}_n(A_i,W_i)) + \bar{Q}_n(1,W_i) - \bar{Q}_n(0,W_i)\big).$$

It is worth noting that the A-IPTW estimator is not a substitution estimator: it cannot be written as the value of Ψ at a particular P. The A-IPTW may thus sometimes take values outside of the parameter space $[0, 1]$ where $\Psi(P_0)$ is known to live. On the contrary, an instantiation of the TMLE template yields a substitution estimator which, by construction, belongs to $[0, 1]$. This is a desirable property. For instance, a TMLE estimator can be constructed by applying the TMLE algorithm below (which incorporates the negative log-likelihood loss function and logistic fluctuation; see comment below).

I **Estimating $\bar{Q}_0$.** Derive an initial estimator $\bar{Q}_n^0$ of $\bar{Q}_0$.

II **Estimating g_0.** Derive an estimator g_n of g_0.

III **Building the so called "clever covariate".** Define $H_n(A, W)$ as in (1.1).

IV **"Fluctuating" the initial estimator.** Fit the logistic regression of Y on $H_n(A, W)$ with no intercept, using $\text{logit}(\bar{Q}_n^0(A_i, W_i))$ as i-specific offset/intercept. This yields a minimum loss estimator ϵ_n. Update the initial estimator $\bar{Q}_n^0$ into $\bar{Q}_n^*$ given by $\bar{Q}_n^*(A, W) =$

$$\text{expit}(\text{logit}(\bar{Q}_n^0(A, W)) + \epsilon_n H_n(A, W)). \tag{1.2}$$

V **C**onstructing the TMLE. Evaluate

$$\psi_n^{\text{TMLE}} = \frac{1}{n} \sum_{i=1}^{n} (\bar{Q}_n^*(1, W_i) - \bar{Q}_n^*(0, W_i)). \tag{1.3}$$

In steps I and II, it is highly recommended to avoid making parametric assumptions, as any parametric model is likely mis-specified. Relying on SL [36] is a good option. Step IV aims to reduce bias in the estimation of $\Psi(P_0)$ by enhancing the initial estimator derived from $\bar{Q}_n^0$ and the marginal empirical distribution of W as an estimator of its counterpart under P_0. It is dubbed a "fluctuation" step because it consists, here, in *(i)* building a parametric model through $\bar{Q}_n^0$ and *(ii)* finding the optimal fluctuation of $\bar{Q}_n^0$ in it w.r.t. the chosen loss function. In practice, bounded continuous outcomes and binary outcomes are fluctuated on the logit scale (hence the expression "logistic fluctuation") to ensure that bounds on the model space are respected [15].

In the context of the above TMLE algorithm, step IV consists in minimizing $\epsilon \mapsto L_n(\bar{Q}_n^0(\epsilon))$ over $\mathbb{R}$, where

$$L_n(\bar{Q}_n^0(\epsilon)) = \sum_{i=1}^{n} \Big(Y_i \log(\bar{Q}_n^0(\epsilon)(A_i, W_i)) + (1 - Y_i) \log(1 - \bar{Q}_n^0(\epsilon)(A_i, W_i)) \Big) \tag{1.4}$$

is the empirical loss of $\bar{Q}_n^0(\epsilon)$ given by (1.2) with ϵ substituted for ϵ_n. Moreover, the fluctuation in step IV is made in such a way that the EIC equation is solved: $\sum_i D^*(\bar{Q}_n^*, g_n)(O_i) = 0$, which justifies why $\bar{Q}_n^*$ is said to be "targeted" toward $\Psi(P_0)$. This is the key to the TMLE estimator being DR and asymptotically efficient under regularity conditions [38].

Standard errors and confidence intervals (CIs) can be computed based on the variance of the influence curve. Proofs and technical details are available in the literature [39, 38].

1.3 Collaborative Targeted Maximum Likelihood Estimation for the ATE

When implementing an instantiation of the TMLE template, one relies on a single external estimate of the nuisance parameter, g_0 in the ATE example (see step 2 in Section 1.2). In contrast, an instantiation of the C-TMLE template involves constructing a series of nuisance parameter estimates and corresponding TMLE estimators using these estimates in the targeting step. Section 1.3 presents the C-TMLE general template and Section 1.3 its first instantiation, called the greedy C-TMLE algorithm.

The C-TMLE Template

When the ATE is the parameter of interest, the C-TMLE template can be summarized recursively like this (see Algorithm 1 for a high-level algorithmic presentation).

1. **I**nitialization. Build an initial triplet $(g_{n,0}, \bar{Q}_{n,0}, \bar{Q}^*_{n,0})$ where $g_{n,0}$ estimates g_0 and $\bar{Q}_{n,0} = \bar{Q}^0_n$ and $\bar{Q}^*_{n,0}$ estimate $\bar{Q}_0$, the latter estimator being targeted toward $\Psi(P_0)$ for instance as in step IV of the TMLE algorithm presented in Section 1.2.

   Suppose that k triplets $(g_{n,0}, \bar{Q}_{n,0}, \bar{Q}^*_{n,0})$, . . ., $(g_{n,k-1}, \bar{Q}_{n,k-1}, \bar{Q}^*_{n,k-1})$ have been built.

2. **D**eriving the next triplet.

   a) Tentatively set $\bar{Q}_{n,k} = \bar{Q}_{n,k-1}$.

   b) Derive candidate estimators $g^j_{n,k}$ of g_0 $(1 \leq j \leq J_{n,k})$ so that the empirical fit provided by each $g^j_{n,k}$ is better than that of $g_{n,k-1}$.

   c) For each j, build $\bar{Q}^{j,*}_{n,k}$ by fluctuating $\bar{Q}_{n,k}$ based on $g^j_{n,k}$ as in step IV of the TMLE algorithm presented in Section 1.2 for instance.

   d) Find $\jmath$ such that the empirical loss (see (1.4) in Section 1.2 for an example) of $\bar{Q}^{\jmath,*}_{n,k}$ equals the minimum among the empirical losses of $\bar{Q}^{j,*}_{n,k}$ $(1 \leq j \leq J_{n,k})$, then tentatively set $(g_{n,k}, \bar{Q}_{n,k}, \bar{Q}^*_{n,k}) = (g^\jmath_{n,k}, \bar{Q}_{n,k}, \bar{Q}^{\jmath,*}_{n,k})$.

   e) If the empirical loss of the candidate $\bar{Q}^*_{n,k}$ is smaller than that of $\bar{Q}^*_{n,k-1}$, then accept the candidate triplet.

   f) If the empirical loss of the candidate $\bar{Q}^*_{n,k}$ is larger than that of $\bar{Q}^*_{n,k-1}$, then set $\bar{Q}_{n,k} = \bar{Q}^*_{n,k-1}$, go back to step 2b and carry out steps 2b, 2c, 2d and 2e.

3. **S**electing the best triplet. Once all the triplets have been built, identify the triplet $(g_{n,k_n}, \bar{Q}_{n,k_n}, \bar{Q}^*_{n,k_n})$ that minimizes a cross-validated, loss-based, penalized empirical risk, with the same loss function as that used in step 2c to fluctuate $\bar{Q}_{n,k}$.

4. **C**onstructing the C-TMLE. Evaluate

$$\psi_n^{\text{C-TMLE}} = \frac{1}{n}\sum_{i=1}^{n}(\bar{Q}^*_{n,k_n}(1,W_i) - \bar{Q}^*_{n,k_n}(0,W_i)).$$

As in step 1 of the TMLE instantiation presented in Section 1.2, we recommend relying on SL in step 1 of the above general template of C-TMLE. Two comments are in order regarding step 2. First, to achieve collaborative DR eventually, the sequence of estimators $(g_{n,k} : k)$ derived in steps 2b and 2d should be arranged in such a way that the estimator becomes increasingly nonparametric, with asymptotic bias and variance respectively decreasing and increasing, and so that $g_{n,k}$ converges (in k) to a consistent estimator of g_0 [38]. One could for instance rely on a nested sequence of models, see Section 1.3. By doing so, the empirical fit for g_0 improves as k increases [38, 16]. Second, if step 2f is carried out, then it necessarily holds that the empirical risk of $\bar{Q}^*_{n,k}$ is smaller than that of $\bar{Q}^*_{n,k-1}$ the second time step 2e is undertaken, so the candidate triplet is accepted. In step 3, k_n is formally defined as

$$k_n = \arg\min_k \left\{\text{cv}Risk_k + \text{cv}Var_k + n \times \text{cv}Bias_k^2\right\}$$

where $\text{cv}Risk_k$, $\text{cv}Var_k$, $\text{cv}Bias_k$ are respectively given by

$$\sum_{v=1}^{V}\sum_{i\in \text{V}al(v)} \text{loss}(\bar{Q}^*_{n,k}(P^0_{nv}))(O_i),$$

$$\sum_{v=1}^{V}\sum_{i\in \text{V}al(v)} D^*(\bar{Q}^*_{n,k}(P^0_{nv}), g_{n,k}(P^0_{n,v}))(O_i)^2,$$

$$\frac{1}{V}\sum_{v=1}^{V}[\Psi(\bar{Q}^*_{n,k}(P^0_{nv})) - \Psi(\bar{Q}^*_{n,k}(P_n))]$$

where $\Psi(\bar{Q}^*_{n,k}(P^0_{nv}))$ and $\Psi(\bar{Q}^*_{n,k}(P_n))$ are shorthand notation for (1.3) with $\bar{Q}^*_{n,k}(P^0_{nv})$ and $\bar{Q}^*_{n,k}(P_n)$ substituted for $\bar{Q}^*_n$, and where loss is the loss function used in step 2c to fluctuate $\bar{Q}_{n,k}$. That could be for instance the least-square loss function, in which case $\text{cv}Risk_k$ would equal

$$\text{cv}RSS_k = \sum_{v=1}^{V}\sum_{i\in \text{V}al(v)} (Y_i - \bar{Q}^*_{n,k}(P^0_{nv})(W_i, A_i))^2.$$

In the two previous displays, $\text{V}al(v)$ is the set of indices of observations used for validation in the v-th fold, P^0_{nv} is the empirical distribution of the observations indexed by $i \notin \text{V}al(v)$, P_n is the empirical distribution of the whole data set, and $Z(P^0_{nv})$ (respectively, $Z(P_n)$) means that Z is fitted using P^0_{nv} (respectively, P_n). The penalization terms $\text{cv}Var_k$ and $\text{cv}Bias_k$ robustify the finite sample performance when the positivity assumption is violated [40].

The C-TMLE eventually defined in step 4 inherits all the properties of the plain TMLE estimator defined in (1.3) [40]. It is DR and asymptotically efficient under appropriate regularity conditions. [50] discusses and compares TMLE and C-TMLE with other DR estimators, including A-IPTW.

Section 1.3 presents the first instantiation of the C-TMLE general template.

Algorithm 1 General Template of C-TMLE

```
1: Construct an initial estimator $\bar{Q}_n^0$ for $\bar{Q}_0$.
2: Create candidate $\bar{Q}_{n,k}^*$, using different estimators $g_{n,k}$ of $g_0$, such that the empirical risks
   of $\bar{Q}_{n,k}^*$ and $g_{n,k}$ are decreasing in $k$.
3: Select the best candidate $\bar{Q}_n^* = \bar{Q}_{n,k_n}^*$ using loss-based cross-validation, with the same
   loss function as in the TMLE targeting step.
```

The Greedy C-TMLE Algorithm

We refer to the first instantiation of the C-TMLE template as the greedy C-TMLE algorithm. It uses a forward selection algorithm to build the sequence of estimators of g_0 based on a nested sequence of models for g_0 that we call PS models. Let us describe the algorithm in the case that W consists of p covariates. The steps we refer to are those of the C-TMLE template of Section 1.3.

The construction of $g_{n,0}$ in step 1 relies on the PS model defined as the one-dimensional logistic model with only an intercept (the "intercept model"). Therefore, if the PS model is fitted based on P_n, then $g_{n,0}$ is given by $g_{n,0}(1|W) = 1 - g_{n,0}(0|W) = P_n(A = 1)$. The derivation of $\bar{Q}_{n,0}^*$ from $\bar{Q}_{n,0}$ and $g_{n,0}$ in step 1 is then carried out by fitting the logistic regression of Y on $H_{g_{n,0}}(A, W)$ with i-specific offset/intercept $\text{logit}(\bar{Q}_{n,0}(A_i, W_i))$, where

$$H_{g_{n,k}}(A, W) = A/g_{n,k}(1|W) - (1 - A)/g_{n,k}(0|W), \tag{1.5}$$

leading to

$$\text{logit}(\bar{Q}_{n,k}^*(A, W)) = \text{logit}(\bar{Q}_{n,k}(A, W)) + \epsilon_k H_{g_{n,k}}(A, W) \tag{1.6}$$

(with $k = 0$). We denote by $\mathcal{L}_0$ the empirical risk of $\bar{Q}_{n,0}^*$ w.r.t. the negative log-likelihood function $\mathcal{L}$.

Assume that $g_{n,1}, \ldots, g_{n,k-1}$ have already been derived by fitting PS models for g_0 where the ℓth PS model is included (as a set) in the $(\ell + 1)$th PS model because in the latter A is regressed on an intercept, the same $(\ell - 1)$ covariates as in the former *and* on an additional covariate (for each $1 \leq \ell \leq k$). To construct the $(k+1)$th PS model in step 2b, each covariate W_j ($1 \leq j \leq p$ such that W_j has not been included yet) is considered in turn as a candidate additional covariate added to the kth PS model to form the $(k + 1)$th PS model. By fitting the corresponding candidate $(k+1)$th PS model, we obtain a candidate $g_{n,k}^j$. Step 2c consists in defining the corresponding $H_{g_{n,k}^j}$ and $\bar{Q}_{n,k}^{j,*}$ as in (1.5) and (1.6). To carry out step 2d, let the empirical risk of $\bar{Q}_{n,k}^{\jmath,*}$ w.r.t. $\mathcal{L}$ be the smallest of the empirical risks of $\bar{Q}_{n,k}^{j,*}$ (for all considered js), let the $(k+1)$th PS model be the one where $W_\jmath$ is added to the kth PS model, and set $(g_{n,k}, \bar{Q}_{n,k}, \bar{Q}^*_{n,k}) = (g^\jmath_{n,k}, \bar{Q}_{n,k-1}, \bar{Q}^{\jmath,*}_{n,k})$. Let $\mathcal{L}_k$ be the empirical risk of $\bar{Q}^*_{n,k}$ w.r.t. $\mathcal{L}$. In step 2e, we assess whether $\mathcal{L}_k \leq \mathcal{L}_{k-1}$ or not. If the inequality is met, then the candidate triplet is accepted. Otherwise, we reset $\bar{Q}_{n,k} = \bar{Q}^*_{n,k-1}$ and repeat steps 2c and 2d. It is then guaranteed that the empirical risk of $\bar{Q}^*_{n,k}$ w.r.t. $\mathcal{L}$ is smaller than $\mathcal{L}_{k-1}$, and the candidate triplet is accepted.

This forward stepwise procedure is carried out recursively until all p covariates have been incorporated into the PS model for g_0. In the discussed setting, choosing the first covariate requires p comparisons, choosing the second covariate requires $(p-1)$ comparisons and so on.

Fitting a PS model to derive an estimator $g_{n,k}$ and fluctuating a current $\bar{Q}_{n,k}$ based on the resulting $H_{g_{n,k}}$ does not take much computational time. We consider this time as the time unit, and can thus claim that the time complexity w.r.t. p of the greedy C-TMLE algorithm is $\mathcal{O}(\sum_{k=1}^{p} k) = \mathcal{O}(p^2)$ time units (the $\mathcal{O}$ accounts for the cross-validation).

Chapter 2

Scalable Collaborative Targeted Learning for Variable Selection in High-dimensional Data

2.1 Introduction

The general template of collaborative double robust targeted minimum loss-based estimation (C-TMLE; "C-TMLE template" for short) builds upon the targeted minimum loss-based estimation (TMLE) template [38, 40]. Both the TMLE and C-TMLE templates can be viewed as meta-algorithms which map a set of user-supplied choices/hyper-parameters (e.g., parameter of interest, loss function, submodels) into a specific machine-learning algorithm for estimation, that we call an instantiation of the template.

Constructing a TMLE or a C-TMLE involves the estimation of a nuisance parameter, typically an infinite-dimensional feature of the distribution of the data. For a plain TMLE estimator, the estimation of the nuisance parameter is addressed as an independent statistical task. In the C-TMLE template, on the contrary, the estimation of the nuisance parameter is optimized to provide a better bias-variance trade-off in the inference of the targeted parameter. The C-TMLE template has been successfully applied in a variety of areas, from survival analysis [69], to the study of gene association [74] and longitudinal data structures [68] to name just a few.

In the original instantiation of the C-TMLE template of [40], that we henceforth call "the greedy C-TMLE algorithm", the estimation of the nuisance parameter aiming for a better bias-variance trade-off is conducted in two steps. First, a greedy forward stepwise selection procedure is implemented to construct a sequence of candidate estimators of the nuisance parameter derived by fitting a nested sequence of models. Second, cross-validation is used to select the candidate from this sequence which minimizes a criterion that incorporates a measure of bias and variance with respect to (w.r.t) the targeted parameter (the algorithm is described in Section 1.3). The authors show that the greedy C-TMLE algorithm exhibits superior relative performance in analyses of sparse data, at the cost of an increase in time complexity. For instance, in a problem with p baseline covariates, one would construct and select from p candidate estimators of the nuisance parameter, yielding a time complexity of order $\mathcal{O}(p^2)$. Despite a criterion for early termination, the algorithm does not scale to large-scale and high-dimensional data. The aim of this article is to develop novel C-TMLE algorithms that overcome these serious practical limitations without compromising finite sample or asymptotic performance.

We propose two such "scalable C-TMLE algorithms". They replace the greedy search at each step by an easily computed data adaptive pre-ordering of the candidate estimators of the nuisance parameter. They include a data adaptive, early stopping rule that further reduces computational time without sacrificing statistical performance. In the aforementioned problem with p baseline covariates where the time complexity of the greedy C-TMLE algorithm was of order $\mathcal{O}(p^2)$, those of the two novel scalable C-TMLE algorithms is of order $\mathcal{O}(p)$.

Because one may be reluctant to specify a single a priori pre-ordering of the candidate estimators of the nuisance parameter, we also introduce a SL-C-TMLE algorithm. It selects the best pre-ordering from a set of ordering strategies by super learning (SL) [36]. SL is an example of ensemble learning methodology which builds a meta-algorithm for estimation out of a collection of individual, competing algorithms of estimation, relying on oracle properties of cross-validation.

We focus on the estimation of the average (causal) treatment effect (ATE). It is not difficult to generalize our scalable C-TMLE algorithms to other estimation problems, by simply replacing the greedy search part in the corresponding greedy C-TMLE algorithm with the scalable version when building the sequence of candidate estimates, while leaving other building blocks unchanged.

The performance of the two scalable C-TMLE and SL-C-TMLE algorithms are compared with those of competing, well established estimation methods: G-computation [56], inverse probability of treatment weighting (IPTW) [22, 57], augmented inverse probability of treatment weighted estimator (A-IPTW) [54, 55, 61]. Results from unadjusted regression estimation of a point treatment effect are also provided to illustrate the level of bias due to confounding.

This chapter is organized as follows. Section 2.2 introduces the two proposed pre-ordered scalable C-TMLE algorithms, and SL-C-TMLE algorithm. Sections 2.3 and 2.3 present the results of simulation studies (based on fully or partially synthetic data, respectively) comparing the C-TMLE and SL-C-TMLE estimators with other common estimators. Section 2.4 presents and compares the empirical processing time of C-TMLE algorithms for different sample sizes and numbers of candidate estimators of the nuisance parameter. Section 2.5 compares the performance of the new C-TMLEs with standard TMLE on three real data sets. Section 2.6 is a closing discussion.

2.2 Scalable C-TMLE Algorithms

Now that we have introduced the background on C-TMLE, we are in a position to present our scalable C-TMLE algorithm. Section 2.2 summarizes the philosophy of the scalable C-TMLE algorithm, which hinges on a data adaptively determined pre-ordering of the baseline covariates. Sections 2.2 and 2.2 present two such pre-ordering strategies. Section 2.2 discusses what properties a pre-ordering strategy should satisfy. Section 2.2 proposes a discrete Super Learner-based model selection procedure to select among a set of scalable C-TMLE estimators, which is itself a scalable C-TMLE algorithm. Finally, Section 2.2 sketches how to adapt scalable C-TMLEs to other estimation problems, with the example of the relative risk (RR).

Outline

A $\mathcal{O}(p^2)$ time complexity when there are p covariates is unsatisfactory for large scale and high-dimensional data, a situation which is increasingly common in health care research. For example, the high-dimensional propensity score (hdPS) is a method to extract information from electronic medical claims data that produces hundreds or even thousands of candidate covariates, increasing the dimension of the data dramatically [66].

In order to make it possible to apply C-TMLE algorithms to such data sets, we propose to add a new pre-ordering procedure after the initial estimation of $\bar{Q}_0$ and before the stepwise construction of the candidate $\bar{Q}^*_{n,0}, \bar{Q}^*_{n,1}, \ldots, \bar{Q}^*_{n,k}, \ldots$. We present two pre-ordering procedures in Sections 2.2 . By imposing an ordering over the covariates, only one covariate is eligible for inclusion in the PS model at each step when constructing the next candidate $\bar{Q}^*_{n,k}$. In other words, $J_{n,k}$ equals 1 in steps 2b and 2c, and $\jmath = j = 1$ in step 2d of the C-TMLE general template presented in Section 1.3. Therefore, the computational time of a scalable C-TMLE algorithm w.r.t. p is $\mathcal{O}(\sum_{i=1}^{p} 1) = \mathcal{O}(p)$ time units (the $\mathcal{O}$ accounts for the cross-validation).

Logistic Pre-Ordering Strategy

The logistic pre-ordering procedure is similar to step 2 of the C-TMLE general template specialized to the greedy C-TMLE algorithm of Section 1.3. However, instead of selecting one single covariate before going on, we use the empirical losses w.r.t. $\mathcal{L}$ to order the covariates by how much they can improve the predictive performance of $\bar{Q}^0_n$ (or, *heuristically, by their ability to reduce bias).* More specifically, for each covariate W_k ($1 \leq k \leq p$), we construct an estimator $g_{n,k}$ of the conditional distribution of A given W_k only (one might also add W_k to a fixed baseline model); we define a clever covariate as in (1.5) using $g_{n,k}$ and fluctuate $\bar{Q}^0_n$ as in (1.6); we compute the empirical loss of the resulting $\bar{Q}^*_{n,k}$ w.r.t. $\mathcal{L}$, yielding $\mathcal{L}_k$. Finally, the covariates are ranked by increasing values of the empirical loss. This is summarized in Algorithm 2.

Algorithm 2 Logistic Pre-Ordering Algorithm

```
for each covariate $W_k$ in $W$ do
    Construct an estimator $g_{n,k}$ of $g_0$ using a logistic model with $W_k$ as predictor.
    Define a clever covariate $H_{g_{n,k}}(A, W_k)$ as in (1.5).
    Fit $\epsilon_k$ by regressing $Y$ on $H_{g_{n,k}}(A, W_k)$ with $i$-specific offset/intercept logit$(\bar{Q}_n^0(A_i, W_{k,i}))$.
    Define $\bar{Q}_{n,k}^*$ as in (1.6).
    Compute the empirical loss $\mathcal{L}_k$ w.r.t. $\mathcal{L}$.
end for
Rank the covariates by increasing $\mathcal{L}_k$.
```

Partial Correlation Pre-Ordering Strategy

In the greedy C-TMLE algorithm described in Section 1.3, once k covariates have already been selected, the $(k+1)$th is that remaining covariate which provides the largest reduction in the empirical loss w.r.t. $\mathcal{L}$. Heuristically, the $(k+1)$th covariate is the one that best explains the residual between Y and $\bar{Q}_{n,k}^*$. Drawing on this idea, the partial correlation pre-ordering procedure ranks the p covariates based on how each of them is correlated with the residual between Y and *the initial* $\bar{Q}_n^0$ within strata of A. This second strategy is less computationally demanding than the previous one because there is no need to fit any regression models, all one has to do is merely to estimate p partial correlation coefficients.

Let $\rho(X_1, X_2)$ denote the Pearson correlation coefficient between X_1 and X_2. Recall that the partial correlation $\rho(X_1, X_2|X_3)$ between X_1 and X_2 given X_3 is defined as the correlation coefficient between the residuals R_{X_1} and R_{X_2} resulting from the linear regression of X_1 on X_3 and of X_2 on X_3, respectively [19]. For each $1 \leq k \leq p$, we introduce $R = Y - \bar{Q}_n^0(A, W)$,

$$\rho(R, W_k|A) = \frac{\rho(R, W_k) - \rho(R, A) \times \rho(W_k, A)}{\sqrt{(1 - \rho(R, A)^2)(1 - \rho(W_k, A)^2)}}.$$

The partial correlation pre-ordering strategy is summarized in Algorithm 3.

Algorithm 3 Partial Correlation Pre-Ordering Algorithm

```
for each covariate $W_k$ in $W$ do
    Estimate the partial correlation coefficient $\rho(R, W_k|A)$ between $R = (Y - \bar{Q}_n^0(A, W))$ and $W_k$ given $A$.
end for
Rank the covariates based on the absolute value of the estimates of the partial correlation coefficients.
```

Discussion of the Design of Pre-ordering

Sections 2.2 and 2.2 propose two pre-ordering strategies. In general, a rule of thumb for designing a pre-ordering strategy is to rank the covariates based on the impact of each in reducing the residual bias in the target parameter which results from the initial estimator $\bar{Q}_n^0$ of $\bar{Q}_0$. In this light, the logistic ordering of Section 2.2 uses TMLE to reflect the importance of each variable w.r.t. its potential to reduce residual bias. The partial correlation ordering of Section 2.2 ranks the covariates according to the partial correlation of residual of the initial fit and the covariates, conditional on treatment.

Because the rule of thumb considers each covariate in turn separately, it is particularly relevant when the covariates are not too dependent. For example, consider the extreme case where two or more of the covariates are highly correlated and can greatly explain the residual bias in the target parameter. In this scenario, these dependent covariates would *all* be ranked towards the front of the ordering. However, after adjusting for *one* of them, the others would typically be much less helpful for reducing the remaining bias. This redundancy may harm the estimation. In cases where it is computationally feasible, this problem can be avoided by using the greedy search strategy, but many other intermediate strategies can be pursued as well.

Super Learner-Based C-TMLE Algorithm

Here, we explain how to combine several C-TMLE algorithms into one. The combination is based on a Super Learner (SL). Super learning is an ensemble machine learning approach that relies on cross-validation. It has been proven that a SL selector can perform asymptotically as well as an oracle selector under mild assumptions [36, 35, 73].

As hinted at above, a SL-C-TMLE algorithm is an instantiation of an extension of the C-TMLE template. It builds upon several competing C-TMLE algorithms, each relying on a different strategy to construct a sequence of estimators of the nuisance parameter. A SL-C-TMLE algorithm can be designed to select the single best strategy (discrete SL-C-TMLE algorithm), or an optimal combination thereof (ensemble SL-C-TMLE algorithm). A SL-C-TMLE algorithm can include both greedy search and pre-ordering methods. A SL-C-TMLE algorithm is scalable if all of the candidate C-TMLE algorithms in the library are scalable themselves.

We focus on a scalable discrete SL-C-TMLE algorithm that uses cross-validation to choose among candidate scalable (pre-ordered) C-TMLE algorithms. Algorithm 4 describes its steps. Note that a single cross-validation procedure is used to select both the ordering procedure m and the number of covariates k included in the PS model. It is because computational time *is* an issue that we do not rely on a nested cross-validation procedure to select k for each pre-ordering strategy m.

Algorithm 4 Super Learner C-TMLE Algorithm

```
Define $M$ covariates pre-ordering strategies yielding $M$ C-TMLE algorithms
for each pre-ordering strategy $m$ do
    Follow step 2 of Algorithm 1 to create candidate $\bar{Q}^*_{n,m,k}$ for the $m$-th strategy.
end for
The best candidate $\bar{Q}^*_n$ is the minimizer of the cross-validated losses of $\bar{Q}^*_{n,m,k}$ across all
  the $(m,k)$ combinations.
```

The time complexity of the SL-C-TMLE algorithm is of the same order as that of the most complex C-TMLE algorithm considered. So, if only pre-ordering strategies of order $\mathcal{O}(p)$ are considered, then the time complexity w.r.t. p of the SL-C-TMLE algorithm is $\mathcal{O}(p)$ as well (the $\mathcal{O}$ accounts for the cross-validation). Given a constant number of user-supplied strategies, the SL-C-TMLE algorithm remains scalable, with a processing time that is approximately equal to the sum of the times for each strategy.

We compare the pre-ordered C-TMLE algorithms and SL-C-TMLE algorithm with greedy C-TMLE algorithm and other common methods in Sections 2.3 and 2.5.

Extend to Other Estimation Problems

We have claimed that the scalable C-TMLEs presented so far, which are tailored to the estimation of the ATE, can be easily adapted to other estimation problems. Say for instance that the RR is the target parameter: $\Psi'(P_0) = \mathbb{E}_0[\mathbb{E}_0(Y|A=1,W)]/\,\mathbb{E}_0[\mathbb{E}_0(Y|A=0,W)]$. Then it suffices to adapt the targeting step (1.6). We now define two clever covariates

$$\begin{aligned} H^0_{g_{n,k}}(A,W) &= -(1-A)/g_{n,k}(0,W), \\ H^1_{g_{n,k}}(A,W) &= A/g_{n,k}(1,W), \end{aligned}$$

and carry out the regression of Y on $H^0_{g_{n,k}}(A,W)$ and $H^1_{g_{n,k}}(A,W)$ with i-specific offset/intercept $\text{logit}(\bar{Q}_{n,k}(A_i,W_i))$, leading to

$$\text{logit}(\bar{Q}^*_{n,k}(A,W)) = \text{logit}(\bar{Q}_{n,k}(A,W)) + \epsilon^0_k H^0_{g_{n,k}}(A,W) + \epsilon^1_k H^1_{g_{n,k}}(A,W).$$

Finally, $\bar{Q}^*_{n,k}$ yields the TMLE estimator of $\Psi'(P_0)$ given as the ratio

$$\frac{1}{n}\sum_{i=1}^{n} \bar{Q}^*_n(1,W_i)/\frac{1}{n}\sum_{i=1}^{n} \bar{Q}^*_n(0,W_i),$$

see [63] for details.

2.3 Simulation Studies

Simulation Studies on Fully Synthetic Data

We carried out four Monte-Carlo simulation studies to investigate and compare the performance of G-computation (that we call MLE), IPTW, A-IPTW, greedy C-TMLE algorithm and scalable C-TMLE algorithms to estimate the ATE parameter. For each study, we generated $N = 1,000$ Monte-Carlo data sets of size $n = 1,000$. Propensity score estimates were truncated to fall within the range $[0.025, 0.975]$ for all estimators.

Denoting $\bar{Q}_n^0$ and g_n two initial estimators of $\bar{Q}_0$ and g_0, the unadjusted, G-computation/MLE, and IPTW estimators of the ATE parameter are given by (2.1), (2.2) and (2.3):

$$\begin{aligned}
\psi_n^{\text{unadj}} &= \frac{\sum_{i=1}^n A_i Y_i}{\sum_{i=1}^n A_i} - \frac{\sum_{i=1}^n (1-A_i) Y_i}{\sum_{i=1}^n (1-A_i)}, && (2.1)\\
\psi_n^{\text{MLE}} &= \frac{1}{n}\sum_{i=1}^n (Q_n^0(1,W_i) - Q_n^0(0,W_i)), && (2.2)\\
\psi_n^{\text{IPTW}} &= \frac{1}{n}\sum_{i=1}^n \frac{(2A_i-1)Y_i}{g_n(A_i,W_i)}, && (2.3)\\
\psi_n^{\text{A-IPTW}} &= \frac{1}{n}\sum_{i=1}^n \frac{(2A_i-1)}{g_n(A_i|W_i)}(Y_i - Q_n^0(W_i,A_i)) \\
&\quad + \frac{1}{n}\sum_{i=1}^n (Q_n^0(1,W_i) - Q_n^0(0,W_i)). && (2.4)
\end{aligned}$$

The A-IPTW and TMLE estimators were presented in Section 1.2. The estimators yielded by the C-TMLE and scalable C-TMLE algorithms were presented in Section 1.3 .

In all simulation studies, the definitions of the TMLE (1.3), IPTW (2.3) and A-IPTW (2.4) estimators involve an estimator g_n of g_0 obtained by fitting a correctly specified, main terms logistic regression PS model. The definitions of the C-TMLEs also involve estimators obtained by fitting main terms logistic regression PS model but with an additional layer of variable selection.

The simulation studies of Section 2.3 illustrate the relative performance of the estimators in scenarios with highly correlated covariates. These two scenarios are by far the most challenging settings for the greedy C-TMLE and scalable C-TMLE algorithms. The simulation studies of Section 2.3illustrate performance in situations where instrumental variables (covariates predictive of the treatment but not of the outcome) are included in the true PS model. In these two scenarios, greedy C-TMLE and our scalable C-TMLEs are expected to perform better, if not much better, than other widely used doubly-robust methods.

Table 2.1: Simulation study 1. Performance of the various estimators across 1000 simulated data sets of sample size 1000.

	well specified model for $\bar{Q}_0$			**mis-specified model for $\bar{Q}_0$**		
	bias (10^{-3})	se (10^{-2})	MSE (10^{-3})	bias (10^{-3})	se (10^{-2})	MSE (10^{-3})
unadj	2766.8	22.6	7706.3	2766.8	22.61	7706.3
A-IPTW	0.7	9.54	9.1	10.8	13.52	18.4
IPTW	75.9	34.91	127.5	75.9	34.91	127.5
MLE	1.0	8.20	6.7	699.4	13.96	508.6
TMLE	0.6	9.55	9.1	1.3	11.05	12.2
greedy C-TMLE	0.8	8.91	7.9	0.4	10.41	10.8
logRank C-TMLE	0.1	8.94	8.0	0.4	10.41	10.8
partRank C-TMLE	0.3	8.94	8.0	0.4	10.41	10.8
SL-C-TMLE	0.1	9.07	8.2	0.4	10.41	10.8

Simulation Study 1: Low-dimensional, highly correlated covariates

In the first simulation study, data were simulated based on a data generating distribution published by [12] and further analyzed by [47]. A pair of correlated, multivariate Gaussian baseline covariates (W_1, W_2) is generated as $(W_1, W_2) \sim N(\mu, \Sigma)$ where $\mu_1 = 0.5, \mu_2 = 1$ and $\Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$. The PS g_0 is given by

$$g_0(1|W) = \text{expit}(0.5 + 0.25 \times W_1 + 0.75 \times W_2)$$

(this is a slight modification of the mechanism in the original paper, which used a probit model to generate treatment). The outcome is continuous, $Y = \bar{Q}_0(A, W) + \epsilon$, with $\epsilon \sim N(0, 1)$ (independent of A, W) and $\bar{Q}_0(A, W) = 1 + A + W_1 + 2 \times W_2$. The true value of the target parameter is $\psi_0 = 1$.

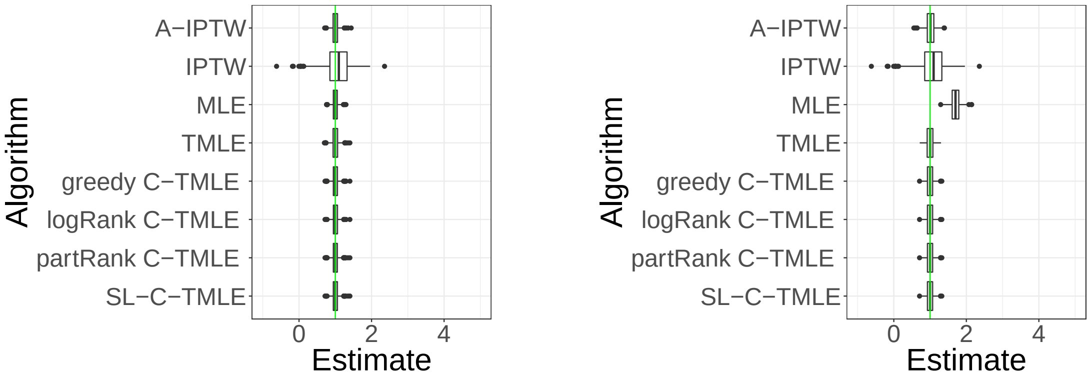
Note that *(i)* the two baseline covariates are highly correlated and *(ii)* the choice of g_0 yields practical (near) violation of the positivity assumption.

Each of the estimators involving the estimation of $\bar{Q}_0$ was implemented twice: by fitting a model correctly specified for $\bar{Q}_0$, and by regressing Y on A and W_1 only in a mis-specified linear model.

Bias, variance, and mean squared error (MSE) for all estimators across 1000 simulated data sets are shown in Table 2.1. Box plots of the estimated ATE are shown in Fig. 2.1.

When the model for $\bar{Q}_0$ was correctly specified, all estimators had very small bias. As Freedman and Berk discussed, even when the correct PS model was used, near positivity violations could lead to finite sample bias for IPTW estimators [see also 47]. Scalable C-TMLEs had smaller bias than the other DR estimators, but the distinctions were small.

When the model for $\bar{Q}_0$ was not correctly specified, the G-computation/MLE estimator was expected to be biased. Interestingly, A-IPTW was more biased than the other DR estimators. All C-TMLE estimators had identical performance, because each approach produced the same treatment model sequence.



(a) Well specified model for $\bar{Q}_0$.

(b) Mis-specified model for $\bar{Q}_0$.

Figure 2.1: Simulation 1: Box plot of the ATE estimates with well/mis- specified models for $\bar{Q}_0$. The green lines indicate the true parameter value.

Simulation Study 2: Highly correlated covariates

In the second simulation study, we tackle the case that multiple confounders are highly correlated with each other. Here, we use the notation $W_{1:k} = (W_1, \dots, W_k)$. The data-generating distribution is described as follows:

$$\begin{aligned}
W_1, W_2, W_3 \overset{iid}{\sim} & Bernoulli(0.5), \\
W_4 | W_{1:3} \sim & Bernoulli(0.2 + 0.5 \times W_1), \\
W_5 | W_{1:4} \sim & Bernoulli(0.05 + 0.3 \times W_1 + 0.1 \times W_2 \\
& + 0.05 \times W_3 + 0.4 \times W_4), \\
W_6 | W_{1:5} \sim & Bernoulli(0.2 + 0.6 \times W_5), \\
W_7 | W_{1:6} \sim & Bernoulli(0.5 + 0.2 \times W_3), \\
W_8 | W_{1:7} \sim & Bernoulli(0.1 + 0.2 \times W_2 + 0.3 \times W_6 \\
& + 0.1 \times W_7), \\
g_0(1|W) = & \operatorname{expit}(-0.05 + 0.1 \times W_1 + 0.2 \times W_2 \\
& + 0.2 \times W_3 - 0.02 \times W_4 \\
& - 0.6 \times W_5 - 0.2 \times W_6 - 0.1 \times W_7)
\end{aligned}$$

and, finally, for $\epsilon \sim N(0,1)$ (independent from A and W),

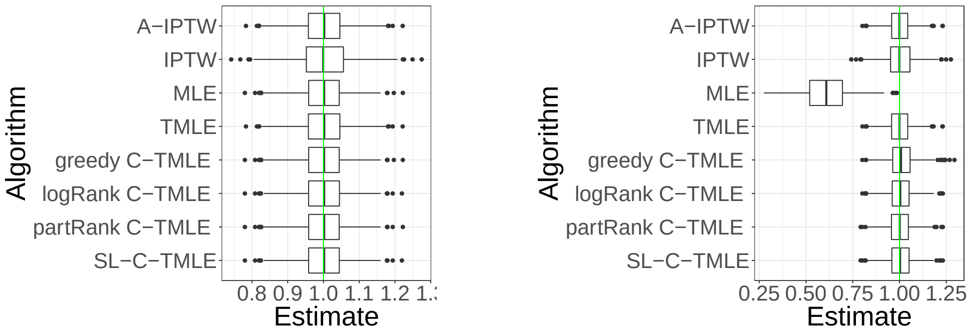
$$Y = 10 + A + W_1 + W_2 + W_4 + 2 \times W_6 + W_7 + \epsilon.$$

The true ATE for this simulation study is $\psi_0 = 1$.

In this case, the true confounders are W_1, W_2, W_4, W_6, W_7. Covariate W_5 is most closely related to W_6. Covariate W_3 is mainly associated with W_7. Neither W_3 nor W_5 is a confounder (both of them are predictive of treatment A, but do not influence directly outcome Y). Including either one of them in the PS model should inflate the variance [4].

Table 2.2: Simulation study 2. Performance of the various estimators across 1000 simulated data sets of sample size 1000.

	well specified model for $\bar{Q}_0$			**mis-specified model for $\bar{Q}_0$**		
	bias (10^{-3})	se (10^{-2})	MSE (10^{-3})	bias (10^{-3})	se (10^{-2})	MSE (10^{-3})
unadj	392.9	12.65	170.3	392.9	12.65	170.3
A-IPTW	2.4	6.54	4.3	2.0	6.53	4.3
IPTW	2.1	7.78	6.0	2.1	7.78	6.0
MLE	2.6	6.52	4.3	391.2	12.39	168.4
TMLE	2.4	6.54	4.3	2.0	6.53	4.3
greedy C-TMLE	2.6	6.52	4.3	11.4	7.01	5.0
logRank C-TMLE	2.5	6.52	4.3	6.3	6.72	4.6
partRank C-TMLE	2.6	6.52	4.3	2.5	6.67	4.4
SL-C-TMLE	2.5	6.52	4.3	5.2	6.79	4.6



(a) Well specified model for $\bar{Q}_0$.

(b) Mis-specified model for $\bar{Q}_0$.

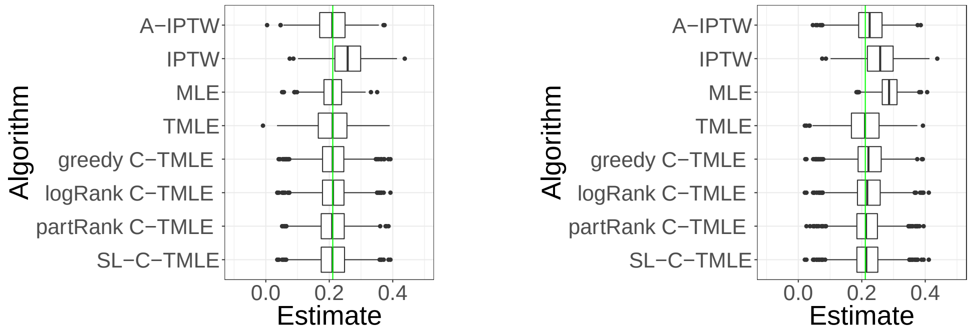
Figure 2.2: Simulation 2: Box plot of the ATE estimates with well/mis- specified models for $\bar{Q}_0$. The green line indicates the true parameter value.

As in Section 2.3, each of the estimators involving the estimation of $\bar{Q}_0$ was implemented twice: by fitting a model correctly specified for $\bar{Q}_0$, and by regressing Y on A only in a mis-specified linear model.

Table 2.2 demonstrates and compares performance across 1000 replications. Box plots of the estimated ATE are shown in Fig. 2.2. When $\bar{Q}_0$ was estimated by fitting a correctly specified model, all estimators except the unadjusted estimator had small bias. The DR estimators had lower MSE than the inefficient IPTW estimator. When $\bar{Q}_0$ was estimated by fitting a mis-specified model, the A-IPTW and IPTW estimators were less biased than the C-TMLE estimators. The bias of the greedy C-TMLE was five times larger. However, all DR estimators had lower MSE than the IPTW estimator, with the TMLE outperforming the others.

Table 2.3: Simulation study 3. Performance of the various estimators across 1000 simulated data sets of sample size 10000.

	well specified model for $\bar{Q}_0$			**mis-specified model for $\bar{Q}_0$**		
	bias (10^{-3})	se (10^{-2})	MSE (10^{-3})	bias (10^{-3})	se (10^{-2})	MSE (10^{-3})
unadj	78.1	3.72	7.5	78.1	3.72	7.5
A-IPTW	1.7	5.62	3.2	13.9	5.64	3.4
IPTW	45.9	6.05	5.8	45.9	6.05	5.8
MLE	0.7	4.20	1.8	76.4	3.61	7.1
TMLE	1.5	6.28	3.9	1.3	6.44	4.1
greedy C-TMLE	0.4	5.39	2.9	12.2	5.79	3.5
logRank C-TMLE	0.9	5.39	2.9	11.2	5.59	3.3
partRank C-TMLE	1.2	5.65	3.2	6.9	5.37	2.9
SL-C-TMLE	0.3	5.73	3.3	7.7	5.46	3.0



(a) Well specified model for $\bar{Q}_0$.

(b) Mis-specified model for $\bar{Q}_0$.

Figure 2.3: Simulation 3: Box plot of the ATE estimates with well/mis- specified models for $\bar{Q}_0$. The green line indicates the true parameter value.

Simulation Study 3: Binary outcome with instrumental variable

In the third simulation, we assess the performance of C-TMLE in a data set with positivity violations. We first generate W_1, W_2, W_3, W_4 independently from the uniform distribution on $[0, 1]$, then $A|W \sim \mathrm{B}ernoulli(g_0(1|W))$ with

$$g_0(1, W) = \mathrm{expit}(-2 + 5 \times W_1 + 2 \times W_2 + W_3),$$

and, finally, $Y|(A, W) \sim \mathrm{B}ernoulli(\bar{Q}_0(A, W))$ with

$$\bar{Q}_0(A, W) = \mathrm{expit}(-3 + 2 \times W_2 + 2 \times W_3 + W_4 + A).$$

As in Section 2.3 , each of the estimators involving the estimation of $\bar{Q}_0$ was implemented twice: by fitting a model correctly specified for $\bar{Q}_0$, and by regressing Y on A only in a mis-specified linear model.

Table 2.3 demonstrates the performance of the estimators across 1000 replications. Fig. 2.3 shows box plots of the estimates for the different methods across 1000 simulation, with a well specified or mis-specified model for $\bar{Q}_0$.

When the model for $\bar{Q}_0$ was correctly specified, the DR estimators had similar bias/variance trade-offs. Although IPTW is a consistent estimator when the model for the estimation of g_0 is correctly specified, truncation of the PS g_n may have introduced bias. However, without truncation it would have been extremely unstable due to violations of the positivity assumption when instrumental variables are included in the propensity score model.

When the model for $\bar{Q}_0$ was mis-specified, the MLE was equivalent to the unadjusted estimator. The DR methods performed well with an MSE close to the one observed when $\bar{Q}_0$ was estimated based on a correctly specified model. All C-TMLEs had similar performance. They out-performed the other DR methods (namely, A-IPTW and TMLE) and the pre-ordering strategies improved the computational time without loss of precision or accuracy compared to the greedy C-TMLE algorithm.

Side note.

Because W_1 is an instrumental variable that is highly predictive of the PS, but not helpful for confounding control, we expect that including it in the PS model would increase the variance of the estimator. One possible way to improve the performance of the IPTW estimator would be to apply a C-TMLE algorithm to select covariates for fitting the PS model. In the mis-specified model for $\bar{Q}_0$ scenario, we also simulated the following procedure:

1. Use a greedy C-TMLE algorithm to select the covariates.

2. Use main terms logistic regression with selected covariates for the PS model.

3. Compute IPTW using the estimated PS.

The simulated bias for this estimator was 0.0340, the SE was 0.0568, and the MSE was 0.0043. Excluding the instrumental variable from the PS model thus reduced bias, variance, and MSE of the IPTW estimator.

Simulation Study 4: Continuous outcome

In the fourth simulation, we assess the performance of C-TMLEs in a simulation scheme with a continuous outcome inspired by that of [17] (we merely increased the coefficient in front of W_1 to introduce a stronger positivity violation). We first independently draw $W_1, W_2, W_3, W_4, W_5, W_6$ from the standard normal law, then A given W with

$$g_0(1, W) = \text{expit}(2 \times W_1 + 0.2 \times W_2 - 3 \times W_3)$$

and, finally Y given (A, W) from a Gaussian law with variance 1 and mean $\bar{Q}_0(A, W) =$

$$0.5 \times W_1 - 8 \times W_2 + 9 \times W_3 - 2 \times W_5 + A.$$

The initial estimator $\bar{Q}_n^0$ was built based on a linear regression model of Y on A, W_1, and W_2, thus partially adjusting for confounding. There was residual confounding due to

W_3. There was also residual confounding due to W_1 and W_2 within at least one stratum of A, despite their inclusion in the initial outcome regression model.

Table 2.4: Simulation study 4. Performance of the various estimators across 1000 simulated data sets of sample size 1000. Omitted in the table, the performance of the unadjusted estimator was an order of magnitude worse than the performance of the other estimators.

	Mis-specified model for $\bar{Q}_0$		
	bias	se	MSE
A-IPTW	4.49	0.84	20.88
IPTW	2.97	0.87	9.60
MLE	12.68	0.47	161.20
TMLE	1.31	1.21	3.17
greedy C-TMLE	0.25	1.01	1.27
logRank C-TMLE	0.36	0.88	0.90
partRank C-TMLE	0.32	0.92	0.95
SL-C-TMLE	0.37	0.88	0.90

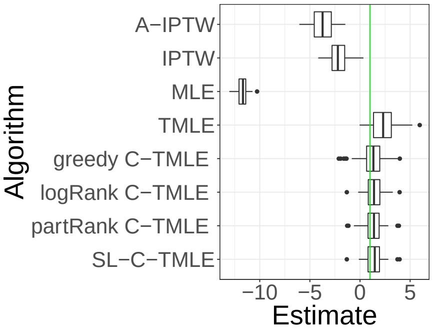


Figure 2.4: Simulation 4: Box plot of the ATE estimates with mis-specified model for $\bar{Q}_0$.

Fig. 2.4 reveals that the C-TMLEs performed much better than TMLE and A-IPTW estimators in terms of bias and standard error. This illustrates that choosing to adjust for less than the full set of covariates can improve finite sample performance when there are near positivity violations. In addition, Table 2.4 shows that the pre-ordered C-TMLEs outperformed the greedy C-TMLE. Although the greedy C-TMLE estimator had smaller bias, it had higher variance, perhaps due to its more data adaptive ordering procedure.

Simulation Study on Partially Synthetic Data

The aim of this section is to compare TMLE and all C-TMLEs using a large simulated data set that mimics a real-world data set. Section 2.3 starts the description of the data-generating scheme and resulting large data set. Section 2.3 presents the High-Dimensional Propensity Score (hdPS) method used to reduce the dimension of the data set. Section 2.3 completes the description of the data-generating scheme and specifies how $\bar{Q}_0$ and g_0 are estimated. Section 2.3 summarizes the results of the simulation study.

Data-generating scheme

The simulation scheme relies on the Nonsteroidal anti-inflammatory drugs (NSAID) data set presented and studied in [66, 52]. Its $n = 49,653$ observations were sampled from a population of patients aged 65 years and older, and enrolled in both Medicare and the Pennsylvania Pharmaceutical Assistance Contract for the Elderly (PACE) programs between 1995 and 2002. Each observed data structure consists of a triplet (W, A, Y) where W is decomposed in two parts: a vector of 22 baseline covariates and a highly sparse vector of $C = 9,470$ unique claims codes. In the latter, each entry is a nonnegative integer indicating how many times (mostly zero) a certain procedure (uniquely identified among $C = 9,470$ by its claims code) has been undergone by the corresponding patient. The claims codes were manually grouped into eight categories: ambulatory diagnoses, ambulatory procedures, hospital diagnoses, hospital procedures, nursing home diagnoses, physician diagnoses, physician procedures and prescription drugs. The binary indicator A stands for exposure to a selective COX-2 inhibitor or a comparison drug (a non-selective NSAID). Finally, the binary outcome Y indicates whether or not either a hospitalization for severe gastrointestinal hemorrhage or peptic ulcer disease complications including perforation in GI patients occurred.

The simulated data set was generated as in [14, 10]. It took the form of $n = 49,653$ data structures (W_i, A_i, Y_i) where $\{(W_i, A_i) : 1 \leq i \leq n\}$ was extracted from the above real data set and where $\{Y_i : 1 \leq i \leq n\}$ was simulated by us in such a way that, for each $1 \leq i \leq n$, the random sampling of Y_i depended only on the corresponding (W_i, A_i). As argued in the aforementioned articles, this approach preserves the covariance structure of the covariates and complexity of the true treatment assignment mechanism, while allowing the true value of the ATE parameter to be known. In addition, we can control the bias in the unadjusted estimator by tuning the coefficients of the parametric data generating conditional distribution of Y given (A, W), if there exist covariates associated with the treatment mechanism.

High-Dimensional Propensity Score Method For Dimension Reduction

The simulated data set was large, both in number of observations and number of covariates. In this framework, directly applying any version of C-TMLE algorithms would not be the best course of action. First, the computational time would be unreasonably long due to the large number of covariates. Second, the resulting estimators would be plagued by high variance due to the low signal-to-noise ratio in the claims data. This motivated us to apply the hdPS method for dimension reduction prior to applying the TMLE and C-TMLE algorithms.

Introduced in [66], the hdPS method was proposed to reduce the dimension in large electronic healthcare databases. It is increasingly used in studies involving such databases [52, 46, 11, 71, 33, 30].

The hdPS method essentially consists of two main steps: *(i)* generating so called hdPS covariates from the claims data (which can increase the dimension) then *(ii)* screening the enlarged collection of covariates to select a small proportion of them (which dramatically reduces the dimension). Specifically, the method unfolds as follows [66]:

A **G**roup by resource. Group the data by resource in $\mathcal{C}$ groups

B **I**dentify candidate claims codes. For each group separately, for each claims code c within the group, compute the empirical proportion $Pr(c)$ of positive entries, then sort the claims codes by decreasing values of $\min(Pr(c), 1 - Pr(c))$. Finally, select only the top J claims codes. We thus go from C claims codes to $J \times \mathcal{C}$ claims codes.

C **A**ssess recurrence of claims codes. For each selected claims code c and each patient $1 \leq i \leq n$, *replace* the corresponding c_i with three binary covariates called "hdPS covariates": $c_i^{(1)}$ equal to one if and only if (iff) c_i is positive; $c_i^{(2)}$ equal to one iff c_i is larger than the median of $\{c_i : 1 \leq i \leq n\}$; $c_i^{(3)}$ equal to one iff c_i is larger than the 75%-quantile of $\{c_i : 1 \leq i \leq n\}$. This inflates the number of claims codes related covariates by a factor 3.

D **S**elect among the hdPS covariates. For each hdPS covariate, estimate a measure of its "potential confounding impact" (a heuristic), then sort them by decreasing values of the estimates of the measure. Finally, select only the top K hdPS covariates.

In the current example, we derived $\mathcal{C} = 8$ groups in step A. The groups correspond to the following categories: ambulatory diagnoses, ambulatory procedures, hospital diagnoses, hospital procedures, nursing home diagnoses, physician diagnoses, physician procedures and prescription drugs. See [66, 46] for other examples.

In step B, we chose $J = 50$. The dimension of the claims data thus went from $9,470$ to 400.

In step C, we relied on the following estimate of the measure of the potential confounding impact introduced in [**bross54**]: for hdPS covariate c^ℓ

$$\frac{\pi_n^\ell(1)(r_n^\ell - 1) + 1}{\pi_n^\ell(0)(r_n^\ell - 1) + 1} \tag{2.5}$$

where

$$\begin{aligned} \pi_n^\ell(a) &= \frac{\sum_{i=1}^n \mathbf{1}\{c_i^\ell = 1, a_i = a\}}{\sum_{i=1}^n \mathbf{1}\{a_i = a\}} \quad (a = 0, 1) \quad \text{and} \\ r_n^\ell &= \frac{p_n(1)}{p_n(0)} \quad \text{with} \\ p_n(c) &= \frac{\sum_{i=1}^n \mathbf{1}\{y_i = 1, c_i^\ell = c\}}{\sum_{i=1}^n \mathbf{1}\{c_i^\ell = c\}} \quad (c = 0, 1). \end{aligned}$$

A rationale for this choice can be found in [66], where r_n^ℓ in (2.5) is replaced by $\max(r_n^\ell, 1/r_n^\ell)$. As explained below we chose $K = 100$. As a result, the dimension of the claims data was thus reduced to 100 from $9,470$.

Data-generating Scheme (cont.) and Estimating Procedures

Let us resume here the presentation of the simulation scheme initiated in Section 2.3. Recall that the simulated data set writes as $\{(W_i, A_i, Y_i) : 1 \leq i \leq n\}$ where $\{W_i : 1 \leq i \leq n\}$ is the by-product of the hdPS method of Section 2.3 with $J = 50$ and $K = 100$ and $\{A_i : 1 \leq i \leq n\}$ is the original vector of exposures. It only remains to present how $\{Y_i : 1 \leq i \leq n\}$ was generated.

First, we arbitrarily chose a subset W' of W, that consists of 10 baseline covariates (*congestive heart failure*, *previous use of warfarin*, *number of generic drugs in last year*, *previous use of oral steroids*, *rheumatoid arthritis*, *age in years*, *osteoarthritis*, *number of doctor visits in last year*, *calendar year*) and 5 hdPS covariates. Second, we arbitrarily defined a parameter

$$\beta = (1.280, -1.727, 1.690, 0.503, 2.528, 0.549, 0.238, -1.048, 1.294, 0.825, \\ -0.055, -0.784, -0.733, -0.215, -0.334)^\top$$

(the entries of β were drawn independently from standard normal random variables). Finally, $Y_1, \ldots, Y_n$ were independently sampled given $\{(W_i, A_i) : 1 \leq i \leq n\}$ from Bernoulli distributions with parameters $q_1, \ldots, q_n$ where, for each $1 \leq i \leq n$,

$$q_i = \operatorname{expit}\left(\beta^\top W_i' + A_i\right).$$

The resulting true value of the ATE is $\psi_0 = 0.21156$.

The estimation of the conditional expectation $\bar{Q}_0$ was carried out based on two logistic regression models. The first one was well specified whereas the second one was mis-specified, due to the omission of the five hdPS covariates.

For the TMLE algorithm, the estimation of the PS g_0 was carried out based on a single, main terms logistic regression model including all of the 122 covariates. For the C-TMLE algorithms, main terms logistic regression model were also fitted at each step. An early stopping rule was implemented to save computational time. Specifically, if the cross-validated loss of $\bar{Q}^*_{n,k}$ is smaller than the cross-validated losses of $\bar{Q}^*_{n,k+1}, \ldots, \bar{Q}^*_{n,k+10}$, then the procedure is stopped and outputs the TMLE estimator corresponding to $\bar{Q}^*_{n,k}$.

The scalable SL-C-TMLE library included the two scalable pre-ordered C-TMLE algorithms and excluded the greedy C-TMLE algorithm.

Results

Table 2.5 reports the point estimates for ψ_0 as derived by all the considered methods. It also reports the 95% CIs of the form $[\psi_n \pm 1.96\sigma_n/\sqrt{n}]$, where $\sigma_n^2 = n^{-1}\sum_{i=1}^n D^*(\bar{Q}_n, g_n)(O_i)^2$ estimates the variance of the efficient influence curve at the couple $(\bar{Q}_n, g_n)$ yielding ψ_n. We refer the interested reader to [38, Appendix A] for details on influence curve based inference.

Table 2.5: Point estimates and 95% CIs of TMLE and C-TMLE estimators for the partially synthetic data simulation study.

	model for $\bar{Q}_0$	estimate	CI	processing time
TMLE	well specified	0.202	(0.193, 0.212)	0.6s
	mis-specified	0.203	(0.193, 0.213)	0.6s
C-TMLE, greedy	well specified	0.205	(0.196, 0.213)	618.7s
	mis-specified	0.214	(0.205, 0.223)	1101.2s
C-TMLE, logistic ordering	well specified	0.205	(0.196, 0.213)	57.4s
	mis-specified	0.211	(0.202, 0.219)	125.6s
C-TMLE, partial correlation ordering	well specified	0.205	(0.197, 0.213)	22.5s
	mis-specified	0.211	(0.202, 0.219)	149.0s
SL-C-TMLE	well specified	0.205	(0.197, 0.213)	69.8s
	mis-specified	0.211	(0.202, 0.219)	264.3s

All the CIs contained the true value of ψ_0. Table 2.5 also reports processing times (in seconds).

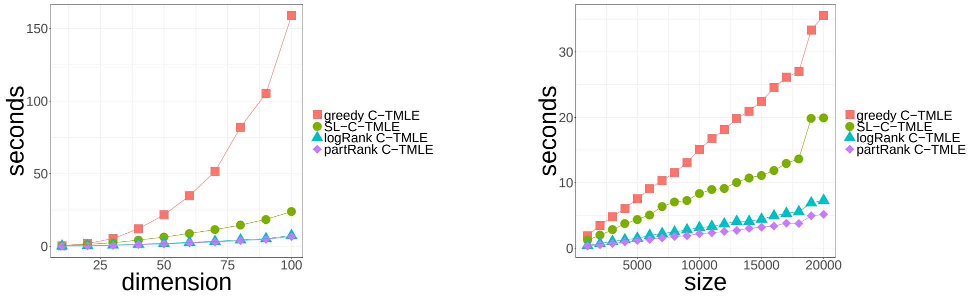
The point estimates and CIs were similar across all C-TMLEs. When the model for $\bar{Q}_0$ was correctly specified, the SL-C-TMLE selected the partial correlation ordering. When the model for $\bar{Q}_0$ was mis-specified, it selected the logistic ordering. In both cases, the estimator with smaller bias was data adaptively selected. In addition, as all the candidates in its library were scalable, the SL-C-TMLE algorithm was also scalable, and ran much faster than the greedy C-TMLE algorithm. Computational time for the scalable C-TMLE algorithms was approximately 1/10th of the computational time of the greedy C-TMLE algorithm.

2.4 Time Complexity

We study here the computational time of the pre-ordered C-TMLE algorithms. The computational time of each algorithm depends on the sample size n and number of covariates p. First, we set $n = 1,000$ and varied p between 10 and 100 by steps of 10. Second, we varied n from $1,000$ to $20,000$ by steps of $1,000$ and set $p = 20$. For each (n, p) pair, the analysis was replicated ten times independently, and the median computational time was reported. In every data set, all the random variables are mutually independent. The results are shown in Figures 2.5a and 2.5b.

Figure 2.5a is in line with the theory: the computational time of the forward stepwise C-TMLE is $\mathcal{O}(p^2)$ whereas the computational times of the pre-ordered C-TMLE algorithms are $\mathcal{O}(p)$. Note that the pre-ordered C-TMLEs are indeed scalable. When $n = 1,000$ and $p = 100$, all the scalable C-TMLE algorithms ran in less than 30 seconds.

Figure 2.5b reveals that the pre-ordered C-TMLE algorithms are much faster in practice than the greedy C-TMLE algorithm, even if all computational times are $\mathcal{O}(n)$ in that framework with fixed p.



(a) Median computational time (across 10 replications for each point), with $n = 1,000$ fixed and p varying.

(b) Median computational time (across 10 replications for each point), with varying n and fixed $p = 20$.

Figure 2.5: Computational times of the C-TMLE algorithms with greedy search and preordering.

2.5 Applications in Electronic Healthcare Database

This section presents the application of variants of the TMLE and C-TMLE algorithms for the analysis of three real data sets. Our objectives are to showcase their use and to illustrate the consistency of the results provided by the scalable and greedy C-TMLE estimators. We thus do not implement the competing unadjusted, G-computation/MLE, IPTW and A-IPTW estimators (see the beginning of Section 2.3).

In Sections 2.3 and 2.3, we knew the true value of the ATE. This is not the case here.

Real Data Sets and Estimating Procedures

We compared the performance of variants of TMLE and C-TMLE algorithms across three observational data sets. Here are brief descriptions, borrowed from [66, 30].

NSAID Data Set. Refer to Section 2.3 for its description.

Novel Oral Anticoagulant (NOAC) Data Set. The NOAC data were collected between October, 2009 and December, 2012 by United Healthcare. The data set tracked a cohort of new users of oral anticoagulants for use in a study of the comparative safety and effectiveness of these agents. The exposure is either "warfarin" or "dabigatran". The binary outcome indicates whether or not a patient had a stroke during the 180 days after initiation of an anticoagulant.

The data set includes $n = 18,447$ observations, $p = 60$ baseline covariates and $C = 23,531$ unique claims codes. The claims codes are manually grouped in four categories: inpatient diagnoses, outpatient diagnoses, inpatient procedures and outpatient procedures.

Vytorin Data Set. The Vytorin data included all United Healthcare patients who initiated either treatment between January 1, 2003 and December 31, 2012, with age over 65 on day of entry into cohort. The data set tracked a cohort of new users of Vytorin and high-intensity statin therapies. The exposure is either "Vytorin" or "high-intensity statin". The outcomes indicates whether or not any of the events "myocardial infarction", "stroke" and "death" occurred.

The data set includes $n = 148,327$ observations, $p = 67$ baseline covariates and $C = 15,010$ unique claims codes. The claims codes are manually grouped in five categories: ambulatory diagnoses, ambulatory procedures, hospital diagnoses, hospital procedures, and prescription drugs.

Each data set is given by $\{(W_i, A_i, Y_i) : 1 \le i \le n\}$ where $\{W_i : 1 \le i \le n\}$ is the by-product of the hdPS method of Section 2.3 with $J = 100$ and $K = 200$ and $\{(A_i, Y_i) : 1 \le i \le n\}$ is the original collection of paired exposures and outcomes.

The estimations of the conditional expectation $\bar{Q}_0$ and of the PS g_0 were carried out based on logistic regression models. Both models used either the baseline covariates only or the baseline covariates *and* the additional hdPS covariates.

To save computational time, the C-TMLE algorithms relied on the same early stopping rule described in Section 2.3. The scalable SL-C-TMLE library included the two scalable pre-ordered C-TMLE algorithms and excluded the greedy C-TMLE algorithm.

Results on the NSAID Data Set

Figure 2.6 shows the point estimates and 95% CIs yielded by the different TMLE and C-TMLE estimators built from the NSAID data set.

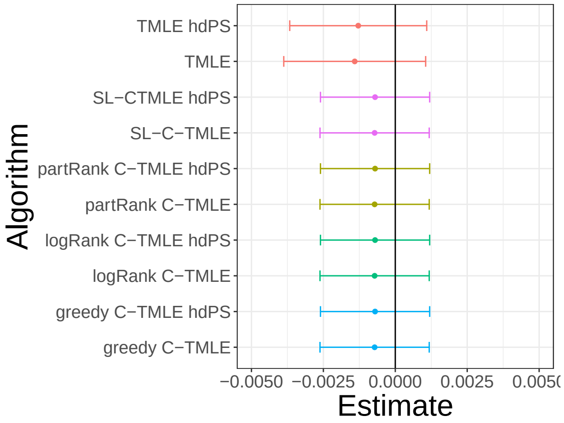


Figure 2.6: Point estimates and 95% CIs yielded by the different TMLE and C-TMLE estimators built on the NSAID data set.

The various C-TMLE estimators exhibit similar results, with slightly larger point estimates and narrower CIs compared to the TMLE estimators. All the CIs contain zero.

Results on the NOAC Data Set

Figure 2.7 shows the point estimates and 95% CIs yielded by the different TMLE and C-TMLE estimators built on the NOAC data set.

We observe more variability in the results than in those presented in section 2.5.

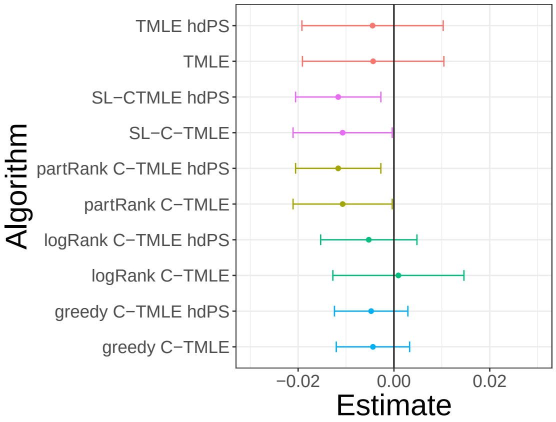


Figure 2.7: Point estimates and 95% CIs yielded by the different TMLE and C-TMLEs built on the NOAC data set.

The various TMLE and C-TMLEs exhibit similar results, with a non-significant shift to the right for the latter. All the CIs contain zero.

Results on the Vytorin Data Set

Figure 2.8 shows the point estimates and 95% CIs yielded by the different TMLE and C-TMLEs built on the Vytorin data set.

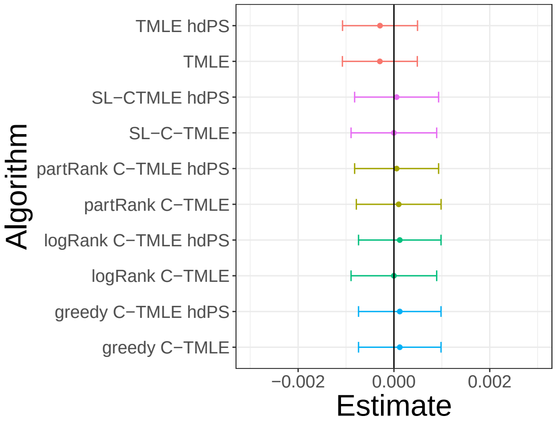


Figure 2.8: Point estimates and 95% CIs yielded by the different TMLE and C-TMLEs built on the Vytorin data set.

The various TMLE and C-TMLEs exhibit similar results, with a non-significant shift to the right for the latter. All the CIs contain zero.

2.6 Discussion

Robust inference of a low-dimensional parameter in a large semi-parametric model traditionally relies on external estimators of infinite-dimensional features of the distribution of the data. Typically, only one of the latter is optimized for the sake of constructing a well behaved estimator of the low-dimensional parameter of interest. For instance, the targeted minimum loss (TMLE) estimator of the average treatment effect (ATE) (1.3) relies on an external estimator $\bar{Q}_n^0$ of the conditional mean $\bar{Q}_0$ of the outcome given binary treatment and baseline covariates, and on an external estimator g_n of the PS g_0. Only $\bar{Q}_n^0$ is optimized/updated into $\bar{Q}_n^*$ based on g_n in such a way that the resulting substitution estimator of the ATE can be used, under mild assumptions, to derive a narrow confidence interval with a given asymptotic level.

There is room for optimization in the estimation of g_0 for the sake of achieving a better bias-variance trade-off in the estimation of the ATE. This is the core idea driving the general C-TMLE template. It uses a targeted penalized loss function to make smart choices in determining which variables to adjust for in the estimation of g_0, only adjusting for variables that have not been fully exploited in the construction of $\bar{Q}_n^0$, as revealed in the course of a data-driven sequential procedure.

The original instantiation of the general C-TMLE template was presented as a greedy forward stepwise algorithm. It does not scale well when the number p of covariates increases drastically. This motivated the introduction of novel instantiations of the C-TMLE general template where the covariates are pre-ordered. Their time complexity is $\mathcal{O}(p)$ as opposed to the original $\mathcal{O}(p^2)$, a remarkable gain. We proposed two pre-ordering strategies and suggested a rule of thumb to develop other meaningful strategies. Because it is usually unclear a priori which pre-ordering strategy to choose, we also introduced a SL-C-TMLE algorithm that enables the data-driven choice of the better pre-ordering given the problem at hand. Its time complexity is $\mathcal{O}(p)$ as well.

The C-TMLE algorithms used in our data analyses have been implemented in Julia and are publicly available at `https://lendle.github.io/TargetedLearning.jl/`. We have also published the R version of the scalable C-TMLE in the official *ctmle* package [27] in The Comprehensive R Archive Network. We undertook five simulation studies. Four of them involved fully synthetic data. The last one involves partially synthetic data based on a real electronic health database and the implementation of a hdPS method for dimension reduction widely used for the statistical analysis of claims codes data. In Section 2.4, we compare the computational times of variants of C-TMLE algorithms. We also showcase the use of C-TMLE algorithms on three real electronic health database. In all analyses involving electronic health databases, the greedy C-TMLE algorithm was unacceptably slow. Judging from the simulation studies, our scalable C-TMLE algorithms work well, and so does the SL-C-TMLE algorithm.

Chapter 3

Model Selection among Continuously-indexed Nuisance Parameter Estimators with Collaborative TMLE

3.1 Introduction

Introduced in 1.1, the propensity score (PS) is defined as the conditional probability of treatment assignment, given a set of observed pre-treatment covariates [64, 24]. The PS, which we will denote as g_0, is widely used to control for confounding bias in observational studies. In practice, the PS is usually unknown and PS based estimators must rely on an estimate of the PS, which we will denote as g_n.

Accurately modeling and assessing the validity of fitted PS models is crucial for all PS-based methods. It is generally recommended that PS models be validated through measures of covariate balance across treatment groups after PS adjustment. In high-dimensional covariate settings, however, evaluating covariate balance on very large numbers of variables can be difficult. Using covariate balance to validate PS models in high-dimensional covariate settings is further complicated when applying machine learning (ML) algorithms and penalized regression methods to reduce the dimension of the covariate set, as it is not always clear on what variables balance should be evaluated. Cross-validated prediction diagnostics can greatly simplify validation of the PS model when applying ML algorithms for PS estimation in high-dimensional covariate settings.

[76] suggested that ML methods (e.g. support vector machines) could enhance the validity of propensity score estimation, and that "external" cross-validation (CV) can be used for model selection. [42] further investigated PS weighted estimators when the PS was estimated by multiple ML algorithms, where the hyper-parameters of the ML algorithms were selected by minimizing the CV loss for treatment prediction. Estimation procedures that are based

on external CV will result in estimated models that optimize the bias-variance tradeoff for treatment prediction (i.e., the true PS function), but they do not consider the ultimate goal of optimizing the bias-variance tradeoff for the treatment effect estimate. We conjecture that PS estimators that are selected by CV will tend to be under-fitted in order to reduce variability in the prediction of treatment assignment, and that the optimal estimator in reducing bias in the estimated treatment effect should be less smooth (or more flexible) compared to the estimator selected by external CV.

To address this limitation of external CV, we studied two recently proposed variations of the C-TMLE algorithm [26, 34, 28], and compared them to other widely used estimators using multiple simulation studies. We focused on strategies that combined the C-TMLE algorithms with LASSO regression, an l-1 regularized logistic regression [70], for PS estimation. Previous studies have shown that LASSO regression can perform well for variable selection when estimating high-dimensional PSs [11]. However, selecting the optimal tuning parameters to optimize confounding control remains challenging. Combining variations of the C-TMLE algorithm with LASSO regression provides a robust data adaptive approach to PS model selection in high-dimensional covariate datasets, but remains untested. We used quasi-experiments based on a real empirical dataset to evaluate the performance of combining variations of the C-TMLE algorithm with LASSO regression and demonstrate that external CV for model selection is insufficient.

This chapter is organized as follows. In section 3.2, we introduce how to use C-TMLE to tune the PS estimator with one-dimensional hyper-parameter by taking LASSO as an example. In section 3.3 we describe how the simulated data are generated from the empirical dataset introduced in 2.5, and how results were analyzed from the simulation, including point estimation, confidence interval, and pair-wise comparisons of estimators. In section 3.4 we apply the vanilla TMLE and novel C-TMLE algorithms to analyze the empirical dataset. Finally, in section 3.5, we discuss the results from the simulations and the scientific findings from the empirical data analysis.

3.2 Shrinkage Parameter Selection for LASSO with C-TMLE

C-TMLE was primarily proposed for variable selection [16]. However, it can easily be adapted to more general model selection problems. In our recent work [26, 34], two instantiations of the C-TMLE algorithm were proposed for a general model selection problem with a one-dimensional hyper-parameter. In this study, we consider an example where the PS model is estimated by LASSO:

$$\begin{aligned}\beta_{n,\lambda} &= \min_{\beta \in \mathbb{R}^p} \left(\frac{1}{n} \sum_{i=1}^{n} L(A_i, \text{logit}(\beta W_i)) + \lambda \|\beta\|_1 \right) \\ g_{n,\lambda}(W_i) &= \text{logit}(\beta_{n,\lambda} W_i)\end{aligned}$$

where L is the negative log-likelihood for the Bernoulli distribution, as A is binary. We used C-TMLE to select the PS estimator, $g_{n,\lambda}$, with the best regularization parameter λ. We applied two C-TMLE algorithms for model selection of LASSO. Here, we provide a brief outline for each of the algorithms. Details are provided in the supplemental appendices.

- LASSO-C-TMLE: First, we briefly introduce the LASSO-C-TMLE (C-TMLE1) algorithm. According to the C-TMLE template outlined above, C-TMLE1 first builds an initial estimate for $\bar{Q}_n$ and a sequence of propensity score estimators, g_{n,λ_k}, for $k \in 0, \dots, K$, each with a penalty λ_k, where λ_k is monotonically decreasing. We recommend to set $\lambda_1 = \lambda_{CV}$ because the cross-validation usually selects an "under-fitted" (for example, a LASSO estimator with a regularization parameter, λ, that is too large) PS estimator; thus, it is unnecessary to consider $\lambda_1 > \lambda_{CV}$. Then, we just follow step 3 in the template described previously, and build a sequence of estimators, $\bar{Q}^*_{n,\lambda}$, each corresponding to $g_{n,\lambda}$. We then select the best $\bar{Q}^*_{n,\lambda_{ctmle}}$ by using cross-validation, with its corresponding initial estimate $\bar{Q}_{n,\lambda_{ctmle}}$. Finally we fluctuate the selected initial estimate $\bar{Q}_{n,\lambda_{ctmle}}$ with each $g_{n,\lambda}$ for $\lambda_K < \lambda < \lambda_{ctmle}$, yielding a new sequence $\bar{Q}^*_{n,\lambda}$. We choose $\bar{Q}^*_n = \bar{Q}^*_{n,\lambda}$, which minimizes the empirical loss, as our final estimate. The final step guarantees that a critical equation:

$$P_n D^+(\bar{Q}^*_{n,\lambda}, g_{n,\lambda}) = \frac{\partial}{\partial \lambda} \sum_{i=1}^{n} H_{g_{n,\lambda}}(A_i, W_i)(Y_i - \bar{Q}^*_{n,\lambda}(A_i, Y_i)) = 0 \quad (3.1)$$

is solved [26, 34]. This guarantees that the resulting C-TMLE estimator is asymptotically linear under regularity conditions even when $\bar{Q}_n$ is not consistent.

- LASSO-PSEUDO-C-TMLE: the LASSO-PSEUDO-C-TMLE (C-TMLE0) algorithm does not select the PS estimator collaboratively. Instead, it is exactly the same as the TMLE algorithm, except it updates the estimate by equation 3.2:

$$\text{logit}(\bar{Q}^*_n(A, W)) = \text{logit}(\bar{Q}_n(A, W)) + \epsilon_1 H_{g_{n,\lambda_k}}(A, W) + \epsilon_2 \tilde{H}_{g_{n,\lambda_k}}(A, W) \quad (3.2)$$

where

$$\begin{aligned}\tilde{H}_{g_{n,\lambda_k}}(A, W) &= \frac{\partial H_{g_{n,\lambda}}(A, W)}{\partial \lambda}|_{\lambda=\lambda_k} \\ &= \frac{1 - A}{(1 - g_{n,\lambda_k}(W))^2} \frac{\partial(1 - g_{n,\lambda})}{\partial \lambda}|_{\lambda=\lambda_k} \\ &+ \frac{A}{g_{n,\lambda}(W)^2} \frac{\partial g_{n,\lambda_k}}{\partial \lambda}|_{\lambda=\lambda_k}.\end{aligned}$$

Note we still call it C-TMLE as it solves the critical equation 3.2. Solving the additional clever covariate $\tilde{H}_{g_{n,\lambda_k}}(A, W)$ could be considered as an approximation of the collaborative selection in C-TMLE1 [26, 34].

Same as the discrete C-TMLE estimator in [16], standard errors for both of the new C-TMLEs are computed based on the variance of the influence curve (IC). With the point estimate, $\hat{\psi}$, and its estimated standard error, $\hat{\text{se}}$, we construct the Wald-type α-level confidence interval: $[\hat{\psi} - z_{1-\alpha/2}\hat{\text{se}}, \hat{\psi} + z_{1-\alpha/2}\hat{\text{se}}]$, where z_α is the α-percentile of the standard normal distribution. More details of IC and the IC based variance estimator can be found in the literature [39, 16].

For simplicity, we denote LASSO-CTMLE as C-TMLE1, and LASSO-Pseudo-C-TMLE as C-TMLE0.

3.3 Quasi-Experiment

Simulation Setting

In this simulation, we generated partially synthetic data based on the NSAID data set introduced in Section 2.3. We designed our own conditional distribution of the outcome, Y, given treatment, A, and baseline covariates, W, while keeping the structure of the treatment mechanism $g_0(A|W)$ so that the relationships between covariates with treatment assignment were preserved [10]. In our study, the conditional distribution of the outcome was defined as:

$$Y_i = 2 + \beta W_i + A_i + \epsilon_i \tag{3.3}$$

where ϵ_i is drawn independently from the standard normal distribution. We then selected 40 covariates that had the highest Pearson correlation with treatment A. The coefficient of β in equation 3.3 was set to zero for all the non-selected covariates. The coefficient for the selected covariates was sampled from separate and independent standard normal distributions, and were fixed across all simulations. We define the marginal distribution of W as the empirical distribution of W_i for $i \in 1 \dots n$. The parameter of interest is the ATE, thus it is identifiable if we know the distribution of the conditional response $Y|A, W$ and marginal distribution of W.

In our simulation, we considered two settings. In the first setting, only the first 10 out of 40 confounders were used to estimate $\bar{Q}_0$. In the second setting, $\bar{Q}_0$ was estimated using the first 20 out of 40 confounders.

By the description above, we have the following:

- There are only 40 confounders in total.
- The true value of the parameter of interest (ATE) is 1.
- The treatment mechanism $g_0(A|W)$ comes from a real world data generating distribution, which is usually non-linear. [30] showed that the PS in this example can be estimated well by linear models. Therefore, in this example the PS model is only mildly misspecified.

- Both $\bar{Q}_0$ and g_0 are estimated with a misspecified model: $\bar{Q}_0$ is estimated with an incomplete predictor set; g_0 is estimated with linear model, while there is no reason to believe it is truly linear.

The results are computed across 500 independent replications, each with sample sizes of 1000.

Competing Estimators

In this study, we focused on PS based estimators, including inverse probability of treatment weight (IPW) estimator, Hajek type IPW estimator, double robust (augmented) inverse probability of treatment weight (DR-IPW, or A-IPW) estimator, Hajek type Bias-correction (HBC) Estimator, weighted regression (WR) estimator, targeted maximum likelihood estimator (TMLE), and the proposed two collaborative-TMLE estimators.

For all PS based estimators, we consider two variations. For the first variation, we first used the cross-validated LASSO (CV-LASSO) algorithm to find the regularization parameter λ_{CV} of LASSO for PS estimation, and then plugged it into the final estimators. In the second variation, we first applied C-TMLE1, and used LASSO with the regularization parameter λ_{C-TMLE} selected by C-TMLE1 to estimate the PS, and then plugged it into the estimator. Taking IPW as example, we used "IPW" to denote the first variation, and "IPW*" for the second variation.

It is important to note that in this case, "TMLE*" is actually a variation of collaborative TMLE, as the PS model is selected collaboratively [16, 40]. However, it is different from the proposed C-TMLE algorithms, as it does not solve the critical equation 3.1.

It is also important to note that both C-TMLE and CV-LASSO use cross-validation. For simplicity, and to avoid ambiguity, we use term "CV" to denote the non-collaborative model selection procedure which relies on the cross-validation w.r.t. the prediction performance for the treatment mechanism itself (e.g. the model selection step in CV-LASSO).

In addition, we also compute an "oracle estimator" for comparison, which is given by a TMLE estimator with the PS estimated by a logistic regression with only confounders.

Point Estimation

We first compared the variance, bias, and mean square error (MSE) for the point estimation from all the competing estimators in two settings.

Table 3.1 and figure 3.1 show the point estimation performance of all the competing estimators. It is not surprising that the oracle TMLE estimator has the best performance for both bias and variance. However, it is not achievable in practice as it is usually unknown which covariates are confounders. IPW has very large variance and bias, which might due to the practical violations of the positivity assumption. We can see that TMLE*, C-TMLE1, CTMLE0, and CTMLE0* outperformed other estimators, with each having similar perfor-

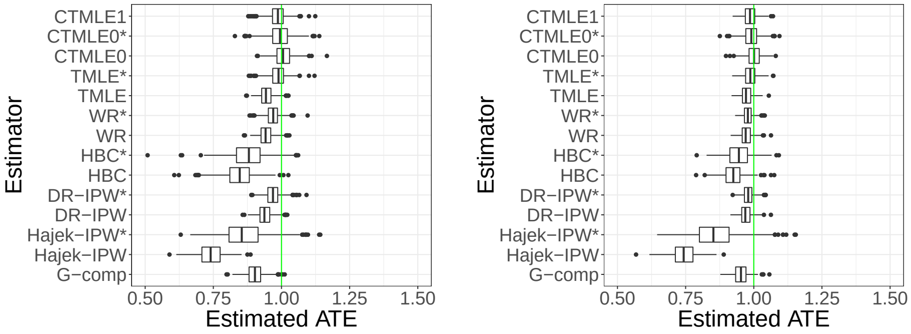


Figure 3.1: Boxplot of the estimated ATE for each estimator across 500 replications, when the initial estimate is fit on 10/20 out of 40 confounders.

Table 3.1: Performance of Point Estimation for Estimators when the initial estimate $\bar{Q}_n$ of $\bar{Q}_0$ is estimated on 10 and 20 out of 40 confounders. The results are computed based on simulations across 500 replications, each with a sample size of 1000 based on the NSAID study. All of the numeric values are on a scale of 10^{-2}.

Initial Fit		unadj	G-comp	WR	WR*	Hajek-BC	Hajek-BC*
10/40	Bias	-59.29	-9.69	-5.68	-3.11	-15.54	-12.29
	SE	8.43	3.36	2.66	2.75	5.80	6.63
	MSE	35.87	1.05	0.39	0.17	2.75	1.95
20/40	Bias	-59.91	-4.72	-2.77	-2.12	-7.56	-5.47
	SE	8.36	2.73	2.27	1.92	4.10	4.54
	MSE	36.59	0.30	0.13	0.08	0.74	0.51
Initial Fit		IPW	IPW*	Hajek-IPW	Hajek-IPW*	DR-IPW	DR-IPW*
10/40	Bias	95.43	128.97	-25.86	-13.61	-6.07	-3.12
	SE	36.55	91.38	4.85	8.21	2.63	3.02
	MSE	104.40	249.69	6.92	2.53	0.44	0.19
20/40	Bias	97.11	125.85	-25.60	-13.70	-2.92	-1.95
	SE	35.98	90.85	4.77	8.56	2.26	2.17
	MSE	107.23	240.75	6.78	2.61	0.14	0.09
Initial Fit		TMLE	TMLE*	CTMLE1	CTMLE0	CTMLE0*	oracle
10/40	Bias	-5.49	-1.23	-1.40	0.70	-0.64	0.36
	SE	2.57	3.46	3.56	3.38	4.40	1.83
	MSE	0.37	0.13	0.15	0.12	0.20	0.03
20/40	Bias	-2.68	-1.28	-1.38	0.08	-0.95	0.04
	SE	2.19	2.53	2.53	2.85	3.07	1.35
	MSE	0.12	0.08	0.08	0.08	0.10	0.02

mance. In addition, C-TMLE0* did not show any improvement compared to C-TMLE0. This is consistent with previous results [34].

We also evaluated the relative performance of other PS based estimators with g_n selected by C-TMLE, compared with g_n selected by CV. For IPW, the performance was still poor. However, for all of the other estimators that rely on the estimated PS, the performance improved considerably. Taking the first setting as an example, the relative empirical effi-

Table 3.2: Coverage of the 95% confidence intervals for semi-parametric efficient estimators when the initial estimate $\bar{Q}_n$ of $\bar{Q}_0$ is estimated on 10 and 20 out of 40 confounders. The results are computed across 500 replications, each with sample sizes of 1000 based on the NSAID study. All of the numerical values are multiplied by 100.

		CTMLE1	CTMLE0	CTMLE0*	DR-IPW	DR-IPW*	TMLE	TMLE*	oracle
10/40	Coverage	0.926	0.920	0.910	0.458	0.914	0.526	0.942	1.000
	Average Length	0.142	0.115	0.142	0.120	0.159	0.119	0.144	0.153
20/40	Coverage	0.934	0.872	0.898	0.748	0.928	0.790	0.946	1.000
	Average Length	0.105	0.087	0.103	0.088	0.112	0.087	0.106	0.111

ciency of DR-IPW* compared to DR-IPW was $\frac{\text{MSE(DR-IPW)}}{\text{MSE(DR-IPW*)}} = 1.52$, while for TMLE it was $\frac{\text{MSE(TMLE)}}{\text{MSE(TMLE*)}} = 1.66$. The relative empirical efficiency for both of these estimators is improved with a reduction in bias and slight increase in variance. These empirical results are consistent with previous theory [26, 34] showing that the model selected by external CV is usually under-fitted. These results illustrate the weakness of using “external” CV for PS model selection.

Confidence Interval

In this section, we evaluate the coverage and the length of the confidence intervals (CIs) for all the double robust estimators.

Table 3.2 shows that the CIs of the oracle TMLE estimator are too conservative, as they achieved 100% coverage. In both settings, TMLE* and C-TMLE1 had the best coverage. We can see that for other estimators, the length of the CIs were usually smaller/under-estimated. This resulted in a less satisfactory coverage even though the point estimation had similar performance (e.g. compare C-TMLE0 to C-TMLE1). With collaboratively selected g_n, the coverage of TMLE and DR-IPW improved significantly. These empirical results illustrate that a more targeted propensity score model selection can improve both causal estimation and inference.

Variable Selection from LASSO

Table 3.3: Average number of covariates selected from CV and C-TMLE. The number in the parentheses is the average number of selected confounders among the selected covariates

Initial Fit	CV	C-TMLE1
10/40	36.6 (13.2)	149.1 (35.1)
20/40	36.6 (13.2)	148.9 (31.4)

Table 3.3 shows the average number of covariates selected by LASSO, with λ determined by CV and C-TMLE1. Recall that there are 222 covariates in total, including 22 baseline

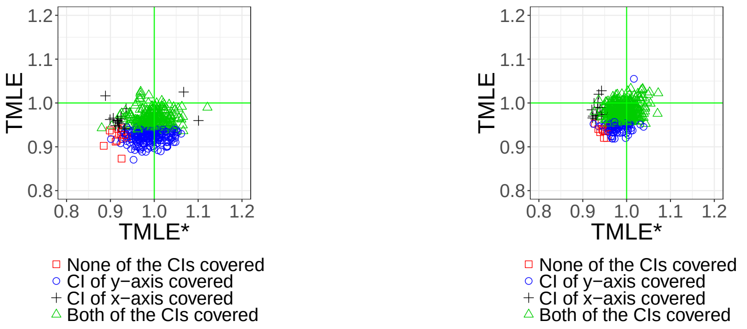
covariates and 200 covariates generated by the hdPS algorithm (see the introduction of the hdPS algorithm in subsection 2.3), including 40 confounders. CV was too conservative: on average it only selected 36.6 covariates, and only included 13.2 confounders. C-TMLE1 selected much less regularization, which leads to a larger model: it successfully picked up more confounders than CV in both experiments.

Pairwise Comparison of Efficient Estimators

In this subsection, we studied the pairwise comparisons for several pairs of the efficient estimators, TMLE, C-TMLE, and DR-IPW, with different PS estimators. The purpose of these pairwise comparisons is to help in understanding the contribution of the collaborative estimation of the PS. We used the shape and color of the points to represent the coverage information of the CIs for each estimates.

Impact of Collaborative Propensity Score Model Selection

We first compared the two pairs. Within the pair, both of the estimators were identical except each had a different PS estimator. The first pair compared TMLE to TMLE*, and the second pair compared C-TMLE0 to CTMLE0*.



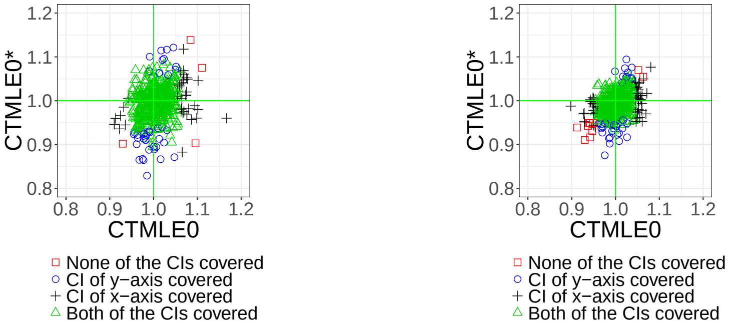
(a) Comparison of TMLE and TMLE*, with the initial estimate Q_n^0 adjusting for 10 out of 40 confounders.

(b) Comparison of TMLE and TMLE*, with the initial estimate Q_n^0 adjusting for 20 out of 40 confounders.

Figure 3.2: Comparison of TMLE wand TMLE*. The only difference within the pair the how the estimator g_n is selected

From figure 3.2a and 3.2b, we can see that a more targeted PS model contributes substantially to the estimation. The vanilla TMLE underestimated the ATE, while TMLE* is close to unbiased. The variance of the two estimators are similar.

From figure 3.3a and 3.3b we can see that the improvement for the CTMLE0 pair is not as significant as the improvement for the TMLE pair. Interestingly, most of the poor



(a) Comparison of C-TMLE0 and C-TMLE0*, with the initial estimate Q_n^0 adjusting for 10 out of 40 confounders.

(b) Comparison of C-TMLE0 and C-TMLE0*, with the initial estimate Q_n^0 adjusting for 20 out of 40 confounders.

Figure 3.3: Comparison of CTMLE0 and CTMLE0*. The only difference within the pair the how the estimator g_n is selected

performance in the CIs for CTMLE0 is from the over-estimated point estimate, while for CTMLE0* is mainly from under-estimation of the point estimate.

As discussed in [34], such ignorable improvement with collaboratively selecting g_n for the CTMLE0 pair might be due to the redundant collaborative estimation step. Thus, it is not necessary to both select the PS model using C-TMLE and solve for the extra clever covariate equation.

Contribution of Solving Extra Critical Equation

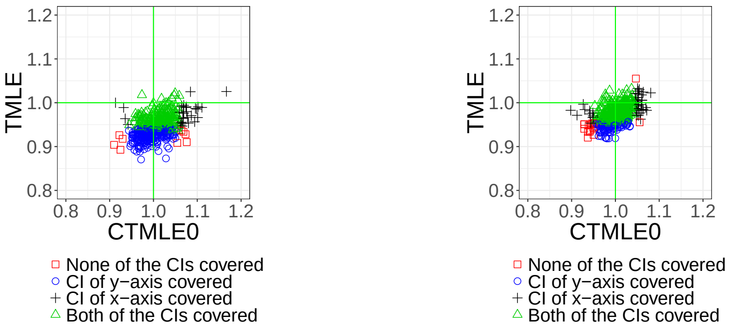
We compared TMLE with C-TMLE0. The only difference between these two estimators is that C-TMLE0 solves for the extra clever covariate equation, which guarantees that the critical equation is solved.

Figure 3.4 shows the improvement of solving an additional clever covariate. C-TMLE0 is less biased compared with TMLE. It is interesting to see that the performance of the estimator can improve substantially with such small change. In addition, this additional change almost requires no additional computation, which makes it more favorable among proposed C-TMLEs when the computation resources are limited.

Comparison of Variations of C-TMLE

We compared the two pairs of variations of C-TMLEs. We used C-TMLE1 as the benchmark, as it gave the best performance for both point estimation and confidence interval coverage.

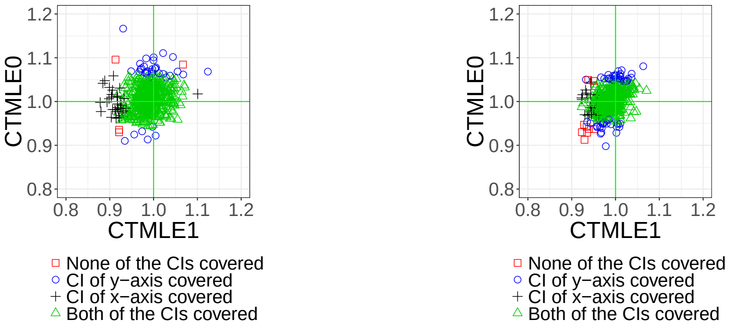
Figure 3.5a and 3.5b show the pairwise performance of C-TMLE1 and C-TMLE0. Both estimators performed well with respect to the MSE. Although the distribution of points looks similar and have variances that appear similar, there were more CIs from C-TMLE0



(a) Comparison of TMLE and C-TMLE0, with the initial estimate Q_n^0 adjusting for 10 out of 40 confounders.

(b) Comparison of TMLE and C-TMLE0, with the initial estimate Q_n^0 adjusting for 20 out of 40 confounders.

Figure 3.4: We compared TMLE with C-TMLE0, where the only difference between the two estimators is that C-TMLE0 solves the extra critical equation with additional clever covariates.



(a) Comparison of C-TMLE1 and C-TMLE0, with the initial estimate Q_n^0 adjusting for 10 out of 40 confounders.

(b) Comparison of C-TMLE1 and C-TMLE0, with the initial estimate Q_n^0 adjusting for 20 out of 40 confounders.

Figure 3.5: We compared C-TMLE1 with C-TMLE0.

that failed to cover the truth. In addition, the failures from C-TMLE1 mainly resulted from the under-estimation of the estimates. In comparison, the failures from C-TMLE0 primarily came from both under/over-estimated estimates. This suggests that the relatively poor CI coverage of C-TMLE0 might be due to its under-estimated standard error.

3.4 Applications in Electronic Healthcare Database

In this section, we applied the methods described previously to the NSAID study. As discussed previously, the goal of this study is to compare the effectiveness of two treatments on improving the risk (probability) of being diagnosed with severe gastrointestinal complications during the follow-up period. The treatment group was prescribed a selective COX-2 inhibitor, while the control group was prescribed a non-selective Nonsteroidal anti-inflammatory drug. To compare the safety of the two treatments, we used the average treatment effect (ATE) as our target parameter.

Method

We followed the hdPS procedure in subsection 2.3, where we generated the hdPS covariates with $k_1 = 100$ and $k_2 = 200$.

We investigated three kinds of initial estimate $\bar{Q}_n^0$ for TMLE and C-TMLE:

- The initial estimate was given by the group means of the treatment and control group.
- The initial estimate was estimated by Super Learner with only baseline covariates.
- The initial estimate was estimated by Super Learner with both baseline covariates and hdPS covariates.

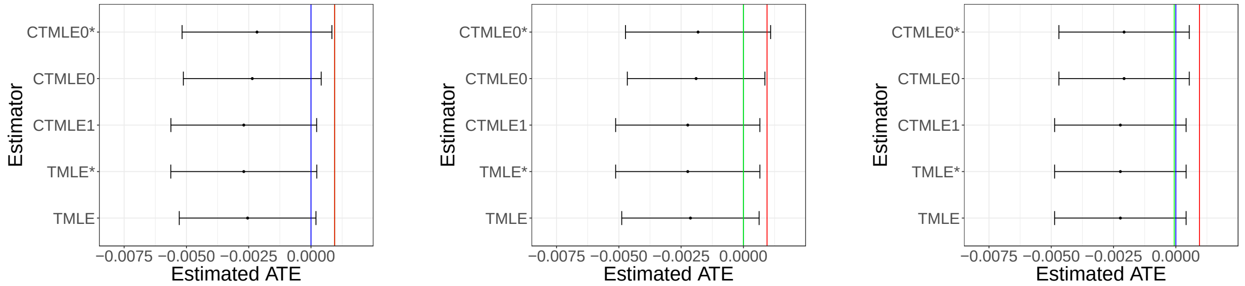
For Super Learners [36, 49], we used library with LASSO [13], Gradient Boosting Machine [53], and Extreme Gradient Boosting [5].

Results

Figure 3.6 shows the point estimates and 95% CIs for all TMLE and C-TMLE estimators. We use the blue line to denote the null hypothesis ($\mathrm{H}_0 : \Psi_0 = 0$), the green line denotes the initial estimate, and use red line to denote the results from the naive difference in means estimator ($\Psi_n^{\text{naive}} = 0.0949\%$).

Figure 3.6c shows that, after adjusting for selection bias using the TMLE/C-TMLE algorithms, all the estimators have similar results, with the estimated ATE being in the negative direction. Similar to the results in simulation, the CIs for TMLE* and C-TMLE0* were wider with PS estimator selected by C-TMLE1, than with PS estimator selected by CV. The details of the point estimates and confidence intervals are reported in table 3.4. We computed the analytic influence curve based confidence interval. None of these intervals, except C-TMLE0*, covered the naive estimate. However, all of them covered the null hypothesis.

In addition, we also compared the results from different initial estimator. Figure 3.6 shows the results for all estimators, with group means (3.6a), Super Learner with baseline covariates (3.6b), and Super Learner with both baseline and hdPS covariates (3.6c). The CV.LASSO



(a) Influence curve based confidence interval for all TMLE based estimators for NSAID study, with the group means as initial estimate.

(b) Influence curve based confidence interval for all TMLE based estimators for NSAID study, with initial estimate provided by Super Learner with baseline covariates.

(c) Influence curve based confidence interval for all TMLE based estimators for NSAID study, with the initial estimate provided by Super Learner with baseline covariates and hdPS covariates.

Figure 3.6: Confidence intervals for TMLE based estimators for the NSAID study.

Table 3.4: The point estimates and confidence intervals for all TMLE/C-TMLE estimators. All the values are on a scale of 10^{-2}.

names	TMLE	TMLE*	CTMLE1	CTMLE0	CTMLE0*
Point Estimate	-0.2381	-0.2491	-0.2491	-0.2208	-0.2093
Analytic SE	0.1414	0.1487	0.1486	0.1417	0.1502

PS estimator selected 137 covariates, with regularization parameter $\lambda = 0.001159$. The C-TMLE estimator with naive initial estimate selected 164 covariates, with $\lambda = 0.000266$. The C-TMLE estimator uses the initial estimate provided by SL with only baseline covariate have similar results: it selected 166 covariates with $\lambda = 0.000238$. For the C-TMLE with initial estimate provided by SL with all covariates, it selected the same model as CV.LASSO. It shows when the initial estimate is biased, C-TMLE selected model with less regularization, thus adjusted more potential confounders. In addition, all the covariates that included by LASSO selected by C-TMLE but not by CV.LASSO are hdPS covariates. This suggests such additional hdPS covariates can be confounder. However, as they have relatively weaker predictive performance for treatment mechanism, they would be mistakenly removed by CV.LASSO.

Figure 3.7 shows the details of the CV loss for each selected PS estimator. The blue line is the λ selected by C-TMLE1 with naive estimator. Its CV binomial deviance (twice the binomial negative log-likelihood) is 1.199632. The purple line is the λ selected by C-TMLE1 with initial estimator provided by SL with only baseline covariates. Its CV binomial deviance is 1.199668. The red line is the λ selected by CV.LASSO, and C-TMLE1 with initial estimator provided by SL with both baseline and hdPS covariates. Its CV binomial deviance

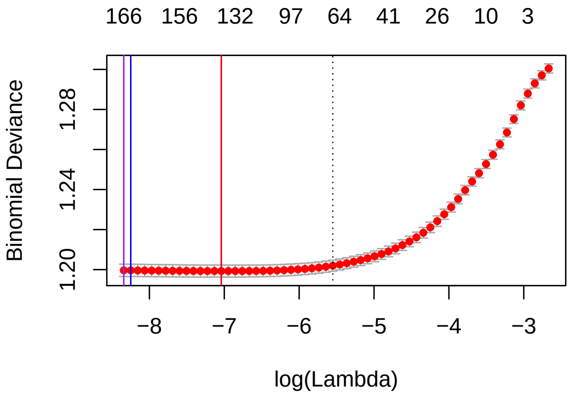


Figure 3.7: Binomial deviance for λ selected by CV.LASSO and C-TMLE with different initial estimators.

is 1.199288.

The estimates and confidence intervals were similar even with different initial estimators. This may be due to the signals in all the initial estimates are too weak: all the initial estimates of ATE are very close to 0. In addition, all the confidence intervals covered null hypothesis. The additive treatment effect in this study is not statistically significant.

Conclusions from the Empirical Study

Patients who received selective COX-2 inhibitors were less likely to get severe gastrointestinal complications during the follow-up period, compared to the patients who received a non- selective nonsteroidal anti-inflammatory drug. The average additive treatment effect was approximately -0.249%, which was estimated using TMLE* and C-TMLE1 (the two estimators achieved the best performance in simulations). The point estimates for other estimators were similar.

Based on the results, the additive treatment effect was not statistically significant. However, this does not necessary imply that there is no difference between the two treatments. More observations or better designed studies are necessary for further comparison of these treatments.

3.5 Discussion

In this study, we described two variations of C-TMLE, and assessed their performance on quasi-experiments based on real empirical data. We assessed the performance of several well studied PS-based estimators in settings where estimated models for both the conditional response $\mathbb{E}(Y|A,W)$ and the propensity score $\mathbb{E}(A|W)$ were misspecified. In particular, we focused on using the LASSO estimator for the PS model. In comparison to our previous work, this study provides a more detailed evaluation of all the estimators by not only assessing their point estimation, but also the confidence intervals for each of the estimators. Results showed that the C-TMLE1 and C-TMLE0 estimators had the best performance in terms of both point estimation and CI. We also evaluated the impact of directly applying the model that was collaboratively selected by C-TMLE1 to other PS non-collaborative estimators. Results showed that all of the PS-based estimators, except the vanilla IPW estimator, improved substantially, in terms of the point estimation, when the collaboratively selected model was applied to these estimators. However, C-TMLE0* did not improve when compared to C-TMLE0 for point estimation. Finally, pairwise comparisons of estimators were also evaluated to help in understanding the contribution of the collaborative model selection.

In comparison to previous work, this study is the first to thoroughly investigate and compare the confidence intervals coverage and length for the novel C-TMLE algorithms, as well as some commonly used competitors. Further, it offers detailed pair-wise comparisons with other competing estimators using different PS model selection procedures. Finally, this study utilizes the quasi-experiments based on a real electronic healthcare dataset and then makes inference on the same database. This makes the conclusions from the real data analysis more convincing.

In conclusion, this study introduces a new direction for PS model selection. It shows the insufficiency of using “external” cross-validation for the LASSO estimator. Thus, we conclude that the ensemble PS estimators, which rely on “external” cross-validation, are not optimal (w.r.t. the causal parameter) for maximizing confounding control. Ensemble learning that is based on C-TMLE is a potential solution to address this issue. We leave this for the future work.

Chapter 4

Adaptive Propensity Score Truncation

The positivity assumption, or the experimental treatment assignment (ETA) assumption, is important for the identifiability for estimating the average treatment effect (ATE). The positivity assumption requires $0 < \bar{G}_0(W) < 1$ for W almost everywhere, where $\bar{G}_0(W)$ is the PS (the probability to be assigned in the treatment group conditional on the pre-treatment baseline covariate vector W). Intuitively, this assumption guarantees that there exist samples in both treatment and control group for each sub-population, so the information for the corresponding potential outcome is available. However, even if the assumption is valid for the true data generating distribution, the randomness in data generating/sampling might cause practical violation of the positivity assumption. For example, there might be few or even no observations in a certain sub-population that are exposed to treatment. This usually challenges the estimation of the treatment effect for this sub-population. For example, it causes extreme values in the PS estimate, which jeopardizes the performance of the PS-based estimators.

Many approaches have been proposed and studied to address practical positivity violations. [47] systematically reviewed several commonly used practices. One simple and practical method is truncating extreme values in the PS estimate[51, 65, 6]. [1] proposed an algorithm that selects the truncation level for the inverse propensity score weighted (IPW) estimator by minimizing its estimated mean squared error (MSE).[43] further studied the sensitivity of a particular PS weighting estimator of ATE, with the PS estimated by four machine learning algorithms, and truncated at different cutpoints. Based on [1], [78] proposed and compared several adaptive truncation methods for marginal structural Cox models. Exclusion of problematic Ws which result in practical positivity violations (restricting the adjustment set [47]) is another commonly used approach [1, 47]. While removing such covariates might increase the bias of the causal estimator from confounding, it usually substantively reduces the variance. Sample trimming (restricting the sample [47]), which discard classes of subjects with limited variability in the observed treatment assignment, is another well-studied approach and has been widely used, especially in the econometrics and social science literature [9, 21, 41, 7].

In this study, we focus on the truncation method to address practical positivity violations. In practice, the PS score is truncated either by a fixed range (e.g. with absolute value restricted in [0.025, 0.975]), or by a fixed percentile (e.g. with value restricted in [0.1, 0.9] percentile): [32] studied the impact of arbitrary cutoffs of the PS at a fixed value for multiple estimators, and [6, 43] investigated the bias-variance trade-off with different truncation percentiles for propensity score weighting estimators. However, it is reasonable to believe that such fixed truncation strategies may not be not efficient. As the optimal cutoff depends on the choice of the PS estimator, the choice of the causal estimator, and the observed data, it impossible to know the optimal cutpoint a priori. It is reasonable to believe data-adaptive truncation methods would improve the finite sample performance of the causal estimator. We extend the collaborative targeted maximum likelihood estimation (C-TMLE) methodology to data-adaptive PS truncation. Developed based on targeted maximum likelihood estimation (TMLE) [39], C-TMLE inherits all the attractive properties of TMLE (e.g. doubly robustness, plug-in estimator) [40]. TMLE has been widely studied and applied in a wide range of topics, including causal inference and genomics [16], survival analysis [68], and safety analysis [44]. [31] proposed scalable C-TMLE by replacing the greedy search in [16] with a user-supplied ordering, and applied this to high-dimensional electronic healthcare data. [50] shows C-TMLE is more robust than TMLE. Recently, [34] developed C-TMLE algorithms for continuous tuning parameter, with the general theorem of the asymptotic normality of the resulting C-TMLE estimators. Based on this work, [34, 29] further proposed LASSO-C-TMLE, where the PS is estimated by LASSO controlled by C-TMLE, and [29] demonstrated its performance on high-dimensional electronic health dataset. We simply consider the truncation quantile γ as a tuning parameter, and extend the C-TMLE algorithm to select the optimal γ for the estimation of the causal parameter.

4.1 Brief review of the framework for causal effect estimation

For simplicity, we model the data generating distribution with a non-parametric structural equation model (NPSEM). Consider each observation, $O_i = (Y_i, A_i, W_i)$, is independently generated from the following data generating system:

$$\begin{cases} W = f_W(U_W), \\ A = f_A(W, U_A), \\ Y = f_Y(A, W, U_Y), \end{cases},$$

where f_W, f_A and f_Y are deterministic functions and U_W, U_A, U_Y are background (exogenous) variables. Each observation is drawn from a data generating distribution: first generate (U_W, U_A, U_Y), then compute W based on U_W, then determine the treatment assignment A based on (W, U_A). Finally compute outcome Y based on (A, W, U_Y). A is a binary indicator for treatment. Then the potential outcome (Y_1, Y_0) could be obtained by intervening on the treatment A in f_Y with 1 or 0:

$$\begin{aligned} Y_i^1 &= f_Y(A_i = 0, W_i, U_{Y,i}), \\ Y_i^0 &= f_Y(A_i = 1, W_i, U_{Y,i}), \end{aligned}$$

where $U_{Y,i}$ is the U_Y for i-th observation, which implies the consistency assumption:

Assumption 1 (Consistency Assumption)

$$Y_i = Y_i^{A_i} = Y_i^0(1 - A_i) + Y_i^1 A_i.$$

We consider the target parameter of the average treatment effect (ATE):

$$\Psi_0 = \mathbb{E}(Y^1) - \mathbb{E}(Y^0),$$

which could be interpreted as the difference between the expectations of the outcome if all the units received treatment, $\mathbb{E}(Y_1)$, versus if all the units did not receive the treatment, $\mathbb{E}(Y_0)$. We further assume background variables are independent $U_W \perp\!\!\!\perp U_A \perp\!\!\!\perp U_Y$, which is a sufficient condition for the conditional randomization assumption:

Assumption 2 (Conditional Randomization)

$$(Y^0, Y^1) \perp\!\!\!\perp A | W.$$

We also need the positivity assumption, or the experimental treatment assignment (ETA) assumption:

Assumption 3 (The Positivity Assumption)

$$0 < \bar{G}_0(W) < 1$$

almost everywhere.

This assumption means that for each subject in the target population, the probability of being assigned to the treatment/control group should be positive, given all the confounders W. We will discuss assumption 3 in more detail in next subsection.

The Importance of the Positivity Assumption

The positivity assumption 3 requires the probability of treatment to be bounded away from 0 and 1, given the smallest subset of observed potential confounders W that makes assumption 2 valid. Notice the propensity score need only conditioning on the covariates required for the conditional randomization assumption. For instance, conditioning on instrumental variables that are predictive for A while not for $(Y^{(1)}, Y^{(0)})$ would not help correcting the bias, and are would then be unnecessary. This is weaker than requiring that all subjects have had practical access to both levels of treatment.

Intuitively, if all the units in a certain sub-population were only assigned to the treatment (control) group, we would never get the information of the potential outcome corresponding to the control (treatment) group for this sub-population. This leads to the non-identifiability of the ATE of the whole population. [47] studied and discussed the estimator-specific behavior of several widely used estimators when the positivity assumption is violated.

Even if the positivity assumption holds in the (unknown) true data generating distribution, it is still possible that there are practical violations (or random violations [75]) of the positivity assumption due to randomness in the data generation. For example, consider a case where the probability that subjects in a subgroup receive the treatment is extremely low. Then only very few, or even none of such subjects in a given study sample are observed to receive the treatment, which makes it challenging to make inference for this subgroup [6, 78]. [75] illustrated the practical positivity violation by a small observational study of daily aspirin intake for prevention of myocardial infarction, where no one aged 31 to 35 years was exposed by chance. In this case, the information of the potential outcome Y_1 for such subpopulation is totally missing.

Practical violations of the positivity assumption can cause poor finite sample performance as it can result in highly influential observations. Consider the case where there is only 1 unit with $W = w$ and low PS of treatment. Then this single individual is now providing all of the information about the potential outcome Y_0 in the strata $W = w$. For estimators that rely on the estimation of the conditional response $\mathbb{E}(Y|A,W)$, one of the potential outcomes Y_a is never observed for some (a, w) and thus may require unreliable extrapolation to regions of (a, w) that are not supported by the data [1]. For weighting based estimators, this individual usually gets a large weight, which leads to the high variance of the resulting causal estimator. In this study, we propose a novel algorithm that provides a stable estimation of the causal parameter when there exists extreme values in the estimated PS due to the practical violation of the positivity assumption.

Notation

We first use $Q(W)$ to denote the marginal distribution of W; $\bar{G}(W)$ to denote the conditional expectation of A given W, $\mathbb{E}(A|W)$, and $\bar{Q}(A,W)$ to denote conditional expectation of Y given (A,W), $\mathbb{E}(Y|A,W)$. We use Q_0, $\bar{G}_0$, and $\bar{Q}_0$ for the corresponding part in the true data generating distribution P_0 of O_i, and use Q_n, $\bar{G}_n$ and $\bar{Q}_n$ to denote the corresponding estimate trained on the whole observed data.

For simplicity, we introduce two loss functions. The first one, $L^{(1)}$, is defined for the conditional outcome $\bar{Q}_0$. One example of the loss function for the estimate $\bar{Q}$ with outcome $Y \in [0,1]$ is:

[1] If $\bar{Q}_n$ is based on a correctly specified parametric model, this extrapolation will be accurate. However, in general we do not have correctly specified parametric models.

$$L^{(1)}(\bar{Q})(O_i) = -\big(Y_i \log(\bar{Q}(A_i, W_i)) + (1 - Y_i)\log(1 - \bar{Q}(A_i, W_i))\big) \tag{4.1}$$

The second one, $L^{(2)}$, is defined for the propensity score $\bar{G}_0$. One example of the loss function for the estimate $\bar{G}$ with binary treatment indicator A is:

$$L^{(2)}(\bar{G})(O_i) = -\big(A_i \log(\bar{G}(W_i)) + (1 - A_i)\log(1 - \bar{G}(W_i))\big) \tag{4.2}$$

In addition, we use $\hat{\bar{G}}_\gamma(\tilde{P}_n)$ to denote the resulting PS estimate by fitting estimator $\hat{\bar{G}}$ (e.g. main term logistic regression) of $\bar{G}_0$ on the training data with a given empirical distribution $\tilde{P}_n$ (e.g. the empirical distribution for the training subsamples), and truncated at γ percentile. Notice we have $\bar{G}_{n,\gamma} = \hat{\bar{G}}_\gamma(P_n)$, where P_n is the empirical distribution of all the observed units. We directly use empirical distribution Q_n to estimate Q_0.

4.2 Data-adaptive Truncation

A consequence of PS truncation is the introduction of bias in the estimated PS, which in turn causes bias in PS-based causal estimators [78]. Thus, PS truncation requires a bias-variance trade-off: too much truncation can make estimators more stable but also introduce more bias [6, 1]. [6] studied the bias-variance trade-off of the PS truncation by progressively truncating the PS weights at different quantiles. However, the optimal truncation varies for different datasets, and is usually unknown. Thus, it is important to define an empirical metric to select the cutpoints for truncation in a data-adaptive manner. Ideally, the optimal cutpoints should be selected minimizing the loss function (e.g. MSE) of the resulting causal estimator. However, the true MSE is not accessible in practice. [1] proposed a closed-form estimate for the expected MSE of a truncated IPW estimator. However, it is difficult to generalize this closed-form MSE estimator to TMLE.

In this study, we propose a data-adaptive method to select the quantile for truncating the PS estimate specially designed for TMLE. In subsection 4.2, we first describe a straightforward cross-validation (CV) selector for cutpoint selection. In subsection 4.2, we discuss the drawbacks of a model-free CV-selector, and present the Positivity-C-TMLE algorithm for cutpoint selection.

For simplicity, we only consider the case where the practical violation of positivity is one-sided. In other words, if we use inverse propensity score weighted (IPW) estimator, almost all the extreme weights are from the units in the control group where the estimated PSs are close to 1. In this case, we only consider the one-side truncation, which could be defined as:

$$\bar{G}_{n,\gamma}(W_i) = \min(\bar{G}_n(W_i), q_\gamma(\bar{G}_n))$$

where $q_\gamma(\bar{G}_n)$ is the γ quantile for the empirical distribution of $\bar{G}_n$.

Notice the framework for one-side truncation could be easily extended to two-side truncation, by adaptively selecting two truncation points.

Data-adaptive Truncation with Cross-validation for $\bar{G}_0$

One of the most straightforward methods to select the cutpoint is cross-validation. Consider the V-fold cross validation:

- Randomly split all the observed data into V groups with similar group size.
- Let $B_n \in \{0,1\}^n$, a random binary vector with length n, be a cross-validation scheme.
- Define the distribution of B_n as a discrete uniform distribution over V potential values. For the v-th potential value of B_n, we set the coordinates corresponding to the observations in the v-th fold to be 1, and all the others to be 0.

Let P^0_{n,B_n} be the empirical probability distribution of the training subsample $\{O_i : B_n(i) = 0, 1 \leq i \leq n\}$ and P^1_{n,B_n} be the empirical probability distribution of the validation subsample $\{O_i : B_n(i) = 1, 1 \leq i \leq n\}$. The cross-validation selector of γ is then defined as

$$\gamma_{n,\mathrm{CV}} \equiv \arg\min_{\gamma \in \Gamma} E_{B_n} P^1_{n,B_n} L^{(2)}(\hat{\bar{G}}_\gamma(P^0_{n,B_n}))$$

where Γ is the set of potential cutpoints γ. $L^{(2)}$ can be any binary loss function, and in this study we used a commonly used one, the negative log-likelihood loss function in equation (4.2).

Data-adaptive Truncation by the Stability of Ψ_n

The CV-selector for $\bar{G}_0$ has the following drawbacks:

- The objective function of CV merely focuses on the predictive performance of $\bar{G}$. In other words, it does not apply any knowledge of the target parameter.
- In addition, such CV procedure is "model-free". It selects the cutpoint independently (without regard to the causal parameter/estimator), and then plugs the resulting estimate of $\bar{G}_0$ into the estimator of the causal parameter. It is reasonable to believe different estimators of different causal parameters might have different optimal cutpoints. For example, the vanilla IPW estimator [23] might need more truncation (lower cutpoint in our setting), compared to the stabilized Hajek-type IPW estimator [20] [2].

To overcome this, it is important to consider a better empirical metric on the parameter of interest (e.g. MSE for the causal parameter). However, this is hard to achieve, as the value of the causal parameter is unknown. [1] proposed a closed-form estimate for the MSE of the IPW estimator $\hat{\Psi}_\gamma$ that uses the estimated PS truncated at γ:

$$\mathrm{MSE}(\hat{\Psi}_\gamma) = \mathrm{Var}(\hat{\Psi}_\gamma) + \mathrm{Bias}^2(\hat{\Psi}_\gamma).$$

[2] The definition of the vanilla and stabilized Hajek-type IPW estimator can be found in section 4.4

It then selects the truncation level that minimizes the estimated MSE. However, this closed-form estimate is hard to extend to more complicated estimators, like TMLE. [78] extended this work by a repeated two-fold cross-validation approach: the first part of the MSE estimate, $\mathrm{Var}(\hat{\Psi}_\gamma)$, is estimated by the variance estimate of the causal estimator. The second part, $\mathrm{Bias}^2(\hat{\Psi}_\gamma) = (\Psi_{n,\gamma} - \Psi_0)^2$, is estimated by the following procedure:

1. Randomly split data into two disjoint halves.
2. Compute $\Psi_{n,\gamma}$ on one of the halves with truncation level γ, and compute $\tilde{\Psi}$ on the other data.
3. Use $\widehat{\mathrm{Bias}}^2(\hat{\Psi}_\gamma) = (\Psi_{n,\gamma} - \tilde{\Psi})^2$ to estimate $\mathrm{Bias}^2(\hat{\Psi}_\gamma)$.

[78] suggested repeating the above procedure k times and taking the average of the bias estimates to stabilize the result.

Note the authors called this procedure “cross-validation” . To distinguish it from the conventional CV procedure mentioned in the previous subsection, we call it multi-view validation (MV) in our paper.

Data-adaptive Truncation with Collaborative Targeted Learning

In this subsection, we propose a new algorithm called Positivity-C-TMLE. It is specially designed for the TMLE estimator. We first introduce targeted minimum loss-based estimation, and then discuss this novel algorithm with details.

Brief review of Targeted Minimum Loss-based Estimation (TMLE)

Targeted minimum loss-based estimation (TMLE) is a general methodology to estimate a user-specified parameter of interest [39]. TMLE estimator is double robust, which means it is consistent as long as at least one of $\bar{G}_n$ and $\bar{Q}_n$ is consistent. In addition, TMLE estimator is efficient if both the input estimator $\bar{G}_n$ and $\bar{Q}_n$ are consistent.

In this study, we consider the TMLE for estimation of ATE, with the negative likelihood as the loss function, and logistic fluctuation. Then the TMLE algorithm can be written as:

Algorithm 5 Vanilla TMLE Algorithm for ATE, with negative log-likelihood loss and logistic fluctuation

```
function TMLE($\bar{Q}_n^0$, $\bar{G}_n$, $P_n$)
    Construct clever covariate $H_{\bar{G}_n}(A_i, W_i) = \frac{A_i}{\bar{G}_n(W_i)} - \frac{1-A_i}{1-\bar{G}_n(W_i)}$
    Fit a logistic regression: the outcome is $Y_i$, with logit($\bar{Q}_n^0$) as intercept, and $H_{\bar{G}_n}(A_i, W_i)$ as the univariate predictor, with coefficient $\epsilon$.
    Fluctuate the initial estimate: Given the logistic model above with fitted coefficient $\epsilon$ of $H_{\bar{G}_n}$, update the initial estimate by
        $\mathrm{logit}(\bar{Q}_n^*(A, W_i)) = \mathrm{logit}(\bar{Q}_n^0(A, W_i)) + \epsilon H_{\bar{G}_n}(A, W_i)$
    for $A \in \{0, 1\}$ and $i \in 1, \cdots, n$.
        return $\bar{Q}_n^*$ (an $n$ by 2 matrix).
end function
```

Then the resulting TMLE estimator for the ATE can be written as:

$$\Psi_n^{TMLE} = \frac{1}{n} \sum_{i=1}^{n} (\bar{Q}_n^*(1, W_i) - \bar{Q}_n^*(0, W_i)). \tag{4.3}$$

To construct a good input $\bar{Q}_n^0$ and $\bar{G}_n$ for algorithm 5, we suggest using Super Learner, a cross-validation based ensemble learning method. Super Learner could easily combine a set of individual machine learning algorithms, and has demonstrated outstanding performance in a wide range of tasks, including causal inference [48, 18, 62, 30, 77], spatial prediction [8], online learning [2], and image classification [25]. We refer the interested reader to the literature on Super Learner [36, 49].

In addition to the double robustness and asymptotic efficiency mentioned above, TMLE has following advantages:

1. Equation 4.3 shows that TMLE is a plug-in estimator, which respects the global constraints of the model by mapping the targeted estimate P^* (defined by $(\bar{Q}_n^*, Q_n)$) of P_0 into the target parameter Ψ. Note some other estimators (e.g. IPW) may produce estimates out of such constraints.

2. The loss function defined in TMLE, $L^{(1)}$ (the negative log-likelihood loss in algorithm 5), offers a metric to evaluate the goodness-of-fit of $(\bar{G}_n, \bar{Q}_n)$, directly w.r.t. the parameter of interest Ψ_0. In this example, the loss is the negative log-likelihood in the logistic regression step of algorithm 5.

3. Previous study shows TMLE is more robust to (near) positivity violations compared to IPW and A-IPW [50].

Positivity-C-TMLE

In this section, we briefly introduce the Positivity-C-TMLE algorithm, which is based on the general template of C-TMLE [40, 31]. The high-level description of of the C-TMLE algorithm is:

1. Sequentially generate a sequence of TMLE candidates $\bar{Q}^*_{n,\gamma}$ indexed by γ, each corresponding to a $\bar{G}_{n,\gamma}$ (here each $\bar{G}_{n,\gamma}$ is from the same PS estimate but truncated at different quantile γ).
2. Applying V-fold cross-validation to find the best TMLE candidate $\bar{Q}^*_{n,\gamma}$, which minimizes the CV risk for $L^{(2)}$ loss.

The input of the C-TMLE algorithm is a user-provided initial estimate $\bar{Q}^0_n$ for $\bar{Q}_0 = \mathbb{E}_0(Y|A,W)$ with the empirical distribution P_n of the observed data $O_i, i = 1, \ldots, n$. Following this template, with a user provided sequence of cutoffs $[\gamma_{\min}, \cdots, \gamma_{\max}]$ and the corresponding sequence of PS estimate $\bar{G}_{n,\gamma}$, the Positivity C-TMLE searches among the cutoffs, finds the γ^* that maximizes the empirical fit of TMLE using γ-specific clever covariate, updates the initial estimate to this TMLE, repeats this by maximizing over the remaining $[\gamma^*, \cdots, \gamma_{\max}]$ range, and proceeds till having reach the cutoff $\gamma_{\max}$. This generates a sequence of TMLEs, $\bar{Q}^*_{n,\gamma}$, for all γ.

We then select the $\bar{Q}^*_{n,\gamma}$ with CV using the $L^{(1)}$ loss for $\bar{Q}_0$: the sequence $\bar{Q}^*_{n,\gamma}$ for all γ defines a sequence of estimators that map data P_n into $\bar{Q}^*_{n,\gamma}$, so that we can run this mapping on a training sample P^0_{n,B_n} and then evaluate its performance on the validation sample P^1_{n,B_n}. The C-TMLE uses V-fold CV to select the best $\bar{Q}^*_{n,\gamma}$ among the generated TMLEs, with respect to the cross-validated predictive performance for $\bar{Q}_0$ with $L^{(1)}$ loss.

Algorithm 7 in the appendix shows the details of C-TMLE algorithm for cutpoint selection. [3]

For simplicity, C-TMLE in later sections also refers to the Positivity-C-TMLE described here.

4.3 Inference after Truncation

Influence Curve based Variance Estimator

We briefly review the influence curve based confidence intervals for TMLE and C-TMLE. The efficient influence curve (EIC) for the ATE parameter is given by

$$\begin{aligned} D^*(\bar{Q}, \bar{G}, \psi)(O_i) &= H(A,W)[Y - \bar{Q}(A,W)] \\ &\quad + \bar{Q}(1,W) - \bar{Q}(0,W) - \psi, \end{aligned}$$

[3] The Positivity-C-TMLE algorithm is almost identical to the LASSO-C-TMLE algorithm in [29, 34]. The only difference is the one-dimensional tuning parameter here is the truncation quantile, instead of the regularization parameter λ for LASSO.

where $H(A, W) = A/\bar{G}(1, W) - (1 - A)/\bar{G}(0, W)$ $(A = 0, 1)$ [59, 38]. Based on the estimated $\hat{\bar{Q}}$, $\hat{\bar{G}}$, and $\hat{\Psi}$, the variance of a TMLE/C-TMLE/A-IPW estimator is given by:

$$\widehat{\mathrm{Var}}(\hat{\Psi}) = \sum_{i=1}^{n} D^*(\hat{\bar{Q}}, \hat{\bar{G}}, \hat{\Psi})(O_i). \tag{4.4}$$

Robust Variance Estimator

In this section, we propose a robust CI based on the robust targeted variance estimator from [67, 72] for Positivity-C-TMLE estimator. Different from the variance estimator in (4.4), this variance estimator is a substitution estimator, and thus more stable when there are near practical positivity violations [38].

Recall that the expectation of the second moment of the efficient influence curve can be calculated as:

$$\mathbb{E}[D^*(\bar{Q}, \bar{G}, \psi)(O_i)]^2 = \mathbb{E}[H(A, W)[Y - \bar{Q}(A, W)]]^2 + \mathbb{E}[\bar{Q}(1, W) - \bar{Q}(0, W) - \psi]^2 \tag{4.5}$$

Given $\bar{Q}_n$, the second part can be estimated with:

$$\frac{1}{n}\sum_{i=1}^{n}[\bar{Q}_n(1, W_i) - \bar{Q}_n(0, W_1) - \Psi_n]^2 \tag{4.6}$$

where

$$\Psi_n = \frac{1}{n}\sum_{i=1}^{n}[\bar{Q}_n(1, W_i) - \bar{Q}_n(0, W_1)].$$

The first part can be decomposed as:

$$\mathbb{E}[H(A, W)[Y - \bar{Q}(A, W)]]^2 = \mathbb{E}(\frac{[Y_1(W) - \bar{Q}(1, W)]^2}{\bar{G}(W)}) + \mathbb{E}(\frac{[Y_0(W) - \bar{Q}(0, W)]^2}{(1 - \bar{G}(W))}).$$

Given the estimate $\bar{Q}_n$ and $\bar{G}_n$ (and the corresponding H_n), each of them can be represented as the mean of a counterfactual

$$S^a(W) = [Y^a(W) - \bar{Q}_n(a, W)]^2 \cdot H_n(a, W), a \in \{0, 1\},$$

with i-th observed outcome:

$$S_i(W) = (Y_i - \bar{Q}_n(A_i, W_i))^2 \cdot H_n(A_i, W_i).$$

Thus we proposed the following robust variance estimation procedure:

- Create transformed observations $\tilde{O}_i = (S_i, A_i, W_i)$, and feed it to a standard TMLE algorithm. This step outputs $\tilde{\bar{Q}}_n^*(A, W)$.

- Compute
$$\frac{1}{n}\sum_{i=1}^{n}[\bar{\tilde{Q}}_n^*(1, W_i) + \bar{\tilde{Q}}_n^*(0, W_i)]$$
as a robust estimate of $\mathbb{E}[H(A, W)[Y - \bar{Q}(A, W)]]^2$, the first part in equation (4.5).

- Finally combine this with (4.6) to compute the robust variance estimate:
$$\frac{1}{n}\sum_{i=1}^{n}[\bar{\tilde{Q}}_n^*(1, W_i) + \bar{\tilde{Q}}_n^*(0, W_i)] + \frac{1}{n}\sum_{i=1}^{n}[\bar{Q}_n(1, W_i) - \bar{Q}_n(0, W_1) - \Psi_n]^2$$

More details of the robust variance estimation can be found in [72].

4.4 Experiment

In this section, we designed simulation studies to assess the performance (bias, variance, and MSE) of several commonly used estimators. For each estimator, we studied different methods to determine cutpoint. Subsection 4.4 presents how data was generated for experiments. Subsection 4.4 reviews the estimators used in the experiments. Subsection 4.4 shows the results from the simulation, and compares the estimators with different empirical metrics for cutpoint selection. The R package *ctmle* [27] can be found on The Comprehensive R Archive Network.

Data Generating Distribution

We consider the following data generating distribution for $O_i = (Y_i, A_i, W_i)$: W_i is the vector of 20 baseline covariates, generated from weakly correlated multivariate normal distribution. The treatment indicator variable A_i is independently generated from a Bernoulli distribution, with:
$$P(A_i = 1|W_i) = \text{logit}[C - (W_{i1} + W_{i2} + \sum_{j=3}^{20}\frac{3}{20}W_{ij})].$$

Thus, the PSs would be closer to 1 with a larger value of intercept C.

Figure 4.1 and 4.2 shows the histogram plots of true propensity score, and estimated propensity score (by logistic regression) for $C = 1, 2$, with sample size $N = 1000$.

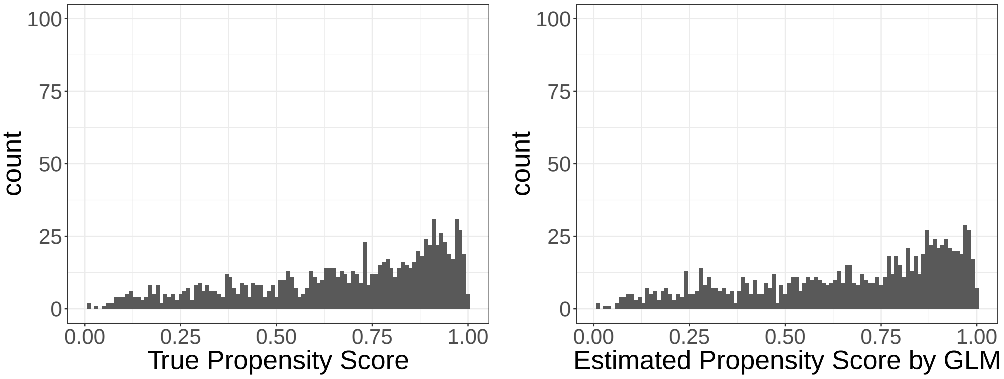


Figure 4.1: Histogram for the true PS and estimated PS for C = 1, Sample size N = 1000

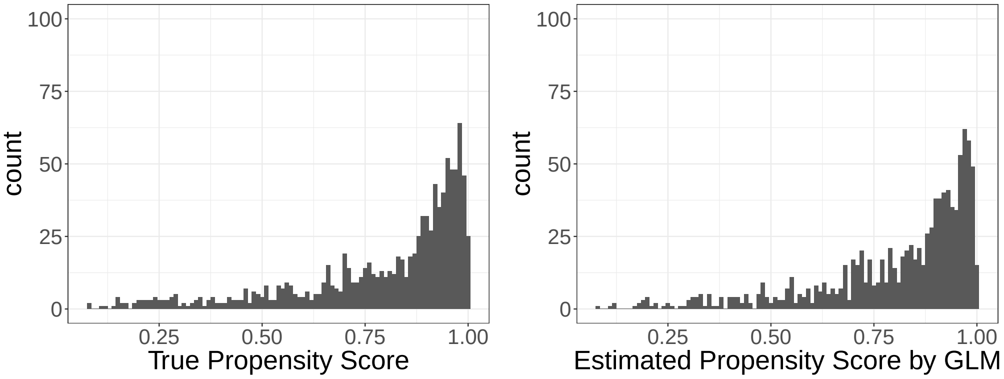


Figure 4.2: Histogram for the true PS and estimated PS for C = 2, Sample size N = 1000

The potential outcomes pair (Y_{i0}, Y_{i1}) is independently generated from a Gaussian distribution, with conditional expectations:

$$\mathbb{E}(Y_{i0} \mid W_i) = 2 + 2(W_{i1} + W_{i2} + W_{i5} + W_{i6} + W_{i8})$$

and

$$\mathbb{E}(Y_{i1} \mid W_i) = 4 + 2(W_{i1} + W_{i2} + W_{i5} + W_{i6} + W_{i8})$$

and the variance is 1 for both Y_{i0} and Y_{i1}. In other words, the observed outcome Y_i is from a normal distribution with variance 1 and expectation:

$$\mathbb{E}(Y_i \mid A_i, W_i) = 2 + 2(W_{i1} + W_{i2} + W_{i5} + W_{i6} + W_{i8}) + 2A_i.$$

Thus the true average treatment effect is 2.

Estimators

In the simulation, we compared several PS-based estimators. First, we consider the widely used inverse propensity score (IPW) estimator, or so-called Horvitz-Thompson estimator [23]:

$$\Psi_n^{IPW} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{A_iY_i}{\bar{G}_n(W_i)} - \frac{(1-A_i)Y_i}{1-\bar{G}_n(W_i)}\right].$$

IPW is a consistent estimator when $\bar{G}_n$ consistently estimates $\bar{G}_0$. However, due to the inverse weighting, the IPW estimator usually has overly large variance, when there exist some weights $A\bar{G}_n + (1-A)(1-\bar{G}_n)$ close to zero. To stabilize the IPW estimator, the Hajek-type IPW (Hajek-IPW) [20] was proposed as:

$$\Psi_n^{Hajek-IPW} = \sum_{i=1}^{n}\left[\frac{A_iY_i/\bar{G}_n(W_i)}{\sum_{i=1}^{n} A_i/\bar{G}_n(W_i)} - \frac{(1-A_i)Y_i/(1-\bar{G}_n(W_i))}{\sum_{i=1}^{n}(1-A_i)/(1-\bar{G}_n(W_i))}\right].$$

Hajek-type IPW is usually more stable compared to the plain IPW estimator. However, this stabilized IPW estimator will still be highly variable and will have a positively skewed distribution if there are very strong covariate-treatment associations [22, 78, 45].

Both of the above estimators only rely on estimation of the PS and will be inconsistent if the PS is not estimated consistently. We further compared the following double robust estimators. The Augmented-IPW (A-IPW, or DR-IPW) estimator [59] can be written as:

$$\Psi_n^{DR-IPW} = \frac{1}{n}\sum_{i=1}^{n} H_{\bar{G}_n}(A_i, W_i)\left[Y_i - \bar{Q}_n(A_i, W_i)\right] + \bar{Q}_n(1, W_i) - \bar{Q}_n(0, W_i)$$

where

$$H_{\bar{G}_n}(A, W) = \frac{A}{\bar{G}_n(W)} - \frac{1-A}{1-\bar{G}_n(W)}.$$

In this study, we also consider the vanilla TMLE estimator:

$$\Psi_n^{TMLE} = \frac{1}{n}\sum_{i=1}^{n}(\bar{Q}_n^*(1, W_i) - \bar{Q}_n^*(0, W_i)).$$

We consider the following estimators to estimate the causal parameter:

- Estimators with fixed truncation level: for all the estimators described above, we provided them with estimated PS truncated at different fixed percentile, from $\gamma = 60\%$ quantile, to $\gamma = 100\%$ quantile (no truncation), with step size 1%.

- Estimators with the truncation level selected by CV: for all the estimators described above, the truncation level for the PS estimate is selected by CV with negative log-likelihood loss on $\bar{G}$.

- TMLE estimator with truncation level selected by MV, or for short, MV-TMLE estimator.
- Positivity-C-TMLE estimator.

For A-IPW, TMLE and C-TMLE estimators which rely on the estimation of $\bar{Q}_0$, we used the estimate $\bar{Q}_n^0$ from a main terms linear regression, with observed outcome, Y, as dependent variable, and treatment, A, along with baseline covariates $W_3, \ldots, W_{10}$ as predictor. In other word, the confounding in the initial estimate is partially controlled. For the estimation of $\bar{G}_0$ for all PS-based estimators, we used a main terms logistic regression with all the covariates as predictors. In other words, the PS is estimated consistently and efficiently. Thus we guarantee the model is correctly specified, and the failure of the estimators in the simulations are from the practical violation of the positivity assumption instead of model misspecification.

For each of the following simulation settings, we generated the data from each corresponding data generating system 200 times independently, and report the average bias, standard error, and mean squared error of all the estimators.

Results

We use solid curves with different color to denote the estimators with different fixed quantiles as cutpoint for truncation. For all estimators with data-adaptive truncation, we use horizontal lines to present the performance.

Mean Squared Error

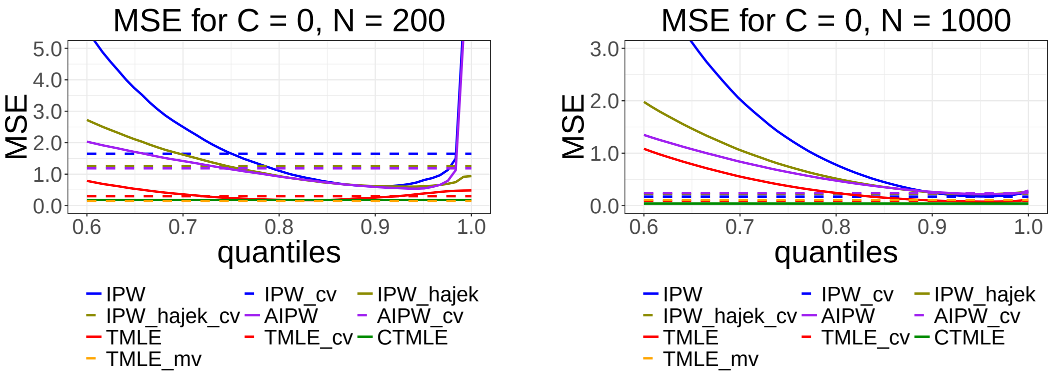


Figure 4.3: Comparison of the MSEs for each estimator with $C = 0$.

First we study the case $C = 0$. When the sample size is 200, small values of γ result in high MSE due to bias and large values of γ result in high MSE due to variance. The optimal cutpoint for different estimator varies. For IPW, IPW-Hajek, and A-IPW, the optimal was about $\gamma = 0.9$. It is not surprising to see the vanilla IPW estimator is the most unstable around $\gamma = 1$. TMLE is the most stable estimator, and it achieved optimal around $\gamma = 0.8$. Among all estimators with cutpoint selected by CV, TMLE performed best, and its MSE was very close to the MSE of C-TMLE. CV-TMLE, MV-TMLE, and C-TMLE have similar performance.

When the sample size is 1000, the optimal cutpoint for all estimators was close to $\gamma = 1$. Intuitively, the larger sample size make the variance of estimators smaller, thus less truncation is necessary. When the sample size is large enough, it would be unnecessary to truncate the PS estimate. All the estimators with cutpoint selected by CV had similar performance. The C-TMLE estimator achieved the best MSE and the CV-TMLE estimator achieved the second best MSE when $N = 1000$. For both $N = 200$ and 1000, the C-TMLE estimator was even better than the oracles of all the competing estimators with fixed quantile: the horizontal line for C-TMLE is below all the curves.

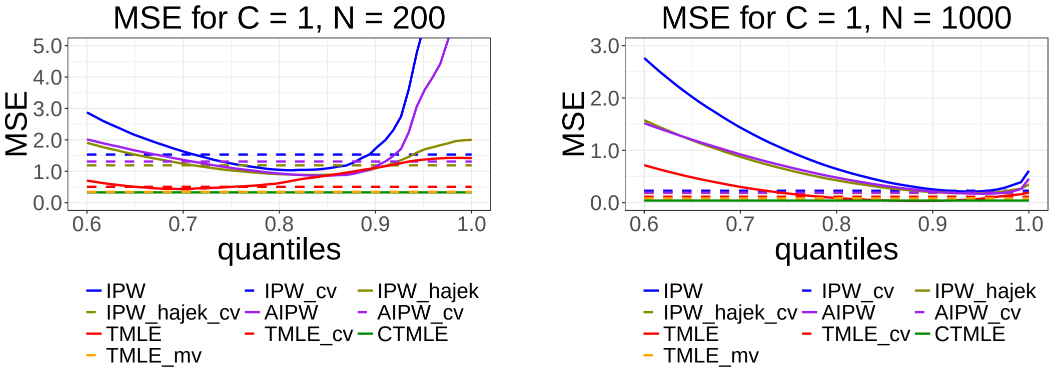


Figure 4.4: Comparison of the MSEs for each estimator with $C = 1$.

We then set $C = 1$ to introduce stronger practical violations of the positivity assumption. For $N = 200$, the IPW and A-IPW estimators became more unstable. The corresponding MSEs increased sharply when γ increased from 0.85 to 1. This might be because of the unstable inverse weighting in these two estimators. Hajek-type IPW was much more stable for mild truncation, in comparison to IPW and A-IPW. TMLE was more stable and had better performance compared to the previous estimators. For estimators with adaptive cutpoint selection, TMLE achieved the best performance among all the estimators with cutpoint selected by CV. MV-TMLE and C-TMLE had the best performance among all the estimators.

When $N = 1000$, all the estimators have similar performance with the previous case where $C = 0, N = 1000$. Due to the relatively large sample size, even the estimators with untruncated PS had satisfactory performance. However, we observe that, different from the case with $N = 1000$ and $C = 0$, the MSE for IPW starts increasing after $\gamma = 0.95$ when $N = 1000, C = 1$, which indicates there are stronger violations of the positivity assumption in this case. In this setting, C-TMLE still achieved the best performance among all estimators and was better than the oracles for all estimators with fixed cutpoint.

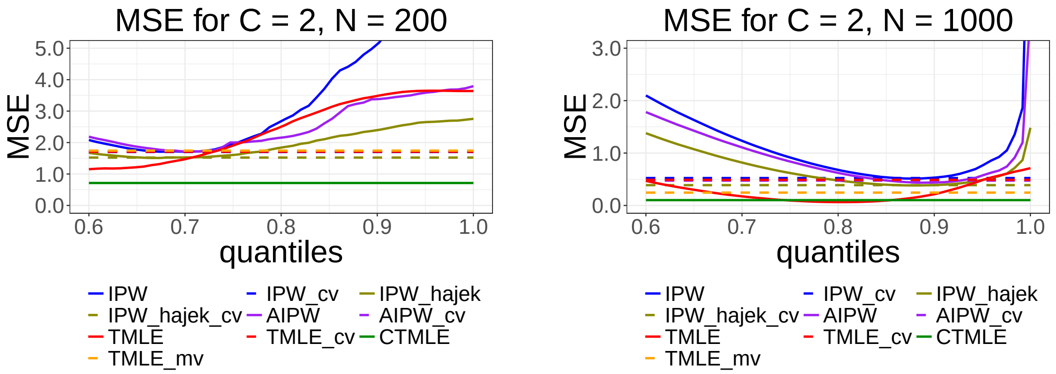


Figure 4.5: Comparison of the MSEs for each estimator $C = 2$.

Finally we studied the case where the positivity parameter $C = 2$. We could see from figure 4.2 that there was strong practical violation of the positivity assumption, as the distribution of the PS is highly concentrated around 1. For $N = 200$, MSEs for all estimators increased compared to the previous cases where $C = 0, 1$. The MSEs for IPW was out of the bound of the plot when the PS was truncated with large quantile. Hajek-type IPW estimator was much more stable compared to IPW and A-IPW in this case. TMLE still had satisfactory performance among all the non-adaptive estimators, and the optimal quantile for truncation of TMLE is around $\gamma = 0.6$. For the estimators with cutpoint selected by CV, Hajek-TMLE achieved the best performance. In this case, where there exist strong practical positivity violations, the gap between C-TMLE estimator and other estimators became larger.

Similar to the previous cases, larger sample size relieved issues from practical violations of the positivity assumption. When $N = 1000$, the optimal quantile for TMLE truncation increased to around $\gamma = 0.84$, while for all the other non-adaptive estimators the optimal quantile was around $\gamma = 0.9$. The estimators with cutpoint selected by CV had similar performance, with MSE around 0.4, and MV-TMLE estimator had slightly better performance. C-TMLE estimator had the best performance among all the adaptive estimators. The oracle for TMLE with fixed cutpoint is slightly better than C-TMLE when $C = 2, N = 1000$, but such optimal cutpoint is unknown in practice.

The Bias-Variance Trade-off

We further studied the bias and variance trade-off for each estimator.

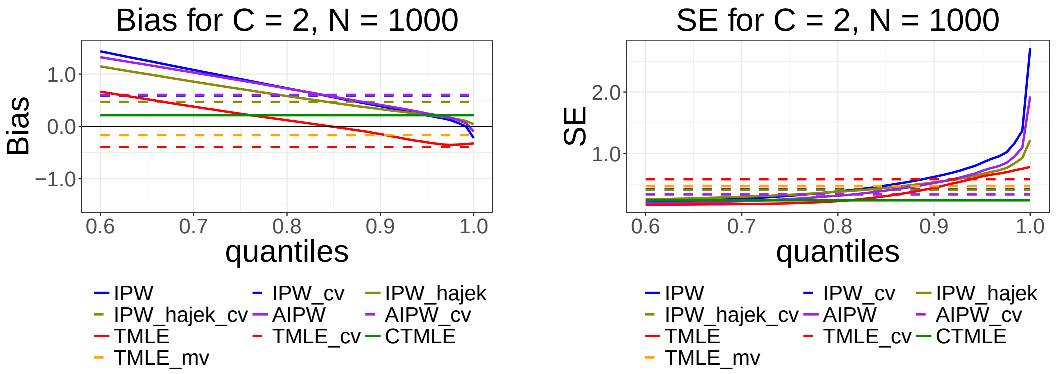


Figure 4.6: Comparison of the MSEs for each estimator $C = 2$. For bias plot (left), a horizontal line at 0 is added for better comparison.

Figure 4.6 shows the bias and the standard error (SE) for each estimator. The figure for bias shows that when the cutpoint is increased, IPW, Hajek-type IPW and A-IPW became less biased. The bias of TMLE decreased from positive to 0, and then became negative. This shows practical violations of positivity would introduce bias for TMLE when no truncation is applied to the PS estimate, even when using the true parametric-model for PS estimation. For the SE, all the estimators with fixed cutpoint show the same pattern: all the SE increase dramatically with truncation quantile increased from 0.8 to 1.0. For all estimators with adaptive cutpoint selection, C-TMLE achieved both the smallest SE and a relatively small bias. In comparison, MV-TMLE and CV-TMLE achieved small absolute bias, but had overly large variance. Among all estimators using CV for cutpoint selection, TMLE has the best MSE (see figure 4.5). More details can be found in table 4.1.

Table 4.1: Detailed results for the data-adaptive truncation methods.

N		$C = 0$			$C = 1$			$C = 2$		
	Estimator	Bias	SE	MSE	Bias	SE	MSE	Bias	SE	MSE
200	CV-TMLE	0.243	0.575	0.389	0.325	0.619	0.488	-0.067	1.471	2.163
	MV-TMLE	0.064	0.369	**0.140**	0.117	0.409	**0.181**	-0.272	1.065	1.206
	C-TMLE	0.042	0.459	0.212	0.193	0.423	0.216	0.062	0.962	**0.927**
1000	CV-TMLE	-0.033	0.294	0.087	-0.147	0.304	0.114	-0.392	0.581	0.489
	MV-TMLE	0.106	0.310	0.107	0.017	0.291	0.084	-0.167	0.467	0.245
	C-TMLE	0.022	0.196	**0.039**	0.070	0.189	**0.040**	0.214	0.237	**0.102**

To further study the estimators with data-adaptive truncation selection, we also compared MSE for each estimator with the positivity parameter C increasing from 0 to 2.

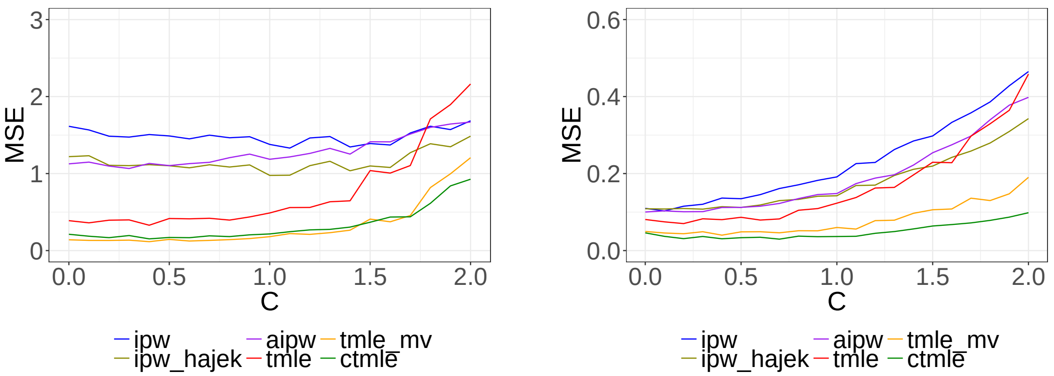


Figure 4.7: Comparison of C-TMLE, MV-TMLE, and the other estimators with cutpoint selected by CV. We varied C from 0 to 2. Left: sample size $N = 200$. Right: sample size $N = 1000$.

Figure 4.7 shows the trend of MSE for each estimator with the positivity parameter C increasing from 0 to 2. C-TMLE kept better performance compared to all the other estimators with cutpoint determined by CV. In addition, the gap between the MSE for C-TMLE and other estimators kept increasing. This suggests that CV is far from the optimal for the cutpoint γ selection.

MV-TMLE has good performance when $N = 200$ and C is small. However, in the setting of $N = 200$, when violations of positivity became stronger, its MSE increased dramatically after $C = 1.5$. When the sample size $N = 1000$, it keeps satisfactory performance. However, it is consistently weaker than C-TMLE across all C.

Comparison of Cutpoints for CV, MV-TMLE, and C-TMLE

To better understand the difference between the cutpoints γ selected by C-TMLE and CV, we study the mean of the quantiles selected for C-TMLE, MV-TMLE and CV. To have a better comparison, we used TMLE estimator with the cutpoint selected by CV (CV-TMLE) to compare with Positivity-C-TMLE (C-TMLE), and the cutpoint selected by MV (MV-TMLE).

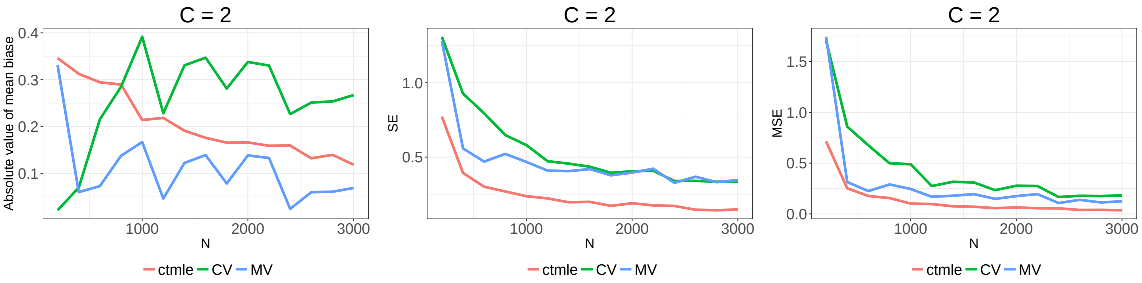


Figure 4.8: Fix positivity parameter $C = 2$, increase $N = 200$ to 3000

Figure 4.8 shows the absolute mean bias, SE, and MSE for CV-TMLE and C-TMLE with the positivity parameter $C = 2$, sample size N changing from 200 to 3000. MSE for both algorithms decreases, which is mainly due to the decreasing SE. The absolute mean bias for C-TMLE shows a decreasing trend, but not clear for CV-TMLE. This might be because CV is too sensitive to the sample size, and selected too mild truncation (too large cutpoint quantile γ). In addition, it is interesting to see that the bias curves of CV-TMLE and MV-TMLE show very similar patterns.

To better understand why C-TMLE outperforms CV-TMLE, we plot the mean cutpoint selected by CV and C-TMLE.

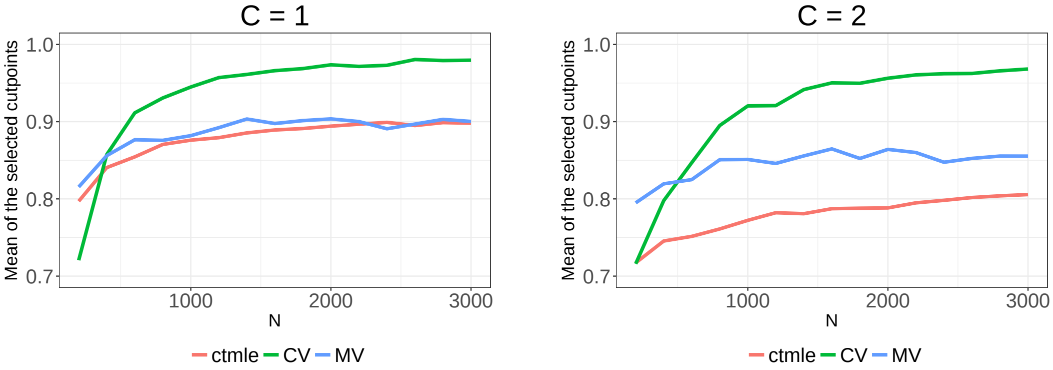


Figure 4.9: Mean of selected quantiles by CV, MV-TMLE, and C-TMLE for fixed $C = 1$ (left) and 2 (right), with sample size N increasing from 200 to 3000.

Figure 4.9 shows the mean quantile selected by CV and C-TMLE. In this experiment, we fixed $C = 1$ (left) and $C = 2$ (right), with sample size N increasing from 200 to 3000. We observe that CV is more sensitive to N in comparison to MV-TMLE and C-TMLE. The cutpoint increased dramatically from around 0.7 to 0.95, when N increased from 200 to 1000. However, C-TMLE tended to be more conservative. Even when the sample size is very large, it still only truncated at around 90%. On the other hand, comparing the two figures with $C = 1$ and $C = 2$, we could see C-TMLE is much more sensitive to the positivity parameter C. In comparison, the lines for CV for $C = 1, 2$ are more similar than the lines for C-TMLE. The cutpoint selected by MV-TMLE is not sensitive either to the sample size or the positivity parameter.

To better understand their behavior from another perspective, we fixed the sample size N and increased the positivity parameter C.

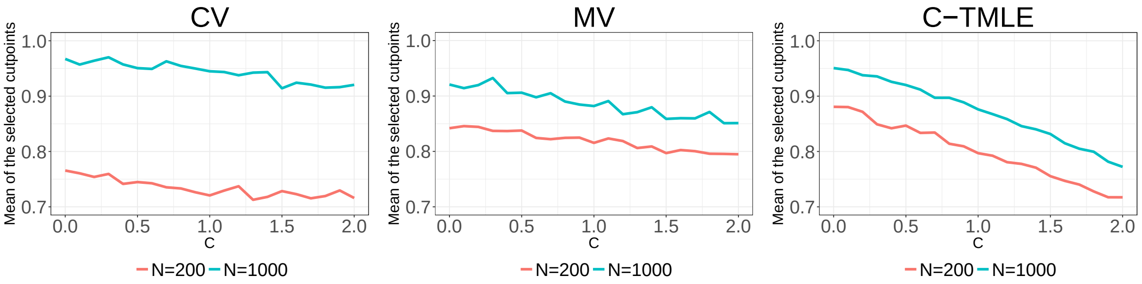


Figure 4.10: Mean of selected quantile by CV, MV-TMLE, and C-TMLE, with sample size $N = 200$ and $n = 1000$ and the positivity parameter C from 0 to 2.

Figure 4.10 shows that the cutpoint selected by C-TMLE is more sensitive to the positivity parameter C, as the curves for CV and MV-TMLE are flatter. This could be explained by the objective function used for CV: the commonly used negative log likelihood loss penalized the observations with:

$$L^{(2)}(\bar{G})(A_i, W_i) = A_i[\log(\bar{G}(W_i))] + (1 - A_i)[\log(1 - \bar{G}(W_i))]$$

Consider the case where the untreated observations are rare. Then for the untreated observations $A_i = 0$, but with high value of the estimated PS, $\bar{G}_n(W_i)$, it would contribute $-\log(1 - \bar{G}_n(W_i))$ to the loss function. However, the performance of the estimators with inverse weighing would suffer more in comparison to the predictive performance of $\bar{G}$, as the inverse of a very small number, $1/\bar{G}(W_i)$, can be much larger/influential. In this sense, the C-TMLE estimator has an attractive property that it determines the cutpoint by minimizing the CV loss for the parameter of interest, instead of the nuisance estimator.

It remains unknown why the cutpoint selected by MV-TMLE is not sensitive to either sample size, or the positivity parameter. Unlike CV, which is model free, MV relies on the choice of the causal estimator. Thus it is possible that the cutpoint would be more sensitive if we switch to a less robust estimator (e.g. IPW estimator).

For C-TMLE, notice this cutpoint selection is different from the general model selection problem. Unlike the general model selection (e.g. selection of the regularization parameter λ for LASSO), the cutpoint γ selection is not closely relevant to the bias-variance trade-off, or smoothness, of $\bar{G}$, as it only affects the tail distribution of $A \mid W$. The negative log-likelihood would always select little truncation (high cutpoint) as the increasing of bias is faster than the decreasing of variance, as the Kullback-Leibler divergence is not sensitive to predicted probabilities close to 0/1. Even $\bar{G}_{n,\gamma}$ selected by CV will yield an asymptotic linear estimator, without suffering from under-smoothing. Thus it does not fit the general theorem of C-TMLE in [34]. However, in the finite-sample cases, the Positivity-C-TMLE uses a more targeted criterion in comparison to CV, which leads to a better practical performance.

Confidence Intervals

In this section, we study the finite sample performance of confidence intervals for double robust estimators.

Table 4.2: Coverage of CI across 200 experiments with sample size 1000 for efficient estimators, with the average relative width of CI to CV-TMLE* in parentheses. Estimators with * use the true SE (provided by a separate Monte Carlo simulation), while the others use the estimated SE.

	$C = 0$	$C = 0.5$	$C = 1$	$C = 1.5$	$C = 2$
CV-TMLE*	0.95 (1.00)	0.93 (1.00)	0.94 (1.00)	0.91 (1.00)	0.91 (1.00)
MV-TMLE*	0.95 (1.05)	0.94 (0.92)	0.96 (0.95)	0.94 (0.77)	0.94 (0.80)
C-TMLE*	0.95 (0.68)	0.94 (0.70)	0.94 (0.69)	0.92 (0.51)	0.92 (0.46)
CV-AIPW*	0.90 (0.97)	0.77 (0.87)	0.71 (0.76)	0.65 (0.65)	0.45 (0.51)
CV-TMLE	0.85 (0.82)	0.85 (0.79)	0.76 (0.69)	0.67 (0.57)	0.61 (0.51)
MV-TMLE	0.92 (0.83)	0.88 (0.71)	0.80 (0.62)	0.79 (0.49)	0.67 (0.40)
C-TMLE	0.95 (0.60)	0.88 (0.60)	0.84 (0.49)	0.82 (0.39)	0.70 (0.34)
CV-AIPW	0.96 (1.57)	0.96 (1.58)	0.92 (1.41)	0.86 (1.17)	0.82 (0.97)
C-TMLE (Robust CI)	0.95 (0.62)	0.97 (0.74)	0.93 (0.68)	0.90 (0.55)	0.87 (0.44)

Table 4.2 shows the average coverage and length of confidence intervals. The positivity would also influence the estimation of the variance of the estimators. To better understand the behaviors of the two estimators, we studied two settings. In the first setting, we used the true SE, $\mathrm{SE}(\Psi_n)$, of the CV-TMLE, MV-TMLE, and C-TMLE (computed by a Monte Carlo simulation), and applied it to construct the CIs: $[\Psi_n - 1.96 \cdot \mathrm{SE}(\Psi_n), \Psi_n + 1.96 \cdot \mathrm{SE}(\Psi_n)]$. In the second case, we applied the estimated SE, $\hat{\mathrm{SE}}(\Psi_n)$, to construct CIs $[\Psi_n - 1.96 \cdot \hat{\mathrm{SE}}(\Psi_n), \Psi_n + 1.96 \cdot \hat{\mathrm{SE}}(\Psi_n)]$ for all the estimators.

First, we observe the TMLE* had much larger variance but smaller bias compared to C-TMLE in this experiment ($C = 2, N = 1000$). The large variance of TMLE helps the coverage for its CI, if we know the true variance (which is not possible). C-TMLE selects the cutpoint by optimizing the bias-variance trade-off to the MSE of the targeted parameter, and thus introduces more bias to reduce the variance in order to achieve better MSE. This is also shown in figure 4.10, where in sample size 1000, CV would on average truncate with a larger quantile. The overly large variance causes a much wider CI, which leads to the satisfactory coverage for TMLE*, though this makes the TMLE estimator less efficient.

However, as the true variance of the estimator is unknown in practice, CIs usually rely on the estimation of the variance. We observe that the variance of CV-TMLE, MV-TMLE, and C-TMLE estimator was underestimated in our experiments. It is also interesting to observe that A-IPW had high coverage, which is due to the over-estimating of its variance: according to table 4.2, the ratio for estimated SE and true SE when $n = 1000$ is 0.97/0.51 = 1.90. For all the estimators, the estimated variances were smaller than the true variances. Extreme weights in the clever covariates $H(A, W)$ cause large variance of the influence curve, thus makes it challenging to estimate the variance of the estimator. The variance estimator for the Positivity-C-TMLE estimator is less biased than the variance estimators for the CV-TMLE and MV-TMLE estimator. The ratio of the mean estimated SE to the corresponding true SE is about $0.88, 0.86, 0.71, 0.76, 0.74$ for $C = 0, 0.5, 1, 1.5, 2$, respectively. While for CV-TMLE and MV-TMLE, the ratio is much smaller. The ratio of the mean estimated SE to the corresponding true SE for the CV-TMLE estimator is about $0.82, 0.79, 0.69, 0.57, 0.51$, and for the MV-TMLE estimator is about $0.79, 0.77, 0.65, 0.64, 0.49$, for $C = 0, 0.5, 1, 1.5, 2$ respectively. This explains why CV-TMLE and MV-TMLE had worse CI coverage than C-TMLE.

We further applied the robust variance estimator for the positivity C-TMLE. The last row in Table 4.2 shows the coverage and relative width of CIs across 200 experiments. The results show the robust variance estimator provided better estimation of the variance, and improved the performance of confidence intervals significantly.

4.5 Conclusion

In this study, we proposed the Positivity-C-TMLE algorithm for adaptive truncation of the PS to address the issues from practical violations of the positivity assumption. We also designed simulations to evaluate and to help understand this novel estimator. We have the following conclusions:

- It is reasonable to believe that the optimal cutpoint varies significantly for different estimators. The Positivity-C-TMLE algorithm was designed for selecting the optimal cutpoint for TMLE, which might be the key point for its outstanding performance in the simulation.

- As discussed in subsection 4.4, the negative log-likelihood function $L^{(2)}$ for $\bar{G}$ is not a good objective function for selecting γ. Positivity-C-TMLE selects γ directly based on the targeted parameter, which is another important factor in its success in the simulation.

- The cutpoint selected by Positivity-C-TMLE is more sensitive to the positivity parameter C than the cutpoint selected by CV. The cutpoint selected by CV is sensitive to the sample size N, but not for the positivity parameter C. The cutpoint selected by MV-TMLE is not sensitive either to N, or to C.

- MV-TMLE has similar performance to C-TMLE when the sample size is large, or when practical violations are mild. However, in small samples with strong positivity violations (e.g. $N = 200, C = 2$), C-TMLE has much better performance than MV-TMLE.

- For Positivity-C-TMLE, the variance is under-estimated in the simulation, especially when practical violation of the positivity assumption is strong. Though the variance estimator for Positivity-C-TMLE is less biased than the one for CV-TMLE or MV-TMLE, a more conservative variance estimator is necessary to build a more reliable confidence interval for finite-sample study. We applied the robust variance estimator [67, 72] and observed a significant improvement.

There are several potential future extensions of this study. First, we only studied the case where the propensity score is estimated by a correctly specified parametric model. In other words, the failure of the estimators in the simulations are only from the practical violations of the positivity assumption, rather than model misspecification. It is important to investigate the behavior of each adaptive truncation method when the estimator for $\bar{G}_0$ is misspecified. The C-TMLE algorithm, when combined with non-parametric estimation of $\bar{G}$, provides a potential solution to the problem of overfitting the propensity score model while still allowing for flexible estimation [34]. In addition, this C-TMLE procedure could be extended to other data structure, like longitudinal data. We leave this for future work.

Appendix

For simplicity, we first introduce algorithm 6 for construction of a sequence of the C-TMLE candidates. In short, algorithm 6 can be considered as a black-box function, which outputs a sequence of TMLE candidates and a set of fluctuation points:

Algorithm 6 Positivity-C-TMLE Candidate Construction Algorithm

```
function GENERATE-CANDIDATES($\bar{Q}_n^0$, $P_n$, $\Gamma_k$, $\gamma_{\min} = 0.6, \gamma_{\max} = 1$, $\eta = 0.01$)
    Train an estimator $\bar{G}_n$ of $\bar{G}_0$ on $P_n$.
    Construct a sequence of propensity score model $\bar{G}_{n,\gamma}$ indexed by the corresponding cutpoint $\gamma$, where $\gamma$ is the $\gamma$-th empirical quantile of the estimated PS.
    Initialize $k = 1$ (the loop index).
    Initialize $\gamma_0 = \gamma_{\min}$ (the left bound of the remaining set of quantiles).
    Initialize $\Gamma$, the set of quantiles under consideration, sampled from $\gamma_{\min}$ to $\gamma_{\min}$, with fixed step size $\eta$
    if $\Gamma_k$ (the set of fluctuation points) is not provided then
        Initialize SearchFluctPoint = True.
        Initialize $\Gamma_k = []$
    else
        Initialize SearchFluctPoint = False
    end if
    while $\Gamma$ is not empty do
        if SearchFluctPoint then
            Apply targeting step for the same initial estimate $\bar{Q}_n^k$ with each $\bar{G}_{n,\gamma}$, $\gamma \in \Gamma$.
            Select $\gamma_k$ corresponding to the $\bar{Q}^*_{n,\gamma_k}$ that achieves the smallest empirical risk $P_n L^{(1)}(\bar{Q}^*_{n,\gamma_k}(A, W))$.
            Append $\gamma_k$ to $\Gamma_k$ (record the current fluctuation point).
        else
            $\gamma_k = \Gamma_k[k]$ (make fluctuation based on the provided set $\Gamma_k[k]$).
        end if
        For $\gamma \in [\gamma_{k-1}, \gamma_k]$, compute the corresponding TMLE using initial estimate $\bar{Q}_n^{k-1}$ and propensity score estimate $\bar{G}_{n,\gamma}$.
        We denote such estimate with $\bar{Q}^*_{n,\gamma}$ and record them ($\gamma \in [\gamma_{k-1}, \gamma_k]$ is no longer under consideration).
        Set a new initial estimate $\bar{Q}_n^k = \bar{Q}^*_{n,\gamma_k}$.
        Update the remaining set of quantiles $\Gamma = (\gamma_k, \gamma_{max}]$ (only consider the quantiles between $\gamma_k$ and $\gamma_{\max}$).
        Set $k = k + 1$.
    end while
      return $[\bar{Q}^*_{n,\gamma}, \gamma \in \Gamma]$, $\Gamma_k$
end function
```

Remark:

- If the fluctuation points set Γ_k is not provided (when trained on the whole observed data), this function generates a sequence of TMLE candidates by fluctuating the estimate when the empirical risk is not get improved.
- Otherwise, if the fluctuation points set Γ_k is provided (during cross-validation stage), it generates a sequence of TMLE candidates with a given set of fluctuation points.

The empirical loss decreases at each fluctuation point. During the cross-validation step, the set of fluctuation points is given (precomputed by whole training sample), so it would only update $\bar{Q}_n$ at each given fluctuation point. Then C-TMLE uses a targeted CV to select the stopping point.

Algorithm 7 Positivity-C-TMLE Algorithm

```
1: function Positivity-C-TMLE($\bar{Q}^0_n$, $P_n$, $\gamma_{\min} = 0.6, \gamma_{\max} = 1, V = 5$)
2:     Build a sequence of candidates using the whole dataset:
         $[\bar{Q}^*_{n,\gamma}], \Gamma_k$ = GENERATE-CANDIDATE($\bar{Q}^0_n, P_n$)
3:     Given the set of the fluctuation points $\Gamma_k$ from the previous step, compute the V-fold
   CV risk for each candidate:
         Build sequence of candidates $\bar{Q}^*_\gamma(P^0_{n,B_n})$ on the empirical distribution of the training
   set, $P^0_{n,B_n}$, with given $\Gamma_k$ and $B_n$ (a CV scheme), by calling
         $[\bar{Q}^*_\gamma(P^0_{n,B_n})]$, = GENERATE-CANDIDATE($\bar{Q}^0_n, P^0_{n,B_n}, \Gamma_k$).
         Repeat this for all the $V$ folds, and compute the average validation loss:
             $$E_{B_n} P^1_{n,B_n} L^{(1)}(\bar{Q}^*_\gamma(P^0_{n,B_n})).$$
4:     Select the best candidate $\bar{Q}^*_{n,\gamma_{ctmle}}$ among $\bar{Q}^*_{n,\gamma}$, with the smallest cross-validated loss
   in step (3), and its corresponding initial estimate $\bar{Q}_{n,\gamma_{ctmle}}$.
5:     Apply one additional targeting step to $\bar{Q}_{n,\gamma_{\text{c-tmle}}}$, with each $g_{n,\gamma}$, $\gamma \in [\gamma_{\min}, \gamma_{\text{c-tmle}})$,
   yielding a new sequence of estimate $\bar{Q}^*_{n,\gamma}$.
6:     Select $\bar{Q}^*_n = \arg\min_{\bar{Q}^*_{n,\gamma}} P_n L^{(1)}(\bar{Q}^*_{n,\gamma}), \gamma \in [\gamma_{\min}, \gamma_{\text{c-tmle}})$ with the smallest empirical
   loss as the final estimate.
         return $\bar{Q}^*_n$
7: end function
```

Remark:

- In step 2, Positivity-C-TMLE algorithm first computes the set of the fluctuation points and a sequence of candidates using the entire observed dataset.
- In step 3 and 4, it uses the V-fold CV to compute the CV-loss for each candidate. It picks the one with the smallest CV-risk, with corresponding initial estimate $\bar{Q}_{n,\gamma_{\text{c-tmle}}}$.

- Finally it fluctuates $\bar{Q}_{n,\gamma_{\text{c-tmle}}}$ with each $\gamma > \gamma_{\text{c-tmle}}$, and selects the $\bar{Q}_n^*$ with the smallest empirical loss, according to [34].

The final estimate for the causal parameter, ATE, is given by:

$$\psi_n^{\text{c-tmle}} = \frac{1}{n}\sum_{i=1}^{n}(\bar{Q}_n^*(1, W_i) - \bar{Q}_n^*(0, W_i))$$

Bibliography

[1] Oliver Bembom et al. *Data-adaptive Selection Of The Adjustment Set In Variable Importance Estimation*. Tech. rep. 2008.

[2] David Benkeser et al. "Online cross-validation-based ensemble learning". In: *Statistics in medicine* 37.2 (2018), pp. 249–260.

[3] Peter J Bickel et al. *Efficient and adaptive estimation for semiparametric models.* Springer-Verlag, 1998.

[4] M Alan Brookhart et al. "Variable selection for propensity score models". In: *American Journal of Epidemiology* 163.12 (2006), pp. 1149–1156.

[5] Tianqi Chen and Tong He. "Xgboost: extreme gradient boosting". In: *R package version 0.4-2* (2015).

[6] Stephen R Cole and Miguel A Hernan. "Constructing inverse probability weights for marginal structural models". In: *American journal of epidemiology* 168.6 (2008), pp. 656–664.

[7] Richard Crump et al. *Moving the goalposts: Addressing limited overlap in the estimation of average treatment effects by changing the estimand.* 2006.

[8] Molly Margaret Davies and Mark J van der Laan. "Optimal Spatial Prediction Using Ensemble Machine Learning". In: *The international journal of biostatistics* 12.1 (2016), pp. 179–201.

[9] Rajeev H Dehejia and Sadek Wahba. "Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs". In: *Journal of the American statistical Association* 94.448 (1999), pp. 1053–1062.

[10] Jessica M Franklin et al. "Plasmode simulation for the evaluation of pharmacoepidemiologic methods in complex healthcare databases". In: *Computational Statistics & Data Analysis* 72 (2014), pp. 219–226.

[11] Jessica M Franklin et al. "Regularized Regression Versus the High-Dimensional Propensity Score for Confounding Adjustment in Secondary Database Analyses". In: *American journal of epidemiology* 187.7 (2015), pp. 651–659.

[12] David A Freedman and Richard A Berk. "Weighting regressions by propensity scores". In: *Evaluation Review* 32.4 (2008), pp. 392–409.

[13] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. “glmnet: Lasso and elastic-net regularized generalized linear models”. In: *R package version* 1.4 (2009).

[14] Gary L Gadbury et al. “Evaluating statistical methods using plasmode data sets in the age of massive public databases: an illustration using false discovery rates”. In: *PLoS Genet* 4.6 (2008), e1000098.

[15] Susan Gruber and Mark J van der Laan. “A Targeted Maximum Likelihood Estimator of a Causal Effect on a Bounded Continuous Outcome”. In: *The International Journal of Biostatistics* 6.1 (2010), Article 26.

[16] Susan Gruber and Mark J van der Laan. “An application of collaborative targeted maximum likelihood estimation in causal inference and genomics”. In: *The International Journal of Biostatistics* 6.1 (2010), Article 18.

[17] Susan Gruber and Mark J van der Laan. “C-TMLE of an additive point treatment effect”. In: *Targeted Learning*. Springer, 2011, pp. 301–321.

[18] Susan Gruber et al. “Ensemble learning of inverse probability weights for marginal structural modeling in large observational datasets”. In: *Statistics in medicine* 34.1 (2015), pp. 106–117.

[19] Joseph F Hair et al. *Multivariate Data Analysis*. Vol. 6. Pearson Prentice Hall Upper Saddle River, NJ, 2006.

[20] J Hajek. “Comment on a paper by D. Basu”. In: *Foundations of statistical inference* 236 (1971).

[21] James J Heckman, Hidehiko Ichimura, and Petra E Todd. “Matching as an econometric evaluation estimator: Evidence from evaluating a job training programme”. In: *The review of economic studies* 64.4 (1997), pp. 605–654.

[22] M. A. Hernan, B. Brumback, and J. M. Robins. “Marginal Structural Models to Estimate the Causal Effect of Zidovudine on the Survival of HIV-Positive Men”. In: *Epidemiology* 11.5 (2000), pp. 561–570.

[23] Daniel G Horvitz and Donovan J Thompson. “A generalization of sampling without replacement from a finite universe”. In: *Journal of the American statistical Association* 47.260 (1952), pp. 663–685.

[24] Guido W Imbens. “The role of the propensity score in estimating dose-response functions”. In: *Biometrika* 87.3 (2000), pp. 706–710.

[25] Cheng Ju, Aurelien Bibaut, and Mark J van der Laan. “The Relative Performance of Ensemble Methods with Deep Convolutional Neural Networks for Image Classification”. In: *arXiv preprint arXiv:1704.01664* (2017).

[26] Cheng Ju, Antoine Chambaz, and Mark J van der Laan. “Collaborative targeted inference from continuously indexed nuisance parameter estimators”. In: *arXiv preprint arXiv:1804.00102* (2018).

[27] Cheng Ju, Susan Gruber, and Mark van der Laan. *ctmle: Collaborative Targeted Maximum Likelihood Estimation.* R package version 0.1.1. 2017. URL: `https://CRAN.R-project.org/package=ctmle`.

[28] Cheng Ju, Joshua Schwab, and Mark J van der Laan. "On Adaptive Propensity Score Truncation in Causal Inference". In: *arXiv preprint arXiv:1707.05861* (2017).

[29] Cheng Ju et al. "Collaborative-controlled LASSO for Constructing Propensity Score-based Estimators in High-Dimensional Data". In: *arXiv preprint arXiv:1706.10029* (2017).

[30] Cheng Ju et al. "Propensity score prediction for electronic healthcare databases using Super Learner and High-dimensional Propensity Score Methods". In: *arXiv preprint arXiv:1703.02236* (2017).

[31] Cheng Ju et al. "Scalable Collaborative Targeted Learning for High-Dimensional Data". In: *arXiv preprint arXiv:1703.02237* (2017).

[32] Joseph Kang et al. "Practice of causal inference with the propensity of being zero or one: assessing the effect of arbitrary cutoffs of propensity scores". In: *Communications for Statistical Applications and Methods* 23.1 (2016), pp. 1–20.

[33] Hiraku Kumamaru et al. "Comparison of high-dimensional confounder summary scores in comparative studies of newly marketed medications". In: *Journal of clinical epidemiology* 76 (2016), pp. 200–208.

[34] Mark J van der Laan, Antoine Chambaz, and Cheng Ju. "C-TMLE for continuous tuning". In: *Targeted Learning in Data Science.* Springer, 2018, pp. 143–161.

[35] Mark J van der Laan and Sandrine Dudoit. "Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples". In: *U.C. Berkeley Division of Biostatistics Working Paper Series* (2003), Working Paper 130.

[36] Mark J van der Laan, Eric C Polley, and Alan E Hubbard. "Super Learner". In: *Statistical Applications in Genetics and Molecular Biology* 6.1 (2007), Article 25.

[37] Mark J van der Laan and James M Robins. *Unified methods for censored longitudinal data and causality.* Springer Science & Business Media, 2003.

[38] Mark J van der Laan and Sherri Rose. *Targeted Learning: Causal Inference for Observational and Experimental Data.* Springer Science & Business Media, 2011.

[39] Mark J van der Laan and Daniel Rubin. "Targeted maximum likelihood learning". In: *The International Journal of Biostatistics* 2.1 (2006).

[40] Mark J van der Laan, Susan Gruber, et al. "Collaborative double robust targeted maximum likelihood estimation". In: *The International Journal of Biostatistics* 6.1 (2010), Article 17.

[41] Robert J LaLonde. "Evaluating the econometric evaluations of training programs with experimental data". In: *The American economic review* (1986), pp. 604–620.

[42] Brian K Lee, Justin Lessler, and Elizabeth A Stuart. "Improving propensity score weighting using machine learning". In: *Statistics in medicine* 29.3 (2010), pp. 337–346.

[43] Brian K Lee, Justin Lessler, and Elizabeth A Stuart. "Weight trimming and propensity score weighting". In: *PloS one* 6.3 (2011), e18174.

[44] Samuel D Lendle, Bruce Fireman, and Mark J van der Laan. "Targeted maximum likelihood estimation in safety analysis". In: *Journal of clinical epidemiology* 66.8 (2013), S91–S98.

[45] Romain Neugebauer and Mark van der Laan. "Why prefer double robust estimators in causal inference?" In: *Journal of Statistical Planning and Inference* 129.1 (2005), pp. 405–426.

[46] Elisabetta Patorno et al. "Studies with many covariates and few outcomes: selecting covariates and implementing propensity-score–based confounding adjustments". In: *Epidemiology* 25.2 (2014), pp. 268–278.

[47] Maya L Petersen et al. "Diagnosing and responding to violations in the positivity assumption". In: *Statistical methods in medical research* 21.1 (2012), pp. 31–54.

[48] Romain Pirracchio, Maya L Petersen, and Mark van der Laan. "Improving Propensity Score Estimators' Robustness to Model Misspecification Using Super Learner". In: *American Journal of Epidemiology* 181.2 (2015), pp. 108–119.

[49] Eric C Polley and Mark J van der Laan. *Super learner in prediction.* Tech. rep. 2010, U.C. Berkeley Division of Biostatistics Working Paper Series. Working Paper 266.

[50] Kristin E Porter et al. "The relative performance of targeted maximum likelihood estimators". In: *The International Journal of Biostatistics* 7.1 (2011), Article 31.

[51] Frank J Potter. "The effect of weight trimming on nonlinear survey estimates". In: *Proceedings of the American Statistical Association, Section on Survey Research Methods.* Vol. 758763. 1993.

[52] Jeremy A Rassen and Sebastian Schneeweiss. "Using high-dimensional propensity scores to automate confounding control in a distributed medical product safety surveillance system". In: *Pharmacoepidemiology and drug safety* 21.S1 (2012), pp. 41–49.

[53] Greg Ridgeway et al. "gbm: Generalized boosted regression models". In: *R package version* 1.3 (2006), p. 55.

[54] J. M. Robins and A. Rotnitzky. "Comment on the Bickel and Kwon article, 'Inference for Semiparametric Models: Some Questions and an Answer'". In: *Statistica Sinica* 11.4 (2001), pp. 920–936.

[55] J. M. Robins, A. Rotnitzky, and M.J. van der Laan. "Comment on "On Profile Likelihood" by S.A. Murphy and A.W. van der Vaart". In: *Journal of the American Statistical Association – Theory and Methods* 450 (2000), pp. 431–435.

[56] James M Robins. “A New Approach to Causal Inference in Mortality Studies with Sustained Exposure Periods - Application to Control of the Healthy Worker Survivor Effect”. In: *Mathematical Modelling* 7 (1986), pp. 1393–1512.

[57] James M Robins. “Marginal structural models versus structural nested models as tools for causal inference”. In: *Statistical models in epidemiology, the environment, and clinical trials*. Springer, 2000, pp. 95–133.

[58] James M Robins, Miguel Angel Hernan, and Babette Brumback. “Marginal Structural Models and Causal Inference in Epidemiology”. In: *Epidemiology* 11.5 (2000), pp. 550–560.

[59] James M Robins and Andrea Rotnitzky. “Semiparametric efficiency in multivariate regression models with missing data”. In: *Journal of the American Statistical Association* 90.429 (1995), pp. 122–129.

[60] James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. “Estimation of regression coefficients when some regressors are not always observed”. In: *Journal of the American statistical Association* 89.427 (1994), pp. 846–866.

[61] J.M. Robins. “Robust Estimation in Sequentially Ignorable Missing Data and Causal Inference Models”. In: *Proceedings of the American Statistical Association: Section on Bayesian Statistical Science*. 2000, pp. 6–10.

[62] Sherri Rose. “A machine learning framework for plan payment risk adjustment”. In: *Health services research* (2016).

[63] Sherri Rose and Mark J van der Laan. “Understanding TMLE”. In: *Targeted Learning*. Springer, 2011, pp. 83–100.

[64] Paul R Rosenbaum and Donald B Rubin. “The central role of the propensity score in observational studies for causal effects”. In: *Biometrika* (1983), pp. 41–55.

[65] Daniel O Scharfstein, Andrea Rotnitzky, and James M Robins. “Adjusting for nonignorable drop-out using semiparametric nonresponse models”. In: *Journal of the American Statistical Association* 94.448 (1999), pp. 1096–1120.

[66] Sebastian Schneeweiss et al. “High-dimensional propensity score adjustment in studies of treatment effects using health care claims data”. In: *Epidemiology* 20.4 (2009), p. 512.

[67] J Schwab et al. “ltmle: Longitudinal targeted maximum likelihood estimation”. In: *R package version 0.9* 3 (2014).

[68] Ori M Stitelman and Mark J van der Laan. “Collaborative targeted maximum likelihood for time to event data”. In: *The International Journal of Biostatistics* 6.1 (2010), Article 21.

[69] Ori M Stitelman et al. “Targeted maximum likelihood estimation of effect modification parameters in survival analysis”. In: *The International Journal of Biostatistics* 7.1 (2011), Article 19.

[70] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), pp. 267–288.

[71] Sengwee Toh, Luis A Garcia Rodriguez, and Miguel A Hernan. "Confounding adjustment via a semi-automated high-dimensional propensity score algorithm: an application to electronic medical records". In: *Pharmacoepidemiology and drug safety* 20.8 (2011), pp. 849–857.

[72] Linh Mai Tran. "Robust variance estimation and inference for causal effect estimation". PhD thesis. University of California, Berkeley, 2016.

[73] Aad W van der Vaart, Sandrine Dudoit, and Mark J Laan. "Oracle inequalities for multi-fold cross validation". In: *Statistics & Decisions* 24.3 (2006), pp. 351–371.

[74] Hui Wang, Sherri Rose, and Mark J van der Laan. "Finding quantitative trait loci genes with collaborative targeted maximum likelihood learning". In: *Statistics & Probability Letters* 81.7 (2011), pp. 792–796.

[75] Daniel Westreich and Stephen R Cole. "Invited commentary: positivity in practice". In: *American journal of epidemiology* 171.6 (2010), pp. 674–677.

[76] Daniel Westreich, Justin Lessler, and Michele Jonsson Funk. "Propensity score estimation: machine learning and classification methods as alternatives to logistic regression". In: *Journal of clinical epidemiology* 63.8 (2010), p. 826.

[77] Richard Wyss et al. "Using Super Learner Prediction Modeling to Improve High-dimensional Propensity Score Estimation". In: *Epidemiology* 29.1 (2018), pp. 96–106.

[78] Yongling Xiao, Erica EM Moodie, and Michal Abrahamowicz. "Comparison of approaches to weight truncation for marginal structural Cox models". In: *Epidemiologic Methods* 2.1 (2013), pp. 1–20.