Title
Targeted Learning of High-dimensional Parameters and Its Finite Sample Inference

Author
Cai, Weixin

Publication Date
2019

Peer reviewed|Thesis/dissertation

Targeted learning of high-dimensional parameters and its finite sample inference

by

Weixin Cai

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Biostatistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Mark J. van der Laan, Chair
Professor Peng Ding
Professor Alan Hubbard
Professor Maya L. Petersen

Summer 2019

Targeted learning of high-dimensional parameters and its finite sample inference

Abstract

Targeted learning of high-dimensional parameters and its finite sample inference

by

Weixin Cai

Doctor of Philosophy in Biostatistics

University of California, Berkeley

Professor Mark J. van der Laan, Chair

Targeted maximum likelihood estimator (and semiparametric efficient estimators in general) involves deriving the efficient influence function of target parameters and adjusting an estimate of the data distribution towards the target estimand. This adjustment step requires fitting a least favorable submodel on the initial estimator with the same dimensionality of the parameter, which can become unstable for high-dimensional target parameters. Another direction that will vastly improve the credibility of these semiparametric estimators is to improve the finite-sample coverage of confidence intervals. In this dissertation, we first study the robust estimation of high-dimensional target parameters. Then we investigate how to perform finite sample inference in a large semi-parametric model. We also build an estimator that is simultaneously efficient for a large family of target parameters by undersmoothing a single regression.

In Chapter 1, we propose using universal least favorable submodel to robustly estimate high-dimensional target parameters, with applications to survival analysis. We establish a novel connection between a universal least favorable submodel and moving along a sparse local least favorable submodel, and demonstrate the extensions in survival analysis when the whole survival curve needs to be nonparametrically estimated and given statistical inference. We assess the finite sample performance in both a simulation study and an observational study on monoclonal gammopathy.

In Chapter 2, we theoretically develop and extend nonparametric bootstrap inference for the targeted maximum likelihood estimator (TMLE). We establish a formal theorem showing that the nonparametric bootstrap is an asymptotically valid procedure for finite sample TMLE inference using highly-adaptive LASSO (HAL) as the nuisance parameter estimator and demonstrate superior coverage than existing influence-function-based methods. This article explores the problem of applying semiparametric models and machine learning algorithms to small datasets and still have honest causal and statistical inference. Prior to this work, one either has to run nonparametric bootstrap by assuming small parametric models or do estimation in a large semiparametric model where the nonparametric bootstrap has no theoretical guarantee. We propose an effective tuning parameter selection method that

optimizes confidence interval coverage (rather than estimation precision) which shows good coverage even for non-doubly robust causal parameters.

In Chapter 3, we propose two efficient estimators based on highly-adaptive LASSO (HAL): targeted HAL and undersmoothed HAL. Using undersmoothed HAL to estimate the likelihood gives us an efficient estimator for a large family of target parameters. The key is to propose a strategy to choose the tuning parameter that results in a sectional variation norm larger than the one selected using cross-validation. In this chapter, we propose a 'multi-task tuning' method that can be generally applied to a wide range of target parameters. The second method called targeted HAL solves the efficient score equations by including an additional covariate into the LASSO design matrix that targets the statistical parameter of interest. We provide examples of our methods for estimating the average treatment effect and illustrate using two simulations where one favors inverse probability weighting methods (such as estimating equations and TMLE) and another challenging design where there is practical violation of the positivity assumption. We demonstrate the outstanding performance of the undersmoothed HAL in both scenarios. We also show theoretical results that shed light on why undersmoothed HAL is performing well in data generating distributions where positivity assumption is violated.

To my family and friends.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I am grateful to have Professor Mark van der Laan as my advisor. He inspired me into the field of theoretical machine learning and causal inference. Because of him, the past four years have been the most rewarding and thrilling periods of my life. Mark is incredibly generous with his ideas and time, and have influenced me greatly with his rigor, dedication, and passion. I am fortunate to work with him closely and study one of the most critical frontiers of statistics. I am constantly amazed by his ability to distill the essence of a problem so rapidly; his broad knowledge and deep understanding of so many subjects statistics, causal inference, optimization; and by his care for his students with quality time. Over these years, he guided me about how to approach research projects, give talks and write, and most importantly, he taught me what is good research.

I am also grateful for the chance to learn from amazing professors at Berkeley. I received tremendous support from Prof. Alan Hubbard on a few applied statistics projects. I want to thank him for his knowledge and kindness, and for making Berkeley my second home. I thank Prof. Peng Ding for his deep wisdom in causal inference and broadening my knowledge in experiments and designs. I am also thankful to Prof. Maya Petersen for providing me with constructive feedback during my qualifying exam and sharing her wisdom of transitioning from a Ph.D. student to a young researcher. In my earlier graduate years, I was very fortunate to collaborate with Prof. Lexin Li, who guided me into becoming an independent researcher and taught me how to be an eloquent presenter.

I deeply cherish the opportunities to work alongside a group of talented collaborators: David Benkeser, Eytan Bakshy, Maximilian Balandat. They sparked my passion for my lifelong research in statistics and machine learning. I want to thank my friends at Berkeley, with whom I shared happy memories. I thank Siqi Wu for being an incredibly supportive academic brother, whose encouragement and advice at each critical step in my grad career are invaluable to me. I thank all my friends in the biostatistics and statistics department, Yu Wang, Jonathan Levy, Lucia Petito, Yuting Ye, Caleb Miles, Jeremy Coyle, Nima Hejazi, Ivana Malenica, Cheng Ju, Steve Howard, for the academic and non-academic conversations.

Above all, I owe the most to my family, in particular to my parents, for your unconditional love and support.

Chapter 1

One-step Targeted Maximum Likelihood Estimation for Time-to-event Outcomes

1.1 Introduction

Researchers in observational survival analysis are interested in not only estimating survival curve nonparametrically but also having statistical inference for the survival curve as a whole. We consider right-censored failure time data where we observe n independent and identically distributed observations of a vector random variable consisting of baseline covariates, a binary treatment at baseline, a survival time subject to right censoring, and the censoring indicator. We assume the baseline covariates are allowed to affect the treatment and censoring so that an estimator that ignores covariate information would be inconsistent. The goal is to use these data to estimate the counterfactual average survival curve of the population if all subjects are assigned the same treatment at baseline.

Existing methods such as inverse probability of censoring weighted (IPCW) estimator, estimating equations (EE) and targeted maximum likelihood estimator (TMLE) do not produce a monotone estimator of the curve, which translates to large variance. The reason is that these estimators separately estimate the survival curve for each time point. The IPCW estimator [42] re-weights the observed data by the inverse of the product of the propensity score and censoring probability before applying a standard estimation method. The EE estimator [17] is a locally efficient and double robust estimator, which improves the IPCW by adding the sample mean of the efficient influence curve. EE is more efficient than IPCW when the conditional distribution of failure given treatment and baseline covariates is consistently estimated [17]. For IPCW, its consistency relies on correctly estimating the conditional survival function of censoring. In contrast, EE is doubly robust in the sense that if either the conditional failure distribution or both propensity score and conditional censoring probability is correctly estimated, then the EE estimator will be consistent [17].

TMLE is a plug-in doubly robust and locally efficient estimator and is shown to be better than the IPCW and EE methods [38, 44]. In contrast to these methods, TMLE performs an adjustment on the estimate of the data distribution prior to applying the parameter mapping thus always respecting the parameter space (probabilities falling inside [0,1]) [Chapter 6 of 26]. As a result, TMLE is a plug-in estimator that is more robust in finite samples than EE. While TMLE works well to improve the statistical efficiency of EE, it can still give rise to a non-monotone survival curve. The reason is that both EE and TMLE are built on efficiency theory for univariate parameters. As a result, their solutions for estimating the survival curve is a collection of univariate survival probability estimators.

In this article, we propose a TMLE that targets the survival curve as a whole, while still preserving the performance of the point-wise TMLE for the survival curve at a point. Due to the joint targeting, the resulting estimator is a monotone function. The method we propose is built upon the recent advancement of TMLE theory called one-step TMLE [21]. This powerful framework estimates the entire survival curve and ensures monotonicity. We also discover that the proposed new algorithm is more stable and computationally more efficient than classic TMLE. We also give a new insight into one-step TMLE by comparing it to the high-dimensional penalized regression literature, which will shed light on the superior finite sample performance of our method.

Organization of paper We start in Section 1.2 by defining the right-censored data, stating the parameter of interest, and reviewing the efficient influence curve of the parameter. In Section 1.3 we review nonparametric regressions used in observational survival analysis, and in Section 1.4 we formally review the IPCW, EE, and classic TMLE estimators. In Section 1.4 we present intuition on why EE and classic TMLE do not always produce a monotonically decreasing survival curve. We use this intuition to build a TMLE that ensures monotonicity in Section 1.5. In Section 1.6 we present a simulation study demonstrating the finite sample performance of the estimators, and in Section 1.7 we present an applied example.

1.2 Statistical formulation of estimation of the survival curve

Let the full data be $\mathbf{X}_i = (\mathbf{W}_i, A_i, C_{1i}, C_{0i}, T_{1i}, T_{0i}), i = 1, ..., n$, where $\mathbf{W}$ is a vector of baseline covariates, $A \in \{0, 1\}$ is binary treatment assigned at baseline, T_1 is the failure time under treatment, T_0 is the failure time under control, C_1 is the censoring time under treatment, C_0 is the censoring time under control. Our observed data is $\boldsymbol{O}_i = (\mathbf{W}_i, A_i, \Delta_i, \widetilde{T}_{A_i}) \sim^{i.i.d} P_0 \in \mathcal{M}$ for $i = 1, ..., n$, where $\widetilde{T} \triangleq \min(T_A, C_A)$ is the last measurement time of the subject, and $\Delta \triangleq I(T_A \leqslant C_A)$ is the censoring indicator. P_0 denotes the true probability distribution of $\boldsymbol{O}$, and we use p_0 to denote the true probability density. $\mathcal{M}$ is the model space of distributions which is believed to be nonparametric.

The causal parameter is the marginal survival curve in the whole population where every subject is under the same treatment

$$P(T_a > t), t = 1, ..., t_{max},$$

where T_a is the counterfactual failure time one would have observed had an individual's treatment been set, possibly contrary to fact, to treatment level a. The parameter can be causally identified from the observed data under the assumptions: (a) no unmeasured confounder, (b) coarsening at random (the joint variable of censoring and treatment is conditionally independent of the full data given the observed data), and (c) positivity assumption [17, 14, 43]. After causal identification, our task is reduced to estimating the statistical parameter

$$\mathbf{\Psi}_{A=a}(P)(t) = E[P(T > t|A = a, \mathbf{W})], t = 1, ..., t_{max}.$$

This $\mathbf{\Psi} : \mathcal{M} \to [0, 1]^{t_{max}}$ is a mapping from model space $\mathcal{M}$ to the parameter space of survival probabilities. $\mathbf{\Psi}(P)$ is whole survival curve and $\mathbf{\Psi}(P)(t)$ is the survival probability at t. For the rest of the paper, we demonstrate estimators focusing on example in this parameter family, the treatment-specific marginal survival curve $\mathbf{\Psi}_{A=1}$. Symmetric arguments can be made about $\mathbf{\Psi}_{A=0}$, and thus all transformations of the two parameters (such as difference of two counterfactual survival probabilities). The components needed to plug into $\mathbf{\Psi} \equiv \mathbf{\Psi}_{A=1}$ for the estimand are the conditional survival curve for failure event and the distribution of $\mathbf{W}$, which need to be learned from the observed data. For performing observational survival analysis, the conditional survival function for censoring and propensity score also need to be estimated. Under the causal identification assumptions, the probability density under P factorizes as follows:

$$\begin{aligned} p(\boldsymbol{O}) = q_W(\mathbf{W})g(\mathbf{W}) \prod_{t \leqslant \widetilde{T}} \lambda_N(t|A, \mathbf{W})^{dN(t)}[1 - \lambda_N(t|A, \mathbf{W})]^{1-dN(t)} \\ \prod_{t \leqslant \widetilde{T}} \lambda_{A_c}(t|A, \mathbf{W})^{dA_c(t)}[1 - \lambda_{A_c}(t|A, \mathbf{W})]^{1-dA_c(t)}, \end{aligned} \tag{1.1}$$

where q_W is the density of probability distribution of $\mathbf{W}$; $g(\mathbf{W}) = P(A|\mathbf{W})$ is the propensity score; $\lambda_N(t|A, \mathbf{W})$ and $\lambda_{A_c}(t|A, \mathbf{W})$ are the conditional hazards of the failure event and censoring event; $dN(t)$ and $dA_c(t)$ are the counting process indicators of the failure event and censoring event. We will formally define them in Section 1.3.

Efficient influence curve

The EE and TMLE methods to be discussed in this paper are built around the parameter's efficient influence curve offer a straightforward approach to estimation. [3] show that a regular estimator for a statistical parameter in a semiparametric model is asymptotically efficient (i.e., the estimator has minimal asymptotic variance), if it is asymptotically linear with influence curve (influence function) equal to the efficient influence curve (EIC). Under our model space $\mathcal{M}$, the EIC for $\mathbf{\Psi}$ was derived and presented in [38] as

$$
\begin{aligned}
D_t^*(P) = \sum_{k \leqslant t} h_t(g_{0,A}, S_{0,A_c}, S_{0,N})(k, A, \mathbf{W}) \Big[I(\widetilde{T} = k, \Delta = 1) - \\
I(\widetilde{T} \geqslant k) \lambda_{0,N}(k | A = 1, \mathbf{W}) \Big] + S_{0,N}(t | A = 1, \mathbf{W}) - \Psi_d(P)(t) \\
\equiv D_{1,t}^*(g_{0,A}, S_{0,A_c}, S_{0,N}) + D_{2,t}^*(P),
\end{aligned} \tag{1.2}
$$

where

$$
h_t(g_{0,A}, S_{0,A_c}, S_{0,N})(k, A, \mathbf{W}) = -\frac{I(A = 1) I(k \leqslant t)}{g_{0,A}(A = 1 | W) S_{0,A_c}(k_- | A, \mathbf{W})} \frac{S_{0,N}(t | A, \mathbf{W})}{S_{0,N}(k | A, \mathbf{W})}. \tag{1.3}
$$

1.3 Nonparametric estimation of components for observational survival analysis methods

After causal identification, existing observational survival analysis methods depend on estimating four components nonparametrically: (1) conditional survival function for failure event given treatment and confounders, (2) conditional survival function for censoring event given treatment and confounders, (3) propensity score of treatment given confounders, and (4) distribution of confounders in the population of interest.

Conditional survival function for failure event
The conditional survival function is estimated by first estimating the conditional hazard of the failure event, and then transforming into the conditional survival function. The definition of the conditional hazard is

$$
\begin{aligned}
\lambda_N(t | A, \mathbf{W}) &= P(\widetilde{T} = t, \Delta = 1 | \widetilde{T} \geqslant t, A, \mathbf{W}) && (1.4) \\
&= P[dN(t) = 1 | N(t-1) = 0, A_c(t-1) = 0, A, \mathbf{W}], && (1.5)
\end{aligned}
$$

where $N(t) = I(\widetilde{T} \leqslant t, \Delta = 1)$, $A_c(t) = I(\widetilde{T} \leqslant t, \Delta = 0)$ and

$$
dN(t) = \begin{cases} 1, \text{if } N(t) = 1 \text{ and } N(t-1) = 0 \\ 0, \text{otherwise,} \end{cases} \tag{1.6}
$$

$$
dA_c(t) = \begin{cases} 1, \text{if } A_c(t) = 1 \text{ and } A_c(t-1) = 0 \\ 0, \text{otherwise.} \end{cases} \tag{1.7}
$$

The definition (1.5) gives guidance of how to construct a classification task and estimate the conditional hazard. We first construct a training data where each subject $\boldsymbol{O}_i$ is mapped

into t_{max} rows in a new data with covariates $(dN(t)_i, N(t-1)_i, A_c(t-1)_i, A_i, \mathbf{W}_i, t), t = 1, ..., t_{max}$. Estimating the conditional hazard now becomes classification of $dN(t)_i$, using $(N(t-1)_i, A_c(t-1)_i, A_i, \mathbf{W}_i, t)$ as features, performed on the subset of rows that satisfy the criteria $N(t-1)_i = 0$ and $A_c(t-1)_i = 0$. Note that we include an extra feature t into the design matrix and pool data from all $t = 1, ..., t_{max}$ into one classification model. Empirically we found that smoothing over t accelerates the training of classification algorithms. We follow the common standard to transform the conditional hazard into the conditional survival function:

$$S_N(t|A, \mathbf{W}) = P(T > t|A, \mathbf{W}) = \prod_{k=1}^{t} [1 - \lambda_N(k|A, \mathbf{W})].$$

Conditional survival function for censoring event

The conditional survival function for censoring is estimated in the same fashion as that for the failure event, while swapping the role of N and A_c when constructing the classification dataset.

$$\begin{aligned}
\lambda_{A_c}(t|A, \mathbf{W}) &= P(\widetilde{T} = t, \Delta = 0|\widetilde{T} \geqslant t, A, \mathbf{W}) \\
&= P[dA_c(t) = 1|N(t-1) = 0, A_c(t-1) = 0, A, \mathbf{W}], \\
S_{A_c}(t|A, \mathbf{W}) &= P(C > t|A, \mathbf{W}) = \prod_{k=1}^{t} [1 - \lambda_{A_c}(k|A, \mathbf{W})].
\end{aligned}$$

Propensity score

We estimate the propensity score by running a classification of A against $\mathbf{W}$ as features.

$$g(\mathbf{W}) = P(A = 1|\mathbf{W}).$$

Distribution of confounders

We model the joint distribution of confounders using the empirical probability distribution of $\mathbf{W}_1, ..., \mathbf{W}_n$, which we denote as $Q_{n,W}$.

1.4 Review of existing observational survival analysis methods

Inverse probability of censoring weighted estimator

The inverse probability of censoring weighted (IPCW) estimator re-weights the observed data by the inverse of the product of the propensity score and censoring probability in order to make the treatment arms among the uncensored subjects comparable with respect

to confounders, and then applies standard estimation as if treatment was randomized and censoring was non-informative. The IPCW estimator for $\boldsymbol{\psi}_0(t)$ is

$$\boldsymbol{\psi}_{n,IPCW}(t) = \frac{1}{n}\sum_{i=1}^{n} \frac{I(\widetilde{T}_i > t, \Delta_i = 1, A_i = 1)}{S_{A_c}(\widetilde{T}_i|\boldsymbol{W}_i, A = 1)g(\boldsymbol{W}_i)}. \tag{1.8}$$

Estimating equations method

The estimating equation (EE) method is an asymptotically linear estimator based on solving the efficient influence curve equation:

$$\frac{1}{n}\sum_{i=1}^{n} D_t^*(P_n)(\boldsymbol{O}_i) = 0. \tag{1.9}$$

We remind readers that a regular estimator $\boldsymbol{\psi}_n$ of $\boldsymbol{\psi}_0$ is asymptotically linear if and only if $\sqrt{n}(\boldsymbol{\psi}_n - \boldsymbol{\psi}_0)$ behave approximately as an empirical mean of a mean-zero, finite-variance function of the observed O, where $\boldsymbol{\psi}_0 = \boldsymbol{\Psi}(P_0), \boldsymbol{\psi}_n = \boldsymbol{\Psi}(P_n)$ are the estimand and the estimate. This function is referred to as the estimator's influence curve (1.2). The EE method is one way for constructing estimators with user-specified influence curve, which applies an EIC-based correction to the plug-in estimate. Once the empirical influence curve is evaluated for each observation, the EE method is the IPCW estimator added to the sample mean of EIC evaluated on each observation.

$$\boldsymbol{\psi}_{n,EE}(t) = \boldsymbol{\psi}_{n,IPCW}(t) + \frac{1}{n}\sum_{i=1}^{n} D_{t,n}^*(\boldsymbol{O}_i), \tag{1.10}$$

where $D_{t,n}^*(\boldsymbol{O}_i) = D_t^*(P_n)(\boldsymbol{O}_i) = D_t^*(g_n, Q_n)(\boldsymbol{O}_i)$ is calculated by plugging in the initial estimators of $Q_n = (Q_{n,W}, S_{n,N})$ and $g_n = (g_{n,A}, S_{n,C})$ into D_t^* and evaluate at $\boldsymbol{O}_i$.

Targeted maximum likelihood estimator

TMLE is a general framework for constructing plug-in estimators that satisfy user-specified equations, which in our case is the EIC equation (1.9). It is a plug-in estimator in the sense that the estimators for $S_N(t|A = 1, \boldsymbol{W})$ can be plugged into the mapping $\boldsymbol{\Psi}$ to calculate an estimate as

$$\boldsymbol{\Psi}(Q_n)(t) = \frac{1}{n}\sum_{i=1}^{n} S_{n,N}(t|A = 1, \boldsymbol{W}_i).$$

Since TMLE updates parts of the likelihood before applying the parameter mapping, it is guaranteed to fall inside the range $[0, 1]$ of the survival probability.

For the TMLE of $\boldsymbol{\Psi}(t)$, the method is implemented in two steps. First, initial estimators of the four components are generated by user in Section 1.3. Subsequently, the initial estimators are carefully modified such that (i) the modified estimators inherit desirable properties

of the initial estimators (e.g., their rate of convergence); and (ii) relevant, user-specified equations are satisfied. For the present problem, the conditional survival function of failure event is iteratively updated to form a targeted estimator $\boldsymbol{\Psi}_n^* = \boldsymbol{\Psi}(P_n^*) = \boldsymbol{\Psi}(g_n, S_{n,A_c}, S_{n,N}^*)$, such that the EIC estimating equation $\frac{1}{n}\sum_{i=1}^n D_t^*(P_n^*)(\boldsymbol{O}_i) = 0$ is satisfied. This can be achieved, for example, by defining a logistic regression working model for the failure event conditional hazard, with $\text{logit}(\lambda_{(k)}) = \text{logit}[\lambda_{n,N}(k|A=1,\boldsymbol{W})]$ as an offset, no intercept term, and a single covariate $h_{(k)}$, regressed onto the binary outcome $N_{(k)} = I(\widetilde{T} = k, \Delta = 1)$. For each $(k, W), k = 1, ..., t_{max}$, we define this covariate as $h_{(k)} = h_t(g_{n,A}, S_{n,A_c}, S_{n,N})(k, 1, \boldsymbol{W})$. The maximum likelihood estimator ε_n of the regression coefficient ε associated with the covariate $h_{(k)}$ is estimated (via iterative re-weighted least squares). For each W, we define the so-called targeted $S_{n,N}^*$ as the conditional survival function transformed from the targeted conditional hazard $\lambda_{n,N}^*(k|A=1,\boldsymbol{W}) = \text{expit}[\text{logit}(\lambda_{(k)}) + \varepsilon_n h_{(k)}]$. For notation simplicity, we use P_n and P_n^* for the initial and targeted distribution of P_0, where $P_n = (g_n, S_{n,A_c}, Q_{n,W}, \lambda_{n,N})$ and $P_n^* = (g_n, S_{n,A_c}, Q_{n,W}, \lambda_{n,N}^*)$.The g_n, S_{n,A_c} and $Q_{n,W}$ are never updated because they are tangent to our statistical parameter of interest and only $\lambda_{n,N}$ is updated. Here we illustrate one iteration of the targeting step and assume it has converged, while in practice one iteration is not enough and one might have to iterate many times until $\|\varepsilon_n\|$ is small or explicitly check the value of $\frac{1}{n}\sum_{i=1}^n D_t^*(P_n^*)(\boldsymbol{O}_i)$ smaller than a threshold. It is straightforward to show that the score of the coefficient ε at $\varepsilon = 0$ evaluated at a typical observation $\boldsymbol{O}$, equals $D_t^*(P_n)(\boldsymbol{O})$; thus, we may deduce that the EIC estimating equation is satisfied by the updated failure event conditional survival function $S_{n,N}^*$. The TMLE $\boldsymbol{\Psi}_n^*$ of the treatment-specific marginal survival curve is computed as the plug-in estimator based on the modified conditional survival function, $\boldsymbol{\Psi}(Q_n^*)(t) = \boldsymbol{\Psi}(S_{n,N}^*, Q_{n,W})(t) = \int S_{n,N}^*(u|A=1,\boldsymbol{W})dQ_{n,W}(u) = \frac{1}{n}\sum_{i=1}^n S_{n,N}^*(t|A=1,\boldsymbol{W}_i)$.

Under regularity conditions on the initial estimates $S_{n,N}$, S_{n,A_c} and g_n, the TMLE is regular and asymptotically linear [26], so $\sqrt{n}[\boldsymbol{\Psi}_n^*(t) - \boldsymbol{\Psi}_0(t)] \to^d N(0, \sigma_t^2)$. When $S_{n,N}$, S_{n,A_c} and g_n are consistent estimators for $S_{0,N}$, S_{0,A_c} and g_0, the variance σ_t^2 is the variance of the EIC. In order to estimate the variance σ_t^2, we can use an estimate of the sample variance of the EIC. Wald type hypothesis tests can be performed, and confidence intervals can be constructed with the estimated variance $\widehat{\sigma}_t^2$. TMLE is also double robust in the sense that the TMLE is consistent if either (a) the propensity score $g(W)$ and the censoring event conditional survival probability $S_{A_c}(A, \boldsymbol{W})$ are consistently estimated or (b) the failure event conditional survival probability $S_N(A, \boldsymbol{W})$ is consistently estimated.

Motivation: Why existing TMLE for survival curve is not monotone

The existing TMLE for the marginal treatment-specific survival curve can be viewed as an application of TMLE in Section 1.4 repeated for survival probabilities at $t = 1, ..., t_{max}$. The steps for the TMLE algorithm outlined in Section 1.4 can be summarized in the pseudo-code as follows:

Algorithm 1: iterative TMLE for survival curve

Data: initial estimator: conditional hazard for failure event, conditional survival curve for censoring event, propensity score

Result: TMLE for the counter-factual marginal survival curve $\mathbf{\Psi}_{A=1}$

for $t = 1, ..., t_{max}$ **do**
  initialize $S^{(0)} = S_{n,N}$ with the initial estimator for the survival curve of the failure event;
  $j = 0$;
  while *True* **do**
    for $i = 1, ..., n$ **do**
      for $k = 1, ..., \widetilde{T}_i$ **do**
        evaluate $h^{(j)}_{(i,k)} = h_t(g_{n,A}, S_{n,A_c}, S^{(j)})(k, A_i, \boldsymbol{W}_i)$;
        evaluate $N_{(i,k)} = I(\widetilde{T}_i = k, \Delta = 1)$;
        evaluate $\lambda^{(j)}_{(i,k)} = \lambda^{(j)}(k, A = 1, \boldsymbol{W}_i)$;
      end
    end
    concatenate into vectors $h^{(j)}$, N and $\lambda^{(j)}$;
    get $\widehat{\varepsilon}$ by running a logistic regression $\text{logit}N = \text{logit}(\lambda^{(j)}) + \varepsilon h^{(j)}$;
    evaluate $\lambda^{(j+1)} = \text{expit}[\text{logit}(\lambda^{(j)}) + \widehat{\varepsilon} h^{(j)}]$;
    transform to $S^{(j+1)}$;
    $j+ = 1$;
    if $|\widehat{\varepsilon}| \leq 1e-3$ **then**
      break
    end
  end
  $\mathbf{\Psi}^*(t) = \frac{1}{n} \sum_{i=1}^{n} S_i^{(j)}(t)$;
end
concatenate the $\mathbf{\Psi}^*(t)$ to get the entire curve $\mathbf{\Psi}^*(t), t = 1, ..., t_{max}$;

Note that the method creates t_{max} different $\lambda^*_{n,N,\widetilde{t}}, \widetilde{t} = 1, ..., t_{max}$ for each $\mathbf{\Psi}(\widetilde{t})$ task, therefore transforming the multiple $\lambda^*_{n,N,\widetilde{t}}$ into survival probabilities does not create a monotone decreasing survival curve.

1.5 One-step TMLE targeting the entire survival curve

The logistic submodel we use in the previous section is also called the local least favorable submodel (LLFM) around $\lambda_{n,N}$:

$$\text{logit}[\lambda_{n,N,\varepsilon}(k|A=1,\boldsymbol{W})] = \text{logit}[\lambda_{n,N}(k|A=1,\boldsymbol{W})] + \varepsilon h_{(k)}, \tag{1.11}$$

because it has the property that

$$\frac{d}{d\varepsilon}\log\frac{dP_{n,\varepsilon}}{dP}|_{\varepsilon=0} = D_t^*(P_n),$$

where $D_t^*(P_n)$ is the short notation for the EIC at $(g_{n,A}, S_{n,A_c}, \lambda_{n,N})$ and $P_{n,\varepsilon}$ is the distribution at $(g_{n,A}, S_{n,A_c}, \lambda_{n,N,\varepsilon})$. This is a key result that ensures TMLE is solving the EIC estimating equation by running a logistic regression along the submodel (1.11), but it also implies that the results hold only if we use the submodel around $\varepsilon = 0$, that is, we don't update along the submodel with a large step size $\widehat{\varepsilon}$. Doing a logistic regression on this submodel (1.11), however, does not guarantee that $\widehat{\varepsilon} \approx 0$. This intuition explains why doing TMLE on a high-dimensional parameter can often lead to diverging results, because TMLE is an iterative algorithm and because the first few iterations usually involve large step sizes.

[21] proposed a novel targeting step to modify the initial estimators called one-step TMLE. The idea is that since the gradient equals the EIC only locally when we update the initial estimators, one-step TMLE only performs the update locally. If we make the step size small enough, the submodel has the property that at any ε

$$\frac{d}{d\varepsilon}\log\frac{dP_{n,\varepsilon}}{dP} = D_t^*(P_{n,\varepsilon}) = D^*(\lambda_{n,N,\varepsilon}, Q_{n,W}, g_n).$$

This submodel is known as the universal least favorable submodel (ULFM) around $\lambda_{n,N}$, which takes the form

$$\begin{aligned}\text{logit}[\lambda_{n,N,\varepsilon}(k|A=1,\boldsymbol{W})] = \text{logit}[\lambda_{n,N}(k|A=1,\boldsymbol{W})] + \\ \int_0^{\varepsilon} h_t(g_{n,A}, S_{n,A_c}, S_{n,N,x})(k,1,\boldsymbol{W})dx.\end{aligned} \tag{1.12}$$

This theoretical formulation gives an insight into how this methodology works, but is not useful when analyze our survival curve problem because it involves integration of a complex function of $S_{n,N,x}$ (which itself is a function of $\lambda_{n,N,x}$).

In execution, the one-step TMLE is carried out by many LLFMs (performed in logistic regressions) with small step sizes. The one-step TMLE updates in small steps locally along LLFM, making sure only using the update direction $h_t(.)$ that is optimal around the current probability density. One-step TMLE also allows the analyst to update the conditional hazard for all points on the survival curve (or any high-dimensional parameter in general), so that the

conditional hazard can be transformed into a monotone survival curve after the algorithm. To do this, one replaces the univariate $h_t(.)(k, 1, \boldsymbol{W})$ in (1.11) with a high dimensional vector $\boldsymbol{h}_t(.) = [h_t(.)(1, 1, \boldsymbol{W}), ..., h_t(.)(t_{max}, 1, \boldsymbol{W})]$, each one corresponding to the clever covariate of survival probability at one time point. Fitting the high-dimensional logistic regression will not hurt the performance since we never update with large step size. Another way to view the one-step TMLE is that the logistic regression we used within classic TMLE is replaced with a logistic ridge regression, where the coefficient L-2 norm is constrained to be smaller than a tiny value. Because the logistic ridge regression generally outperforms classic logistic regression in high dimensions, the one-step TMLE is better than classic TMLE for high-dimensional target parameters. Given the same input and output, one-step TMLE leads to a new targeting procedure. The essential steps becomes the pseudo-code Algorithm 2 as follows, where the differences between one-step TMLE and classic TMLE are highlighted.

Algorithm 2: one-step TMLE for the survival curve

Data: initial estimator: conditional hazard for failure event, conditional survival curve for censoring event, propensity score

Result: TMLE for the counter-factual marginal survival curve $\boldsymbol{\Psi}_{A=1}$

initialize $S^{(0)} = S_{n,N}$ with the initial estimator for the survival curve of the failure event;
$j = 0$;
while *True* **do**
  for $i = 1, ..., n$ **do**
    for $k = 1, ..., t_{max}$ **do**
      evaluate $N_{(i,k)} = I(\widetilde{T}_i = k, \Delta = 1)$;
      evaluate $\lambda^{(j)}_{(i,k)} = \lambda^{(j)}(k, A = 1, W_i)$;
      for $t' = 1, ..., t_{max}$ **do**
        evaluate $h^{(j)}_{(i,k,t')} = h_{t'}(g_{n,A}, S_{n,A_c}, S^{(j)})(k, A_i, \boldsymbol{W}_i)$;
      end
      concatenate into vector $\boldsymbol{h}^{(j)}_{(i,k)}$;
    end
  end
  concatenate along (i, k) indices (by row) into vectors N, $\lambda^{(j)}$ and matrix $\boldsymbol{h}^{(j)}$;
  get $\widehat{\boldsymbol{\varepsilon}}$ by running a logistic ridge regression $\text{logit}N = \text{logit}(\lambda^{(j)}) + \boldsymbol{\varepsilon}\boldsymbol{h}^{(j)}$ subject to $\|\boldsymbol{\varepsilon}\| \leq 1e-2$;
  evaluate $\lambda^{(j+1)} = \text{expit}[\text{logit}(\lambda^{(j)}) + \widehat{\boldsymbol{\varepsilon}}\boldsymbol{h}^{(j)}]$;
  transform to $S^{(j+1)}$;
  $j+ = 1$;
  if $\|\widehat{\boldsymbol{\varepsilon}}\| \leq 1e-3$ **then**
    break
  end
end
$\boldsymbol{\Psi}^*(t) = \frac{1}{n}\sum_{i=1}^{n} S_i^{(j)}(t), t = 1, ..., t_{max}$;

Note: With abuse of notation, we define $h_{(i,k,t')} = h_{t'}(g_{n,A}, S_{n,A_c}, S_{n,N})(k, A_i, \boldsymbol{W}_i)$ to include an additional subscript t' referring to the clever covariate for estimating $\boldsymbol{\Psi}(t')$ evaluated at observation $\boldsymbol{O}_i$.

Inference

The statistical inference of iterative and one-step TMLE at a single time point can be done in the same procedure. The TMLE estimators, both iterative and one-step, solve the efficient

influence curve equation:

$$\frac{1}{n}\sum_{i=1}^{n} D_t^*(P_n^*)(\boldsymbol{O}_i) = 0, t = 1, ..., t_{max}.$$

Thus, if all components are consistent and under regularity conditions, TMLE is asymptotically linear with influence curve $D_t^*(P_0)$ [25]. Based on this result, TMLE inference is based on the empirical variance of the efficient influence curve $D_t^*(P_n^*)$, assuming the initial estimators (S_N, g_A, S_{A_c}) are consistent. Thus, the asymptotic variance of $n^{1/2}[\boldsymbol{\psi}_n^*(t) - \boldsymbol{\psi}_0(t)]$ is estimated by:

$$\widehat{\sigma}_t^2 = \frac{1}{n}\sum_{i=1}^{n} D_t^{*2}(P_n^*)(\boldsymbol{O}_i).$$

Now a valid $100 \times (1-\alpha)\%$ confidence interval is constructed under the normal distribution in the following way:

$$\boldsymbol{\psi}_n^*(t) \pm q_{1-\alpha/2}\frac{\widehat{\sigma}_t}{\sqrt{n}},$$

where q_β is the β-quantile of the standard normal distribution.

Simultaneous confidence interval

The simultaneous confidence bands for the survival curve estimates can be similarly constructed based on asymptotic linearity of the TMLE uniform in all time points considered. Inference for $\boldsymbol{\psi}_n^*$, the vector of survival probabilities at t_{max} time points, a vector parameter, is also based on the empirical variance of the efficient influence curve $\boldsymbol{D}^*$ itself at the limit of (S_N^*, g_A, S_{A_c}). The asymptotic variance of $n^{1/2}(\boldsymbol{\psi}_n^* - \boldsymbol{\psi}_0)$ may be consistently estimated by the t_{max} by t_{max} empirical covariance matrix of the efficient influence curve:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{D}^*(P_n^*)(\boldsymbol{O}_i)[\boldsymbol{D}^*(P_n^*)(\boldsymbol{O}_i)]^\top.$$

By multivariate central limit theorem, we have

$$n^{1/2}(\boldsymbol{\psi}_n^* - \boldsymbol{\psi}_0) \xrightarrow{d} N(\mathbf{0}, \boldsymbol{\Sigma}_\mathbf{0}). \tag{1.13}$$

As a result, an approximate $100 \times (1-\alpha)\%$ simultaneous confidence band is constructed such that for each $\boldsymbol{\psi}(t)$, the t^{th} component of $\boldsymbol{\psi}$, the region is given by

$$\boldsymbol{\psi}_n^*(t) \pm q_{1-\alpha}\widehat{\boldsymbol{\Sigma}}^{1/2}(t)/\sqrt{n},$$

where $\widehat{\boldsymbol{\Sigma}}(t)$ is the (t,t)-th entry in the empirical covariance matrix, thus the empirical variance of D_t^*. $q_{1-\alpha}$ is an estimate of the $1-\alpha$ quantile of $\max_t \sqrt{n}|\boldsymbol{\psi}_n^*(t) - \boldsymbol{\psi}_0(t)|/\widehat{\boldsymbol{\Sigma}}^{1/2}(t)$. Here

we need to use that the latter random variable behaves as the max over t of $\boldsymbol{Z}(t)$, where $\boldsymbol{Z} \sim N(\mathbf{0}, \boldsymbol{\rho})$ follows t_{max}-dimensional gaussian and $\boldsymbol{\rho}$ is the correlation matrix of the vector influence curve $\boldsymbol{D}^*(P_n^*)(\boldsymbol{O}_i)$. We simulate Monte-Carlo samples of $\boldsymbol{Z}$ and calculate $q_{1-\alpha}$ using the empirical $1-\alpha$ quantile of $\max_t |\boldsymbol{Z}|$ of the random samples. Due to actual weak convergence of the standardized TMLE as a random function in function space endowed with supremum norm, these simultaneous confidence bands are valid even as we take a finer and finer grid of time points as n increases.

1.6 Simulation

To provide an example of the finite sample properties of the estimators discussed in Sections 1.4 and 1.5, we simulate a univariate continuous baseline covariate W, a binary exposure A, a survival outcome T with censoring time C. We simulate data from the following data-generating distribution so that T, A, and C are confounded by W:

$$\begin{aligned} W &\sim \text{Unif}(0, 1.5), \\ A &\sim \text{Bernoulli}(0.4 + 0.5I\{W > 0.75\}), \\ T &\sim \text{log-normal}(\mu = 2 - W + A, \sigma = 0.01), \\ C &\sim \text{Weibull}(1 + 0.5W, 75). \end{aligned}$$

To analyze the above simulated data, we estimate the survival curves under the treatment and control groups. For sample sizes n = 100 and 1000, we simulated 1000 Monte-Carlo repetitions from the previous data-generating distribution, and estimated $\boldsymbol{\Psi}_{A=1}(P_0)$ and $\boldsymbol{\Psi}_{A=0}(P_0)$ using the following estimators: Kaplan-Meier; plug-in SuperLearner estimator of the conditional survival curve [23]; IPCW; EE; classic (iterative) TMLE; one-step TMLE targeting the whole curve. As initial estimators of the components of the likelihood $(g_0, S_{0,A_c}, \lambda_{0,N})$, we used SuperLearner classification combining multiple classification algorithms so that we know the estimates will be consistent. The SuperLearner library includes generalized linear model [39], generalized additive model [16], and multivariate adaptive regression splines [11]. We used empirical distribution $Q_{n,W}$ to estimate $Q_{0,W}$. One-step TML estimation was performed using the R function 'MOSS_hazard' in the open-source package MOSS [5], and the code that reproduces this simulation is presented in Web Appendix. The average and variance of the estimates across the 1000 samples was computed as an approximation to the expectation and variance of the estimator, respectively. We report the bias, variance, mean-squared error (MSE) of different estimators in Figure 1.2, and we use the MSEs to further calculate the relative efficiencies (RE) against iterative TMLE for all estimators:

$$RE_{\text{estimator}}(t) = \frac{MSE_{\text{iterative TMLE}}(t)}{MSE_{\text{estimator}}(t)}, t = 1, ..., t_{max}.$$

The simulation results reflect what is expected based on theory. Figure 1.1 are examples in the simulation where the EE and classic TMLE methods do not produce monotone survival

curves. Figure 1.2 computes the metrics at different time points of the entire survival curve. One-step TMLE methods has lowest MSE under all sample sizes, with 33% smaller MSE than the second best method (iterative TMLE) in small sample size. EE has a large variance in small sample size (n = 100) and its MSE becomes more comparable to iterative TMLE in larger sample size (n = 1000). Kaplan-Meier is not consistent and has large MSE especially in large samples, although in finite samples its bias is not large compared to its variance. IPCW has the largest variance and MSE under all sample sizes. As sample size increases one-step TMLE converges to iterative-TMLE, and both TMLEs are better than IPCW, EE and Kaplan-Meier.

In Section 1.5, we gave intuition that the universal least favorable submodel can be viewed as a ridge logistic regression applied in the targeting step. Curious readers might be interested in the performance if we use a LASSO logistic regression instead. We also experiment this in the simulation (marked by 'OS TMLE (lasso)', while our proposed one-step TMLE is denoted 'OS TMLE (ridge)'), and we see that the difference between the two kinds of penalizations is small: both types of one-step TMLE outperforms iterative TMLE in finite sample and converge to iterative TMLE in the asymptotic. We find that using LASSO logistic regression improves MSE in large t (where there are fewer samples) at the cost of a slightly larger MSE in small t. Therefore, we only recommend using LASSO logistic regression for targeting step when minimax guaranteed improvement (across t) on the iterative TMLE is preferred.

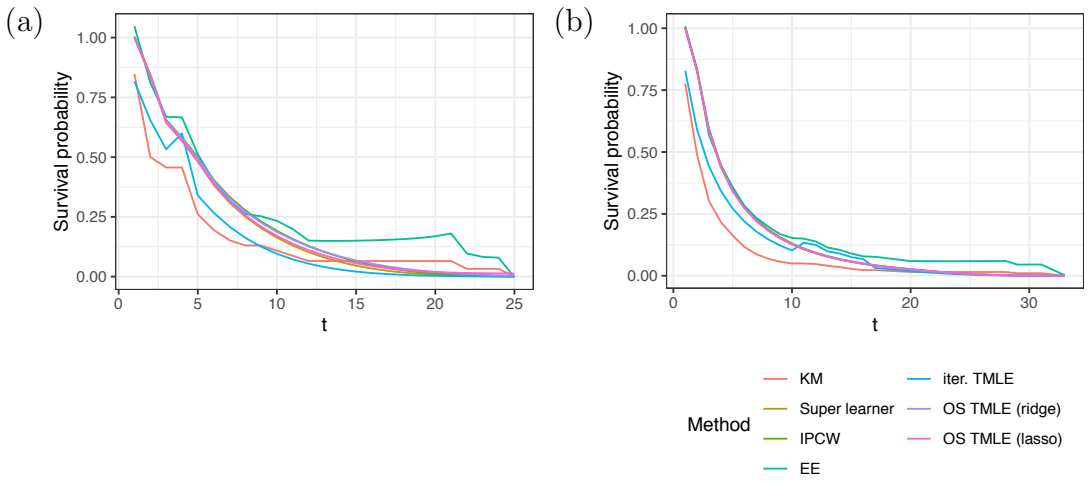

Figure 1.1: Examples of non-monotone EE and TMLE estimators in simulation data of different sample sizes (plot a: n = 100, plot b: n = 1000). The target parameter is the marginal counter-factual survival curve for the treatment group $\mathbf{\Psi_1}(P)$.

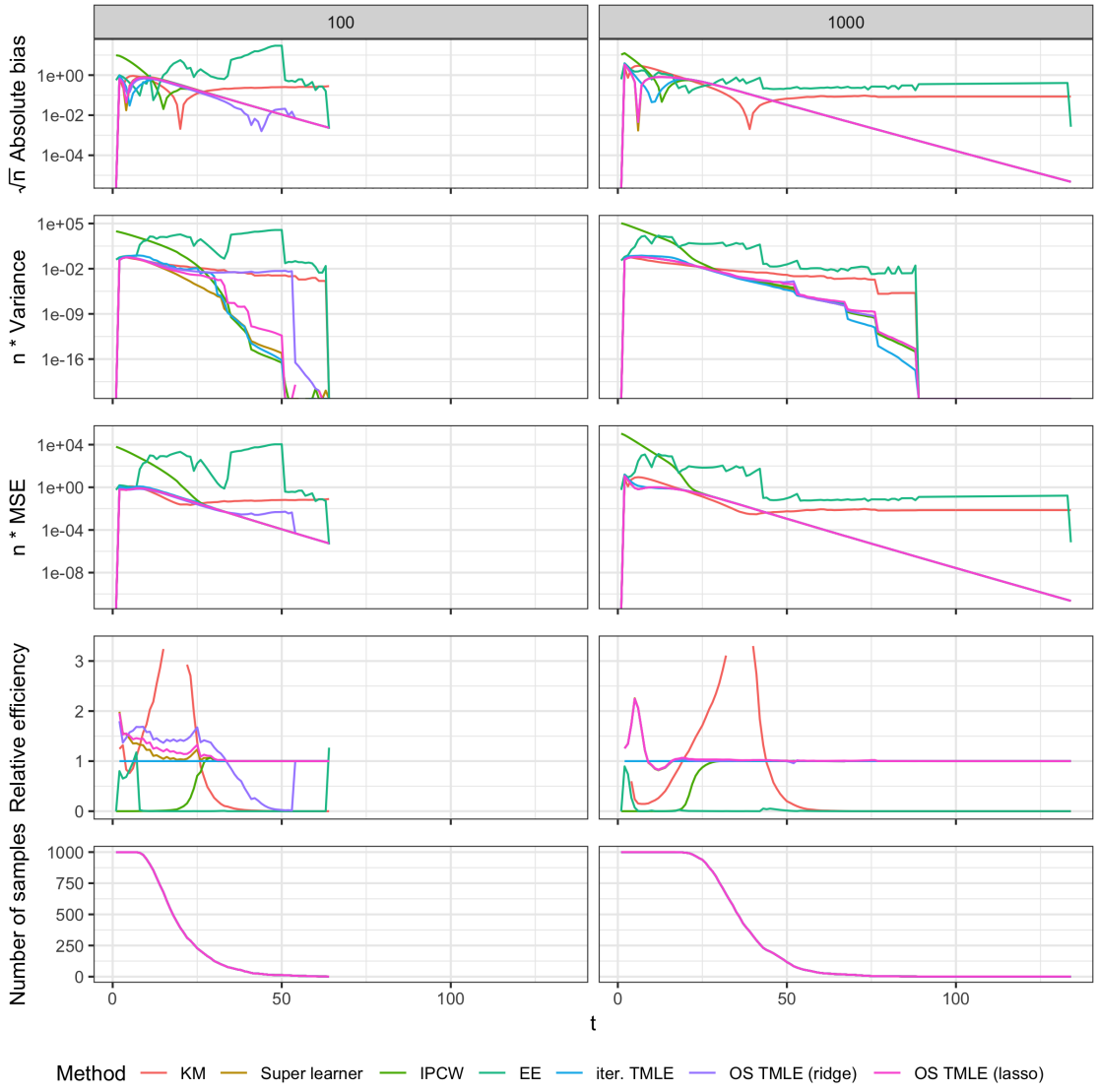

Figure 1.2: Results for comparing different survival curve estimators at all time points. Row 1 is absolute bias times $\sqrt{n}$, row 2 is variance times n, row 3 is MSE times n, row 4 is relative efficiency (larger than 1 means more efficient than iterative TMLE), row 5 is the number of simulations where follow up time is at least t. Within each row, the left plot is under sample size 100 and the right plot is under sample size 1000. Note the relative efficiency value larger than 4 are truncated so that the plot range around [0,1] can be easily interpreted.

1.7 Data analysis

To illustrate the finite sample performance of the one-step TMLE, we use a dataset from a classic monoclonal gammopathy study, an observational survival analysis dataset that first

established the predictive relationship between the initial concentration of serum monoclonal protein and the progression to multiple myeloma or another plasma-cell cancer [19]. For each subject, we define the (right-censored) outcome $\widetilde{T}$ as the time until progression to a plasma cell malignancy or last contact, the treatment A as the monoclonal spike on serum protein electrophoresis (1 = the spike is higher than 1.5 g/dL, 0 = the spike is lower than 1.5 g/dL), and include all baseline covariates $\boldsymbol{W}$ (age, gender, hemoglobin, creatinine) that are measured upon enrollment of the subjects. The original study is on the predictive power of A on the outcome and not the causal relationship, so there are definitely unmeasured confounders left out from this dataset. Nonetheless, we use the data to illustrate the statistical properties of different estimators. The trial measured 1338 complete cases after we discarded 46 subjects with missing data. We find that there is a practical violation of positivity assumption for time larger than 160 months. Therefore, we perform manual truncation of the dataset so that observations with follow-up time beyond 160 months are censored. We also transform the time unit of the dataset for ease of computation $\widetilde{T}_{new} = \lceil \widetilde{T}/20 \rceil$, and we verify that this transformation does not change the scientific results. The preprocessed data contain 405 patients in the treatment group and 933 patients assigned to control.

We first estimate the marginal survival curve for the treatment and control groups. We compare a plug-in parametric fit using generalized linear model, plug-in Super learner fit, IPCW, EE, classic TMLE, and one-step TMLE targeting the whole curve. The Super learner initial fits combine main term generalized linear model, main term generalized additive model [16], main term multivariate adaptive regression splines [11], and random forest [4]. The same learner library is used for fitting the conditional survival for failure event and censoring event, as well as the propensity score. The conditional survival functions estimated by Super learner [23] are presented in Figure 1.3. There is a complex interaction effect between baseline covariates (age and hemoglobin) and time in the conditional hazard of censoring event, so it is crucial to use nonparametric regression methods such as Super learner to fully adjust for the confounders.

Figure 1.4(A) shows the different estimators' results for the treatment and control group survival curves. The one-step TMLE, TMLE, EE and Super learner fits are close to each other, suggesting that the dataset is large. EE is slightly not monotone for the treatment group survival curve. IPCW is drastically different from all other estimators, which is the worst performing method. Second, the delta method is applied to obtain the estimators for the difference in survival probabilities (treatment minus control). Wald 95% confidence bands for EE and TMLE are calculated using the efficient influence curve. Super learner is different from the parametric fit, suggesting that nonparametric regression is crucial for this analysis. EE is not monotone. Lastly, to check how well the estimators perform in a finite sample, we randomly subsample the pre-processed data into smaller sizes and re-compute all methods. The procedure is repeated 100 times, and we count how frequent each estimator yields a monotone curve. The percentages are reported in Table 1.1. We find that EE has the highest probability of becoming not monotone when all other conditions held equal. Classic TMLE outputs a monotone survival curve at least 80% of the times, and one-step TMLE is

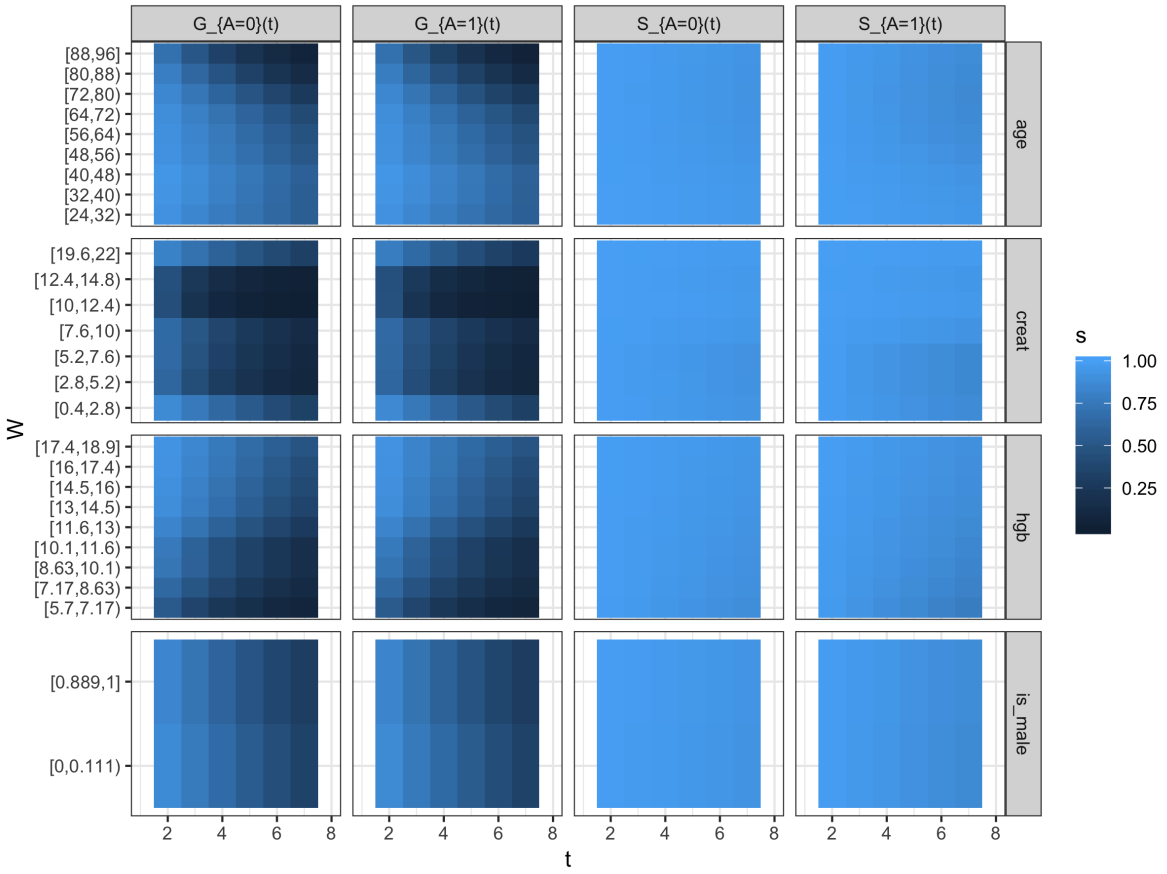

Figure 1.3: Partial dependency plots of the initial super learner fits for the conditional survival curves, where the y-axis is the baseline covariate value, the x-axis is time. Column 1 is the conditional survival of censoring event for control group; Column 2 is the conditional survival of censoring event for treatment group; Column 3 is the conditional survival of failure event for control group; Column 4 is the conditional survival of failure event for the treatment group. Row 1 plots have age on the y-axis; Row 2 plots have creatinine on the y-axis; Row 3 plots have Hemoglobin on the y-axis; Row 4 plots have gender indicator on the y-axis.

guaranteed to be monotone.

1.8 Discussion

In this paper, we provided a one-step TMLE for estimating the treatment-specific survival curve while targeting the entire survival curve at once. The one-step estimator has implications for the survival analysis literature by allowing one to construct a TMLE for the infinite

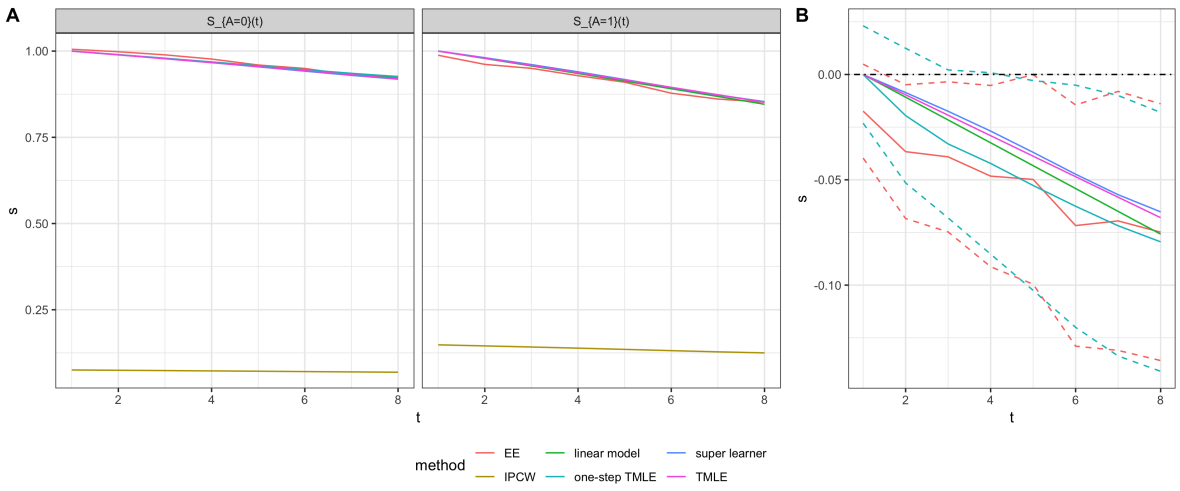

Figure 1.4: Results for different counterfactual survival curve estimators on the Monoclonal gammopathy data. Panel A is survival curve estimates for the control group and treatment group, using different estimators. Panel B is the difference curve in survival probabilities (treatment group minus control group), using different estimators.

(a)

n	EE	TMLE	OS TMLE
100	42%	91%	100%
500	74%	93%	100%
1000	100%	100%	100%

(b)

n	EE	TMLE	OS TMLE
100	38%	81%	100%
500	90%	93%	100%
1000	100%	100%	100%

Table 1.1: For each method and subsample size, the percentage of experiments when the estimator outputs a monotone survival curve in the monoclonal gammopathy study (a: for the treatment group; b: for the control group).

dimensional survival curve in a single step. The new method is asymptotically linear and efficient, just as the iterative TMLE, which adjusts for baseline covariates and accounts for informative censoring through inverse weighting. Additionally, the one-step estimator targeting the entire survival curve respects the monotonically decreasing shape of the estimand. On top of that, the new TMLE for the whole curve also yields a fully compatible TMLE for any function of the entire survival curve, such as the median, quantile, or truncated mean. Thus there is no need to compute a new TMLE for each specific feature of the survival curve, or difference of survival curves. All of these advantages come without requiring any parametric modeling assumptions and is robust to misspecification of the hazard fit. Our simulation confirms the theory in the existing literature: that in situations where targeting is difficult due to high-dimensional estimation scores, using one-step TMLE that fluctuates universal least favorable submodel may provide more robustness and efficiency over iterative TMLE. Under large sample sizes, iterative and one-step TMLE are comparable. We show

that in practical finite sample situations for survival analysis, using universal least favorable submodel to target a multi-dimensional or even infinite-dimensional target parameter is likely to result in a more efficient and stable estimator. It is not clear how our methods compare with applying isotonic regression to the curve defined by the one-step TMLEs targeting one survival probability repeated across all time-points, which represents another valid and possible method to consider if getting the whole survival curve is the goal of the analysis.

Chapter 2

Nonparametric Bootstrap Inference for the Targeted Highly Adaptive LASSO Estimator

2.1 Introduction

We consider the estimation of a pathwise differentiable real-valued target parameter based on observations from a data distribution known to belong in a highly non-parametric statistical model. Targeted Minimum Loss Estimator (TMLE) [27] is an asymptotically unbiased and efficient (substitution) estimator. A TMLE that uses the highly-adaptive LASSO minimum loss-based estimators (HAL-MLE, [1]) as initial estimators for the nuisance parameters is called HAL-TMLE [28].

If the nuisance parameters are in the cadlag function space and have finite sectional variation norm, the HAL-MLEs will converge with respect to (w.r.t) the loss-based dissimilarities at a rate faster than $n^{-1/2}$. Therefore, the HAL-TMLE has been shown to be asymptotically efficient under weaker regularity conditions than TMLE without using HAL [28]. Statistical inference of TMLE (HAL-TMLE included) is usually made based on the normal limit distribution where the asymptotic variance is estimated with an estimator of the variance of the efficient influence curve. This Wald-type confidence interval is asymptotically consistent but directly applying this interval in the finite sample can lead to anti-conservative coverage when the second order remainder term can easily dominate the first order term in the parameter expansion (i.e., the empirical mean of the efficient influence curve).

A natural idea is to perform bootstrap [10], but bootstrap had no theoretical asymptotic validation, due to use of cross-validation/ adaptive machine learning [8, 2, 15]. The conventional approach is to still use the Wald-type interval for finite sample inference. In non-doubly robust problems, the Wald-type interval is shown to be anti-conservative in finite samples. For problems that we know are doubly-robust, the Wald-type interval can sometimes be anti-conservative as well, if one does not use a carefully crafted variance estimator,

which resulted in the variance estimator to be sensitive to positivity. We direct readers to [41, 22, 45, 40] for a detailed discussion, and current applications to longitudinal data (longitudinal TMLE) is another example [41]. Even with a robust variance estimator, the Wald-type interval is ignoring the second order term, so the curse of dimensionality could easily cause trouble for finite sample coverage.

The literature on resampling machine learning algorithms to produce finite sample inference is usually algorithm-specific and only apply to limited families of parameters such as prediction interval. For example, [49] develops infinitesimal jackknife method to create inference for random forest algorithm. Deep learning literature [12, 37] considers using stochastic regularization and optimization techniques to approximate Bayesian inference of the neural network. An alternative approach performs parametric bootstrap that samples from a nonparametric generative model, such as the targeted bootstrap proposed by [8]. Targeted bootstrap performs resampling from a continuous distribution estimate rather than from the empirical distribution as in non-parametric bootstrap. Another distinction between this paper and targeted bootstrap is that [8] performs extra targeting practice to consistently estimate the second moment of the parameter sampling distribution, while in this paper we approximate all higher orders of the distribution, which can be used as a plug-in estimator of all kinds of summary statistics of the sampling distribution.

Recent seminal work by [30] show that non-parametric bootstrap for HAL-TMLE is valid in finite samples. The bootstrap methods in [30] are investigated and worked out in detail in this article in particular examples, and our work results in a newly proposed modification of the non-parametric bootstrap, which is very robust in finite samples respect to coverage. To demonstrate the methods we consider one doubly-robust example and two non-doubly robust examples. The examples can be easily generalized to existing applications of TMLE. We evaluate based on finite sample interval coverage. Our results confirm theory in [30] that the bootstrap HAL-TMLE interval is more effective than the Wald-type interval in finite sample regarding having better confidence interval coverage, given that both of them are asymptotically valid methods. On top of that, a tuning parameter selection method that optimizes the bootstrap interval coverage is very crucial for non-doubly robust parameters as we show in the examples. For doubly-robust problems, the regular bootstrap method works well, and our modifications are less critical. This chapter contains part of the results by the same authors [6].

Organization of article In Section 2.2 we formulate the statistical problem, review the HAL-TMLE bootstrap estimator and present extensions using an example of the average treatment effect estimation. In particular, Section 2.2 sets up the notation and motivates the bootstrap problem. Section 2.2 outlines the HAL-TMLE bootstrap method. Section 2.2 presents the challenge of choosing the tuning parameter for optimal bootstrap inference and gives intuition that motivates our procedure for choosing the optimal tuning parameter, and we propose a "plateau tracking"method for choosing the optimal tuning parameter. We also discuss an approach for improving coverage when the estimator has finite sample

bias in Section 2.2. In addition to describing our method for a general estimation problem, in Section 2.3 we work out how to implement our proposed HAL-TMLE bootstrap in three target parameter examples and in Section 2.4 we put them into action using three simulations and give our simulation results. We apply our method to two popular public datasets in Section 2.5. Finally, we conclude our work and discuss potential extensions in Section 2.6.

2.2 Methodology

Set up: HAL-TMLE of a parameter

Let $O_1, ..., O_n$ be n i.i.d. copies of a random variable $O \sim P_0 \in \mathcal{M}$, where $\mathcal{M}$ is a non-parametric statistical model. Define P_n be the empirical probability measure of $O_1, ..., O_n$. Let $\Psi : \mathcal{M} \to \mathbb{R}$ be a real-valued parameter that is pathwise differentiable at each $P \in \mathcal{M}$, with canonical gradient $D^*(P)$, that is, the Taylor expansion of Ψ at each $P \in \mathcal{M}$ is

$$\Psi(P) - \Psi(P_0) = (P - P_0)D^*(P) + R_2(P, P_0), \tag{2.1}$$

where R_2 is the second-order remainder of the expansion. For any pathwise differentiable parameter Ψ, we can find a function-valued parameter $Q : \mathcal{M} \to Q(\mathcal{M})$ so that $\Psi(P) = \Psi_1(Q(P))$ for some Ψ_1. For notational convenience, we will refer to the target parameter with $\Psi(Q)$ and $\Psi(P)$ interchangeably. Let $G : \mathcal{M} \to G(\mathcal{M})$ be a function-valued parameter so that $D^*(P) = D_1^*(Q(P), G(P))$ for some D_1^*. Again, we will use notations $D^*(P)$ and $D^*(Q, G)$ interchangeably. We can define the exact second-order remainder from (2.1) as

$$R_2(P, P_0) = \Psi(P) - \Psi(P_0) + (P - P_0)D^*(P), \tag{2.2}$$

where $(P - P_0)D^*(P) = -P_0D^*(P)$ since $D^*(P)$ has mean zero under P.

Example: (Average treatment effect)

Let $O = (W, A, Y) \sim P_0 \in \mathcal{M}$, where $A \in \{0, 1\}$ is a binary treatment, and $Y \in \mathbb{R}$ is a continuous outcome. For a possible data distribution P, let $\bar{Q}(P) = \mathbb{E}_P(Y|A, W)$, $G(P) = P(A = 1|W)$ be the outcome regression, $G(P) = P(A = 1|W)$ be the propensity score, and let $Q_W(P)$ be the probability distribution of W. The average treatment effect (ATE) parameter is defined by $\Psi(P) = \mathbb{E}_P[\mathbb{E}_P(Y|A = 1, W) - \mathbb{E}_P(Y|A = 0, W)]$. Let $Q = (\bar{Q}, Q_W)$ and note that the data distribution P is determined by (Q, G). The canonical gradient of Ψ at P is

$$D^*(Q, G) = \frac{I(A = a)}{G(A|W)}(Y - \bar{Q}(A, W)) + \bar{Q}(1, W) - \Psi(Q).$$

The second-order remainder $R_2(P, P_0) \equiv \Psi(P) - \Psi(P_0) + P_0 D^*(P)$ is given by:

$$R_2(Q, G, Q_0, G_0) = \int \frac{(G - G_0)(1|w)}{G(1|w)}(\bar{Q} - \bar{Q}_0)(1, w) - \frac{(G - G_0)(0|w)}{G(0|w)}(\bar{Q} - \bar{Q}_0)(0, w) dP_0(w)$$

The HAL-TMLE procedure consists of two steps: (1) HAL-MLE fitting of the Q and G part of the likelihood and (2) TMLE step that update the initial fit for Q. For step 1, given L_1 and L_2 are the loss functions that identify the true Q_0 and G_0, we apply two HAL-MLEs Q_n and G_n that estimate Q_0 and G_0, where the tuning parameters λ_1 and λ_2 are chosen with cross-validation. For the TMLE step, consider a local least favorable submodel (LLFM) $\{Q_{n,\varepsilon} : \varepsilon\} \subset Q(M)$ through Q_n at $\varepsilon = 0$ so that the linear span of the components of $\frac{d}{d\varepsilon} L_1(Q_{n,\varepsilon})$ at $\varepsilon = 0$ includes $D^*(Q_n, G_n)$. Let $Q_n^* = Q_{n,\varepsilon_n}$ for $\varepsilon_n = \arg\min_\varepsilon P_n L_1(Q_{n,\varepsilon})$. We assume that this one-step TMLE Q_n^* already satisfies

$$|P_n D^*(Q_n^*, G_n)| = o_P(n^{-1/2}), \tag{2.3}$$

when in practice one can iterate multiple times through the submodel until this condition is satisfied. As shown in [28] this holds for the one-step HAL-TMLE under regularity conditions. Alternatively, one could use the one-dimensional universal least favorable submodel (ULFM) [31] or any other TMLE procedure compatible with the estimation problem. After targeting, the HAL-TMLE of Ψ_0 is now the plug-in estimator $\Psi_n^* = \Psi(Q_n^*)$, and it have been shown in [28] that Ψ_n^* is asymptotically efficient under the same regularity conditions in [28]. We restate the sufficient conditions for HAL-TMLE to be asymptotically efficient [28]:

Theorem 1 (asymptotic efficiency of HAL-TMLE)**.** *Consider the statistical model $\mathcal{M}$, target parameter $\Psi : \mathcal{M} \to \mathbb{R}$ and the model assumptions: (1) assume that the loss functions are uniformly bounded in the sense that* $\sup_{Q \in Q(\mathcal{M}),O} \mid L_1(Q)(O) \mid < \infty$ *and* $\sup_{G \in G(\mathcal{M}),O} \mid L_2(G)(O) \mid < \infty$ *[Formula (2) in 6], (2) the canonical gradient map into functions with a sectional variation norm bounded by some universal finite constant: $M_3 \equiv \sup_{P \in \mathcal{M}} \|D^*(P)\|_v^* < \infty$, (3) absolute value of the exact second order remainder is upper bounded [Formula (8) in 6], (4) continuity of efficient influence curve as a function of P at P_0 [Formula (9) in 6]. Let Q_n, G_n be the above defined HAL-MLEs, where we know $d_{01}(Q_n, Q_0)$ and $d_{02}(G_n, G_0)$ are $O_P(n^{-1/2-\alpha/4})$ ($d_{01}(Q, Q_0) = P_0 L_1(Q) - P_0 L_1(Q_0)$ is the loss-based dissimilarities for Q, and $d_{02}(G, G_0)$ is that for G) In addition, assume that the HAL-TMLE Q_n^* is such that it solves the efficient influence curve equation (2.3) up until $o_P(n^{-1/2})$.*

Then the HAL-TMLE $\Psi(Q_n^*)$ of ψ_0 is asymptotically efficient:

$$\Psi(Q_n^*) - \Psi(Q_0) = (P_n - P_0) D^*(Q_0, G_0) + O_P(n^{-1/2-\alpha/4}).$$

The Wald-type 0.95-confidence interval is given by $\Psi_n^* \pm 1.96\sigma_n/n^{1/2}$, where $\sigma_n^2 = P_n\{D^*(Q_n^*, G_n)\}^2$ is a consistent estimator of $\sigma_0^2 = P_0\{D^*(Q_0, G_0)\}^2$. The Wald-type interval is a first-order asymptotic confidence interval and ignores the exact remainder in the Taylor expansion (2.2). [30] shows that under high dimensions or complex models (in terms of large true sectional

variation norm of the Q and G functions), R_2 can outnumber first order term. Since Wald ignores the second-order term, directly applying Wald interval for finite sample inference can lead to anti-conservative results. The key is to get a higher order approximation of the sampling distribution of ψ_n.

Example: (Average treatment effect)

For the ATE parameter, the $\bar{Q}$ function is the outcome regression $E(Y|A,W)$, and $G = P(A=1|W)$ is the propensity score. Let $L_1(\bar{Q})(O) = -\{Y \log \bar{Q}(A,W) + (1-Y)\log(1-\bar{Q}(A,W))\}$ be the negative log-likelihood loss for the outcome regression. Similarly, $L_2(G)$ is the negative-log-likelihood loss for propensity score. When, for some $\delta > 0$, $G > \delta > 0$ and $\delta < \bar{Q} < 1-\delta$, then the loss functions are uniformly bounded with finite universal bounds.

The HAL-MLEs $\bar{Q}_n$ and G_n of $\bar{Q}$ and G, respectively, can be computed with a lasso-logistic regression estimator with large (approximately $n2^d$) number of indicator basis functions (see our example section for more details), where we can select the L^1-norm of the coefficient vector with cross-validation. The least favorable submodel through $\bar{Q}_n$ is given by

$$\text{logit}\bar{Q}_{n,\varepsilon} = \text{logit}\bar{Q}_n + \varepsilon C(G_n), \tag{2.4}$$

where $C(G_n)(A,W) \triangleq A/G_n(W)$. Let $\varepsilon_n \triangleq \arg\min_\varepsilon P_n L_1(Q_{n,\varepsilon})$, which is thus computed with a simple univariate logistic regression MLE, using as off-set $\text{logit}\bar{Q}_n$. This defines the TMLE $\bar{Q}_n^* = \bar{Q}_{n,\epsilon_n}$. Recall that $Q_{W,n}$ is already a nonparametric maximum likelihood estimate so that a TMLE-update based on a log-likelihood loss and local least favorable submodel (i.e., with score $\bar{Q}_n(W) - \Psi(Q_n)$, will not change this estimator. Let $Q_n^* = (Q_{W,n}, \bar{Q}_n^*)$. The HAL-TMLE of ψ_0 is the plug-in estimator $\psi_n^* \triangleq \Psi(Q_n^*) = \frac{1}{n}\sum_{i=1}^n \bar{Q}_n^*(1, W_i)$.

HAL-TMLE bootstrap

A valid method is the nonparametric bootstrap estimator of the sampling distribution of HAL-TMLE. Let $O_1^\#, ..., O_n^\#$ be n i.i.d. draws from the empirical measure P_n (training sample). Let $P_n^\#$ be the empirical measure of this bootstrap sample. Applying the HAL-TMLE algorithm on the bootstrap sample yields $Q_n^{\#^*}$ and $G_n^\#$, which can be used to construct a bootstrap-sample specific $\Psi_n^{\#^*}$. The whole procedure is repeated many times until a large enough sample of $\Psi_n^{\#^*}$ is collected and we use the bootstrap distribution as a proxy of the sampling distribution of Ψ_n, condition on P_n. The bootstrap based 0.95-confidence interval for Ψ_0 is given by

$$[\Psi_n^* - q_{0.975,n}^\#/n^{1/2}, \Psi_n^* - q_{0.025,n}^\#/n^{1/2}],$$

where $q_{\alpha,n}^\# \equiv F_n^{\#^{-1}}(\alpha)$ is the α-quantile of the bootstrap distribution of $Z_n^{1,\#} = n^{1/2}(\Psi(Q_n^{\#^*}) - \Psi(Q_n^*))$. [6] shows that this bootstrap confidence interval is asymptotically consistent for

the normal limit distribution of HAL-TMLE. We restate their theorem as follows:

Theorem 2 (Asymptotically consistency of HAL-TMLE bootstrap)**.** ***Assumption:*** *Assume the conditions of Theorem 1 providing asymptotic efficiency of $\Psi(Q_n^*)$; Consider the above defined HAL-MLEs Q_n, G_n satisfying, with probability tending to 1, $P_nL_1(Q_n) \leq P_nL_1(Q_0)$ and $P_nL_2(G_n) \leq P_nL_2(G_0)$. Consider also the above defined bootstrapped HAL-MLEs $Q_n^\#$, $G_n^\#$ satisfying, with probability tending to 1, conditional on $(P_n : n \geq 1)$, $P_n^\#L_1(Q_n^\#) \leq P_nL_1(Q_n)$ and $P_n^\#L_2(G_n^\#) \leq P_n^\#L_2(G_n)$. Consider the HAL-TMLE $Q_n^{\#*} = Q^\#_{n,\epsilon_n^\#}$ and assume $r_n^\# = P_n^\#D^*(Q_n^{\#*}, G_n^\#) = o_P(n^{-1/2})$.*

TMLE is efficient: *The standardized TMLE is asymptotically efficient: $Z_n^1 \equiv n^{1/2}(\Psi(Q_n^*) - \Psi(Q_0)) \Rightarrow_d N(0, \sigma_0^2)$, where $\sigma_0^2 = P_0D^*(Q_0, G_0)^2$.*

Bootstrapped HAL-MLE: *$d_{01}(Q_n^\#, Q_n) = O_P(n^{-1/2-\alpha/4})$, $d_{02}(G_n^\#, G_0) = O_P(n^{-1/2-\alpha/4})$ and $d_{01}(Q_n^{\#*}, Q_0) = O_P(n^{-1/2-\alpha/4})$.*

Bootstrapped HAL-TMLE: *Conditional on $(P_n : n \geq 1)$, the bootstrapped TMLE is asymptotically linear:*

$$\Psi(Q_n^{\#*}) - \Psi(Q_n) = (P_n^\# - P_n)D^*(Q_n, G_n) + O_P(n^{-1/2-\alpha/4}).$$

As a consequence, conditional on $(P_n : n \geq 1)$, the standardized bootstrapped TMLE converges to $N(0, \sigma_0^2)$: $Z_n^{1,\#} \equiv n^{1/2}(\Psi(Q_n^{\#*}) - \Psi(Q_n^*)) \Rightarrow_d N(0, \sigma_0^2)$.

Consistency of the nonparametric bootstrap for HAL-TMLE at data adaptive selector C_n^u: *Assume the extra model structure on $\mathcal{M}$ such that Q and G are both cadlag functions with bounded sectional variation norm, and its corresponding definitions of the HAL-MLEs indexed by sectional variation norm bounds $C = (C^u, C^l)$. This theorem can be applied to the bootstrap distribution at a data adaptive $C_n = (C_n^u, C^l)$ such as one from cross-validation.*

Bootstrap HAL-TMLE using optimal tuning for inference

The nuisance parameter estimates Q_n and G_n are key inputs of the HAL-TMLE bootstrap. The HAL estimations of these nuisance parameters depend largely on the selection of the upper bound of the sectional variation norm $C^u = (C_1^u, C_2^u)$ (C_1^u for Q_n and C_2^u for G_n). We will focus on a data adaptive selector of C_{1n}^u (replacing C_1^u), for a given selector C_{2n}^u, where the latter is chosen to be the cross-validation selector. Since our target parameter is a function of Q only, we suggest that the selection of C_{1n}^u is fundamentally more important than C_{2n}^u, and also creates enough room for our desired finite sample adjustment of the nonparametric bootstrap. In the software implementation of LASSO, the L_1-norm constraint C_1^u is translated into a penalized empirical risk with L_1-penalty hyper-parameter λ, where a choice of C_1^u corresponds with a unique choice λ. In the sequel, we will propose a selector of λ, and thereby of C_1^u.

Ideally, we want to set $C_1^u = C_{10}^u$ equal to the sectional variation norm of Q_0, so that the bootstrap model for the HAL-MLE $Q_n^\#$ is large enough for unbiased estimation of Q_n.

Due to the asymptotic equivalence of the cross-validation selector $C^u_{1n,CV}$ with the oracle selector that optimizes the loss-based dissimilarity, the cross-validation selector $C^u_{1n,CV}$ will approximate C^u_{10} as sample size increases. However, in finite samples, when the true sectional variation norm C^u_{10} of Q_0 is high (λ_0 is small), the cross-validation selector $C^u_{1n,CV}$ will tend to be smaller than the oracle value C^u_{10} ($\lambda_{CV} > \lambda_0$), That is, $C^u_{1n,CV}$ optimally trades off bias and variance for estimation of Q_0, but fixing C^u_1 at this choice $C^u_{1n,CV}$ might oversimplify the complexity of the target Q^*_n of the bootstrap distribution, and thereby causes the bootstrap to under-estimate the variability of the true sampling distribution of the TMLE. As a result, the bootstrap confidence interval will potentially still be anti-conservative.

Since the oracle choice λ_0 is unknown, we propose to estimate λ_0 with a plateau selection method. Consider a pre-specified ordered (from large to small) sequence of lambda candidates $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_J)$ with corresponding HAL-MLEs Q_{n,λ_j} and HAL-TMLEs Q^*_{n,λ_j}, $j = 1, \ldots, J$. We set $\lambda_1 = \lambda_{n,CV}$ so that we only consider sectional variation norm constraints larger than the cross-validation selector $C^u_{1n,CV}$. The sectional variation norm of Q_{n,λ_j} will thus be increasing in j. For each λ_j we compute the width $w_j = (q^{\#}_{0.975,n,\lambda_j} - q^{\#}_{0.025,n,\lambda_j})\sigma_n$ of the nonparametric bootstrap confidence interval based on bootstrapping the standarized TMLE $n^{1/2}(\Psi(Q^*_{n,\lambda_j}) - \Psi(Q_0))/\sigma_n$, given by $[\Psi(Q^*_n) + q^{\#}_{0.025,n,\lambda_j}\sigma_n, \Psi(Q^*_n) + q^{\#}_{0.975,n,\lambda_j}\sigma_n]$, $j = 1, \ldots, J$. The interval widths monotonically increase and should generally show de-acceleration around λ_0 where it will move towards a plateau, and, eventually it might becoming erratic. Through numerical simulations, we indeed observed that λ_0 is close to where the forming of the plateau begins. This method for selecting a tuning parameter was proposed in another context in [9]. It remains to decide on a method for determining the location of the start of the de-acceleration. A variety of methods could be proposed here. In our concrete implementation demonstrated in our simulation study, we compute the location of the start of the plateau as the location at which the second derivative is maximized, where we use the log λ-scale (due to λ having very small values). Specifically, $\lambda_{plateau} \triangleq \lambda_j$, where

$$j = \arg \max_{j=2,\ldots,J-1} \frac{(w_{j+1} - w_j) - (w_j - w_{j-1})}{(\log(\lambda_{j+1}) - \log(\lambda_j))(\log(\lambda_j) - \log(\lambda_{j-1}))}$$

Figure 2.1 illustrates a simulated example of the curve $\log(\lambda) \to w(\lambda)$. As the value of λ decreases starting at λ_{CV}, we observe a slow increase initially (almost a flat area around λ_{CV}), then an accelerated increase, till it starts reaching its plateau right after λ_0. Our method looks for the reflection point, where the function starts moving towards the plateau. Another method might be to look for the actual start of the plateau, but our concern is that this might corresponds with a plateau due to pure overfitting the data (where the finite sample only allows so much overfitting).

Increasing the scaling σ_n-factor by taking into account bias of bootstrap sampling distribution

Another modification we propose concerns the bias of the bootstrap distribution. We assume that we used the above method for selecting a $\lambda_n = \lambda_{plateau}$. We will use as point estimate

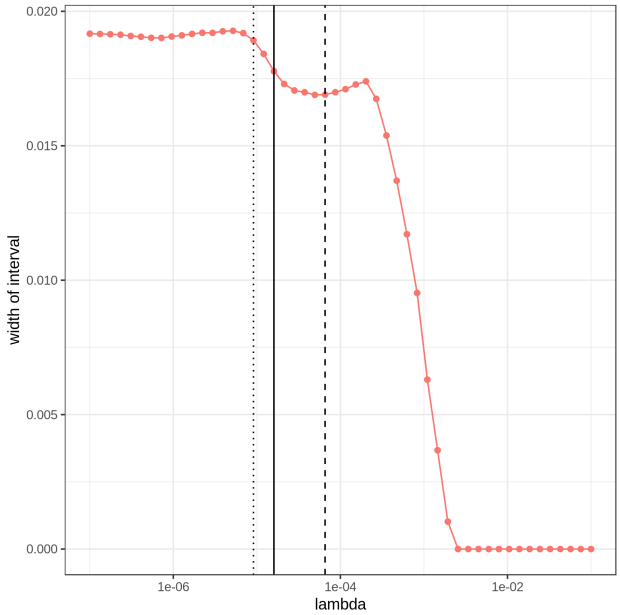

Figure 2.1: A simulated example of Wald-type interval width as a function of λ. Dotted line indicate λ_0, dashed line indicate λ_{CV} and solid line indicate $\lambda_{plateau}$

$\Psi(Q_n^*)$, where $Q_n^* = Q_{n,\lambda_{n,CV}}^*$, i.e, the TMLE using the cross-validated HAL-MLE. So the role of the bootstrap is to determine a confidence interval around this point estimate. Our confidence interval will be of the form $[\Psi(Q_n^*) + q_{n,0.025}^{\#}\sigma_n^{\#}/n^{1/2}, \Psi(Q_n^*) + q_{n,0.975}^{\#}\sigma_n^{\#}/n^{1/2}]$, where we use the nonparametric bootstrap at fixed sectional variation norm implied by λ_n, but centered to have mean zero, to obtain these two quantiles. The bias in the bootstrap distribution will instead be incorporated in $\sigma_n^{\#}$ by defining $\sigma_n^{\#2}$ as the MSE of the bootstrap realizations $\Psi(Q_{n,i}^{\#*})$ relative to $\Psi(Q_n^*)$, $i = 1, \ldots, N$, where N is the number of bootstrap samples drawn from P_n.

The motivation is that in general the nonparametric bootstrap will also inherit bias of the sampling distribution of $n^{1/2}(\Psi(Q_n^*) - \Psi(Q_0))/\sigma_n$. For example, if there is finite sample bias of $\Psi(Q_n^*)$ that is hurting the coverage of a Wald-type confidence interval, the bootstrap distribution (i.e., its quantiles) will likely further bias in the same direction. We choose not to estimate the bias with the bootstrap and compensate the bootstrap distribution accordingly through shifting it, since estimates of bias are typically unreliable. Instead, we widen the bootstrap confidence interval by replacing the scaling factor σ_n by the square root of the MSE of $\Psi(Q_n^{\#*})$ w.r.t. $\Psi(Q_n^*)$. Specifically, the "RMSE-scaled bootstrap" takes the form

$$[\Psi(Q_n^*) + \sigma_n^{\#} q_{n,0.025}^{\#}/n^{1/2}, \Psi(Q_n^*) + \sigma_n^{\#} q_{n,0.975}^{\#}/n^{1/2}], \tag{2.5}$$

where (using short-hand notation)

$$\sigma_n^{\#} \triangleq \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Psi_{i,n}^{\#*} - \Psi(Q_n^*))^2} = \sqrt{\text{bias}(\Psi_{i,n}^{\#*})^2 + \text{stddev}(\Psi_{i,n}^{\#*})^2}$$

is the estimated RMSE of the bootstrap estimator $\Psi_{i,n}^{\#*} = \Psi(Q_{n,i}^{\#*})$, and $q_{n,\alpha}^{\#}$ is the α-quantile of the bootstrap distribution of standardized $Z_{i,n}^{\#} = n^{1/2}(\Psi_{i,n}^{\#*} - \frac{1}{N}\sum_{i=1}^{N}\Psi_{i,n}^{\#*})/\text{stddev}(\Psi_{i,n}^{\#*})$.

The full modified HAL-TMLE bootstrap procedure we propose in this article can be summarized in the following pseudo-algorithm:

Algorithm 3: modified HAL-TMLE bootstrap procedure

pre-specify a grid of λ values, Λ;
for $\lambda \in \Lambda$ **do**
  fit HAL-MLE Q_n using tuning parameter λ;
  perform HAL-TMLE and record point TMLE $\Psi_n^*(\lambda)$;
end
perform cross-validation to select λ_{CV}; record the HAL-TMLE point estimate $\Psi(Q_n^*)$ with $Q_n^* = Q_{n,\lambda_{CV}}^*$;
Compute the plateau selector $\lambda_{plateau}$ among $\lambda \leq \lambda_{CV}$ based on running the nonparametric bootstrap for $n^{1/2}(\Psi(Q_{n,\lambda}^{\#*}) - \Psi(Q_{n,\lambda}^*))/\sigma_{n,\lambda}^{\#}$;
Set $\lambda = \lambda_{plateau}$, perform HAL-TMLE bootstrap N times to obtain quantiles $q_{n,0.025}^{\#}, q_{n,0.975}^{\#}$ of $n^{1/2}(\Psi(Q_{n,\lambda}^{\#*}) - E_{P_n}\Psi(Q_{n,\lambda}^{\#*}))/\sigma_{n,\lambda}^{\#}$;
compute $\sigma_n^{\#} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Psi(Q_{i,n}^{\#*}) - \Psi(Q_n^*))^2}$;
report $\Psi(Q_n^*)$ as the final point estimator; report the 95% confidence interval of the target parameter as $[\Psi(Q_n^*) + \sigma_n^{\#}q_{n,0.025}^{\#}/n^{1/2}, \Psi(Q_n^*) + \sigma_n^{\#}q_{n,0.975}^{\#}/n^{1/2}]$.

2.3 Examples

Average treatment effect

We have illustrated all components required for performing HAL-TMLE bootstrap on average treatment effect parameter alongside the general method description in Section 2.2. In HAL-TMLE, we update the part of the likelihood that is relevant to the target parameter Ψ, which are Q and Q_W. The estimation routine including the tuning parameter search is implemented in the 'ateBootstrap' function in the open-source R package "TMLEbootstrap" [7].

Average density value

One example of a non-doubly robust target parameter is the average density value parameter, whose second-order remainder takes the form of a quadratic function bounded away from

zero. Let $O \in \mathbb{R}^d$ be a multivariate random variable with probability distribution P_0. Let $\mathcal{M}$ be a nonparametric model dominated by Lebesgue measure μ, where we assume that for each $P \in \mathcal{M}$ its density $p = dP/d\mu$ is bounded away from below by 0 and from above by M. The average density value parameter is defined as $\Psi(P) = \mathbb{E}_P p(O) = \int p^2(o)d\mu(o)$. This target parameter is pathwise differentiable at P with canonical gradient $D^*(P)(O) = 2(p(O) - \Psi(P))$. The second-order remainder $R_2 = -\int (p - p_0)^2 d\mu$.

The HAL-TMLE consists of a HAL density learner, combined with a targeting step through the universal least favorable submodel. The HAL density learner first transforms the density estimation task to a longitudinal data format, thus transforming the density estimation task into a classification task. Predicting the probability of the classification surrogate retrieves the HAL density estimator. The software implementations can be found in the 'cv_densityHAL' function in the open-source R package "TMLEbootstrap" [7]. We update the initial density estimator (HAL-MLE) p_n by constructing the universal least favorable submodel $p_{n,\varepsilon}$

$$p_{n,\varepsilon} = p_n \exp(\int_0^\varepsilon D^*(p_{n,x})dx). \tag{2.6}$$

The HAL-TMLE is defined by the $p_n^* \triangleq p_{n,\varepsilon_n}$, where ε_n is the optimal ε along the submodel. Finding the optimal ε_n involves either moving infinitesimal steps along the integration formation (2.6), or recursively applying local least favorable submodels

$$p_{n,\varepsilon+d\varepsilon} = p^{lfm}_{n,\varepsilon,d\varepsilon}, \tag{2.7}$$

where $p^{lfm}_{n,\varepsilon,d\varepsilon} \triangleq (1 + d\varepsilon D^*(p_{n,\varepsilon}))p_{n,\varepsilon}$. The two approaches are mathematically equivalent and will both solve the influence curve equations. We direct readers to [31] for technical details. Once p_n^* is obtained, the HAL-TMLE of the average density parameter is the plug-in estimator $\Psi_n^* \triangleq \Psi(p_n^*) \triangleq \int {p_n^*}^2 d\mu$.

Blip variance

Under the same data generating distribution and notations as in Section 2.3, we can also estimate the blip variance parameter using HAL-TMLE [36]. Define the blip function at $P = (Q, G)$ as $B_P = \mathbb{E}_P[Y|A = 1, W] - \mathbb{E}_P[Y|A = 0, W]$, the blip variance parameter is defined as $\Psi(P) = \mathbb{E}_P(B_P^2) - (\mathbb{E}_P B_P)^2$. The canonical gradient of blip variance Ψ at P is

$$D^*(P) = D_1^*(P) + D_2^*(P) \tag{2.8}$$

$$= \underbrace{2\,(B_P(W) - \mathbb{E}_P B_P)\frac{2A-1}{G(A|W)}(Y - Q(A,W))}_{D_1^*(P)} + \underbrace{(B_P(W) - \mathbb{E}_P B_P)^2 - \Psi(P)}_{D_2^*(P)} \tag{2.9}$$

The second-order remainder is given by:

$$\begin{aligned} R_2(P, P_0) &= \Psi(P) - \Psi(P_0) + P_0 D^*(P) \\ &= (\mathbb{E}_0 B_0(W) - \mathbb{E}B(W))^2 - \mathbb{E}_0(B_0(W) - B(W))^2 \\ &+ \mathbb{E}_0[2(B(W) - \mathbb{E}B(W))(\frac{(G - G_0)(1|W)}{G(1|W)}(Q - Q_0)(1, W) - \frac{(G - G_0)(0|W)}{G(0|W)}(Q - Q_0)(0, W))] \end{aligned}$$

Blip variance is known to have a large non-zero second-order remainder term, which adds to the difficulty of finite sample inference.

The HAL-TMLE consists of a HAL initial estimator of the Q_0 and G_0 nuisance functions, followed by an iterative TMLE (local least favorable submodel). The HAL-MLE for Q_0 and G_0 are done in the same steps as in Section 2.3. The targeting step updates along the local least favorable submodel as

$$\text{logit}Q_{n,\varepsilon} = \text{logit}Q_n - C_2(G_n, Q_n), \tag{2.10}$$

where $C_2(G_n, Q_n) = 2(B_{P_n}(W) - P_n B_{P_n}(W))\frac{2A-1}{G_n(A|W)} P_n D_2^*(P_n)/||P_n D_2^*(P_n)||_2$, where $D_2^*(P_n)$ is the canonical gradient (2.9) evaluated at the empirical distribution P_n. The targeting requires recursion of the submodel (2.10) until convergence. Each step the $Q_{n,\varepsilon}$ is updated using Q_{n,ε_n} from the last iteration, and the stopping criteria is that $P_n D_2^*(P_{n,\varepsilon}) = 0$. Once converge, the HAL-TMLE of Ψ is just the plug-in estimator $\Psi_n^* \triangleq \Psi(P_n^*)$. There are other variants of the TMLE of blip variance parameter, such as using a universal least favorable submodel [31], using cross-validated TMLE [50]. These methods can all fit into the bootstrap framework we propose and we direct readers to [36] for details. The software implementations can be found in the 'blipVarianceBootstrap' function in the "TMLEbootstrap" package [7].

2.4 Simulations

Average treatment effect

To illustrate the finite sample performance of the proposed bootstrap method, we simulate a continuous outcome Y, a binary treatment A, and a continuous covariate W that confounds Y and A. The random variables are drawn from a family of distributions indexed by a_1, which characterizes the conditional distribution of Y, given A and W. The distribution of variables are as follows: $W \sim N(0, 4^2, -10, 10)$ is drawn i.i.d. from a truncated normal distribution with mean equals 0, standard deviation 4, bounded within $[-10, 10]$. $A \sim \text{Bernoulli}(p(W))$ is a Bernoulli binary random variable, with a probability $p(W)$ as a function of W, given by

$$p(W) = \begin{cases} 0.3, & \text{if } 0.3 + 0.1W\sin(0.1W) + \varepsilon_1 < 0.3 \\ 0.7, & \text{if } 0.3 + 0.1W\sin(0.1W) + \varepsilon_1 > 0.7 \\ 0.3 + 0.1W\sin(0.1W) + \varepsilon_1, & \text{otherwise} \end{cases}$$

where $\varepsilon_1 \sim N(0, 0.05^2)$. $Y = 3\sin(a_1 W) + A + \varepsilon_2$ is a sinusoidal function of W, where $\varepsilon_2 \sim N(0, 1)$. a_1 controls the amplitude of the sinusoidal function. It can be shown that

increasing a_1 (frequency) of the sin function increases the sectional variation norm (if G_0 is fixed, which is our setting). The value of the parameter of interest, ATE ψ_0, is 1. The experiment is repeated 500 times.

To analyze the above simulated data, we computed coverages and widths of (i) the Wald-type confidence interval where the nuisance functions (Q_0, G_0) are estimated using HAL(λ_{CV}) and (ii) bootstrap confidence interval discussed in Section 2.2 where HAL($\lambda_{plateau}$) is used for nuisance function estimators. Method (i) reflects common practice in making TMLE inference. We used correctly specified HAL regression and classification models to ease computations and focus ideas, but in practice, we suggest using a SuperLearner [24] which include HAL as part of the learner library to achieve the best out-of-sample generalizability. Results under samples sizes 500 and 1000 are shown in Figure 2.3.

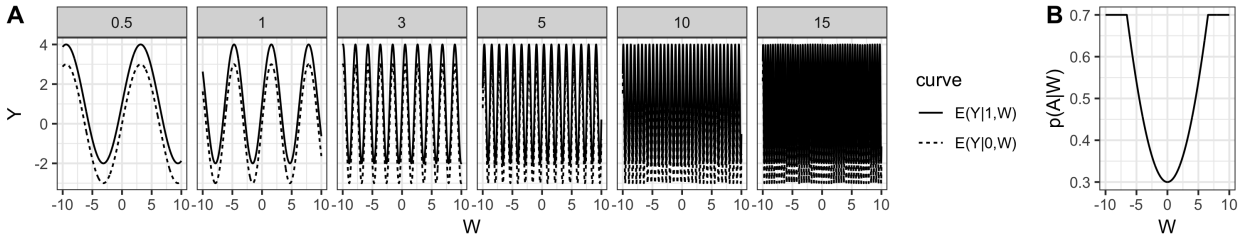

Figure 2.2: (A) True conditional outcome functions $E(Y|A=1,W)$ and $E(Y|A=0,W)$ at $a_1 = 0.5, 1, 3, 5, 10, 15$ and (B) true propensity score function

The simulations results reflect what is expected based on theory. In particular, since the sectional variation norm of the Q_0 function is large (relative to sample size), HAL regression fit in the finite sample is not ideal, which leads to a below than nominal coverage of Wald-type interval. Bootstrap intervals pick up the second-order remainder, and the coverage is very close to nominal and is robust to increasing sectional variation norm (a_1). The results for sample size 500 confirm our asymptotic analysis of the methods, with Wald-type coverage improving and two methods eventually converging to nominal.

Average density value

It is known that the average density value parameter has non-zero second-order remainder term after Taylor expansion. To illustrate our proposed method and explore finite-sample performance, we simulate a family of univariate densities with increasing sectional variation norm.

$$f(x;\theta_K) = \frac{1}{K}\sum_{k=1}^{K} g(x;\mu_k,\sigma_K),$$

where

$$g(x;\mu_k,\sigma_K) = \frac{1}{\sqrt{2\pi}\sigma_K}\exp[-\frac{1}{2}(x-\mu_k)^2/\sigma_K^2].$$

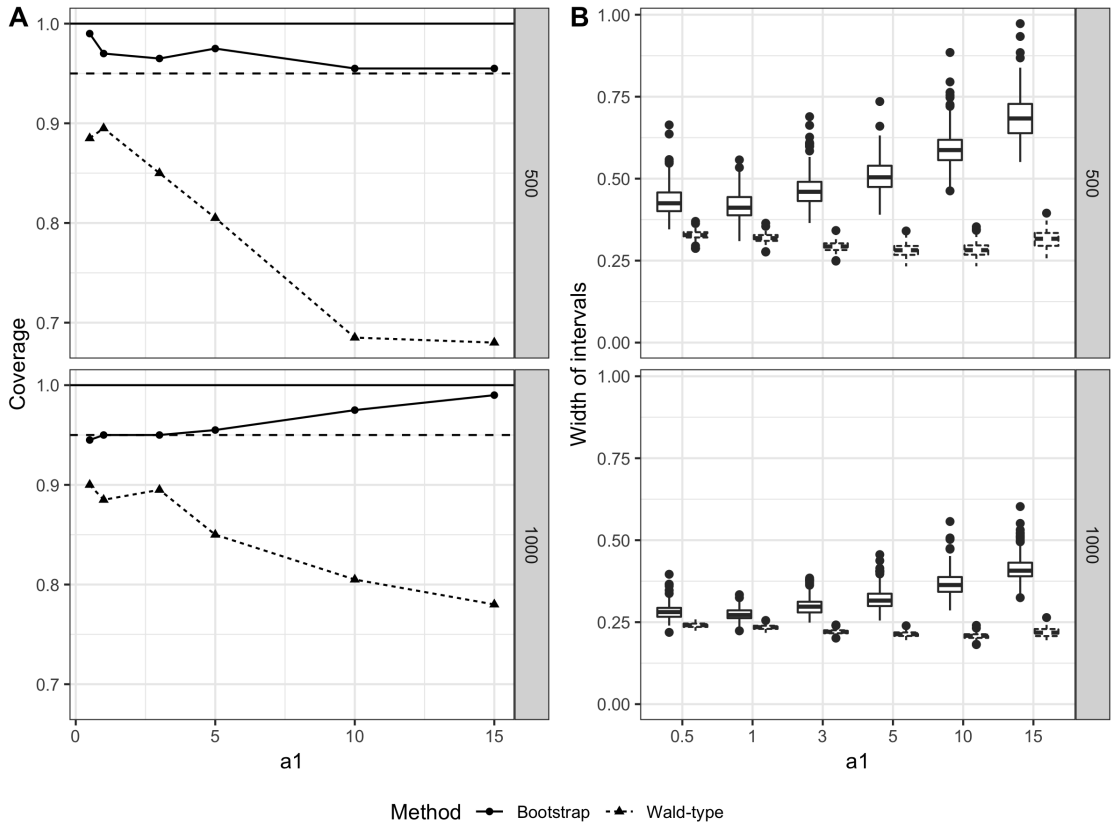

Figure 2.3: Results for ATE parameter comparing our bootstrap method and classic Wald-type method as a function of the a_1 coefficient (sectional variation norm) of the Q_0 function. Panel A is the coverage of the intervals, where dashed line indicate 95% nominal coverage. Panel B is the widths of the intervals. Within each panel, the upper plot is under sample size 500 and the lower plot is under sample size 1000.

For a given K, $\mu_k, k = 1, ..., K$ are equi-distantly placed in interval $[-4, 4]$. $\sigma_K = 10/K/6$. The true sectional variation norm of the density increases roughly linearly with K, that is $\|f_K\| = K\|f_1\|, K = 1, ..., 13$. Examples of the density family for K values used in the simulation are shown in Figure 2.4. We simulate from univariate densities for the sake of presentation and we expect the results under high dimensional densities to hold true, as the sectional variation norm can increase more rapidly with increasing number of dimensions. The estimand ψ_0 is not too variant with K. The experiment is repeated 500 times. Similar to the analysis for ATE, we compute interval coverages and widths for (i) Wald-type (λ_{CV}) and (ii) HAL-TMLE bootstrap ($\lambda_{plateau}$)

The simulations reflect what is expected based on theory: bootstrap methods control type-I error better than Wald-type confidence interval, uniformly across different sample sizes. In particular, when true sectional variation norm increases (with the number of modes

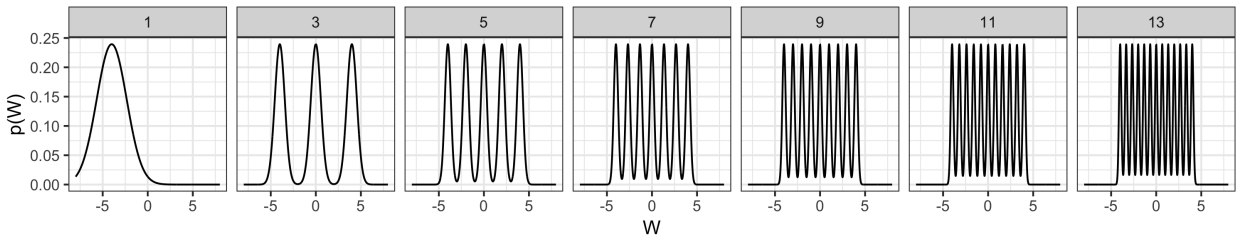

Figure 2.4: True probability density function $f(x;\theta_K)$ at $K = 1, 3, 5, 7, 9, 11, 13$

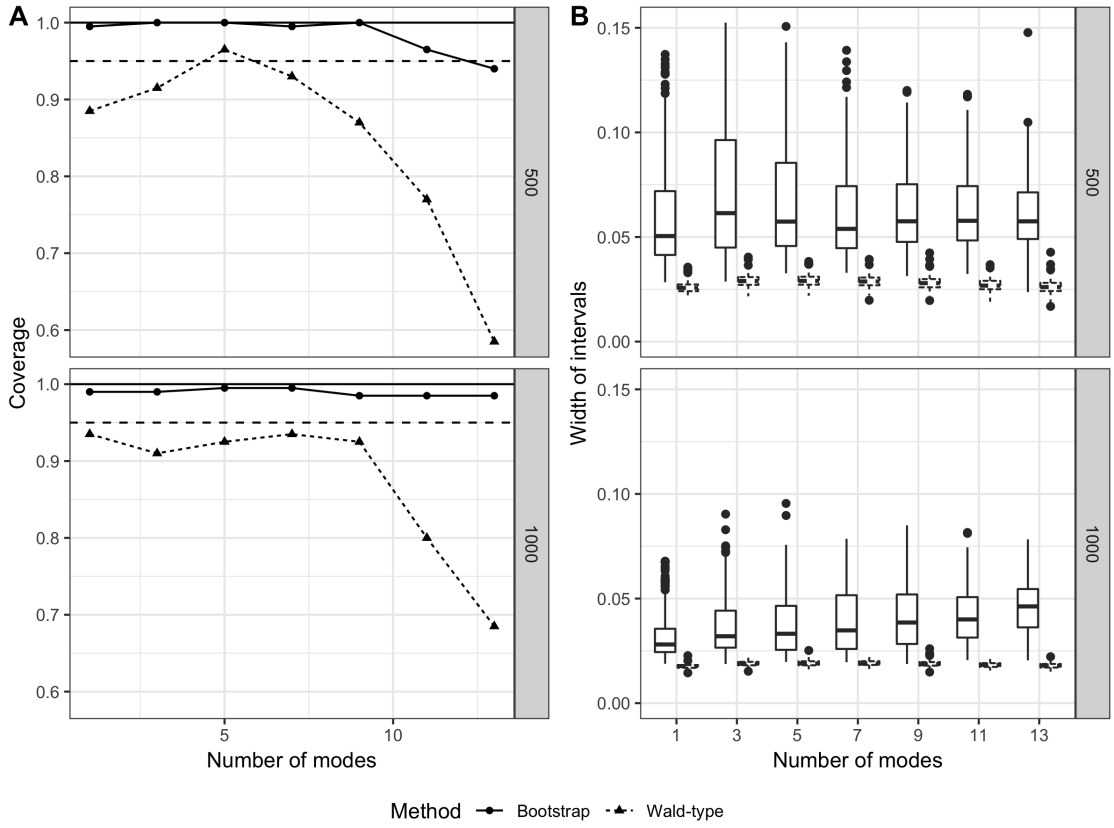

Figure 2.5: Results for average density value parameter comparing our bootstrap method and classic Wald-type method as a function of the number of modes in true density (sectional variation norm). Panel A is the coverage of the intervals, where dashed line indicate 95% nominal coverage. Panel B is the widths of the intervals. Within each panel, the upper plot is under sample size 500 and the lower plot is under sample size 1000.

in the density), non-zero second-order remainder term increases and Wald-type interval coverage declines. On the other hand, bootstrap can pick up the non-zero second-order remainder. Bootstrap confidence interval controls the coverage close to nominal rates and

is less sensitive to the true sectional variation norm of the density function. When sample size increases to 500, the Wald-type interval coverage increases, and in simple cases where the true sectional variation norm is small, Wald-type coverage can reach nominal. Bootstrap confidence interval keeps nominal and is robust to the true sectional variation norm.

Blip variance

We illustrate the proposed bootstrap method with one more non-doubly robust parameter, the blip variance. We simulate a continuous outcome Y, a binary treatment A, and a continuous covariate W that confounds Y and A. The random variables are drawn from a family of distributions indexed by J, which characterizes the blip function $f(W) \triangleq E(Y|A = 1, W) - E(Y|A = 0, W)$. The distribution of variables are as follows: $W \sim \text{Unif}(-4, 4)$ is drawn i.i.d from a uniform distribution between -4 and 4. $A \sim \text{Bernoulli}(0.5)$ is a Bernoulli random variable. $Y \sim N(A * f(W), 0.1^2)$ is continuous with conditional mean $A * f(W)$, so that the blip function $f(W) = 2 \sum_{j=1}^{J} \frac{1}{\sqrt{2\pi}\sigma_J} \exp\{-\frac{1}{2\sigma_J^2}(x - \mu_j)^2\}$, which is mixture of gaussian density function with J (non-overlapping) modes. Each mode is a normal density function. The μ_j are chosen equi-distantly in the interval $[-2, 2]$, the spread of the bell-shaped curve $\sigma_J = 10/J/8$ is chosen so that the modes are not overlapping each other. We multiply the density function by 2, so that the modes are not overwhelmed by Gaussian noise added to Y. Examples of true blip function $f(W)$ with different J used in the simulation are shown in figure 2.6. The value of the true blip variance ψ_0 is not too variant with J. The experiment is repeated 500 times. We compute interval coverages and widths for Wald-type (using λ_{CV}) and HAL-TMLE bootstrap (using $\lambda_{plateau}$)

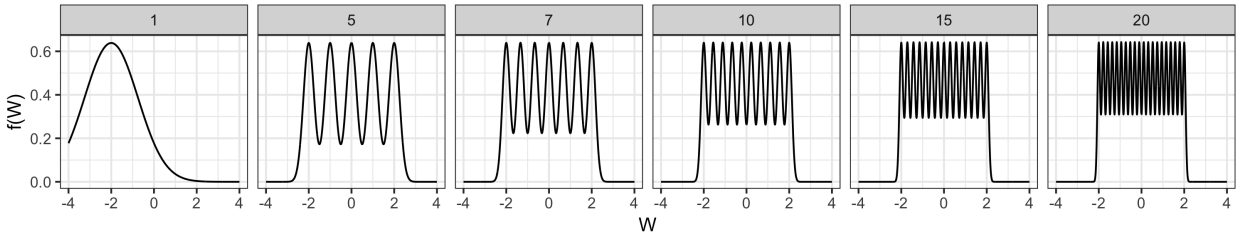

Figure 2.6: True conditional average treatment effect function $f(W)$ at $J = 1, 2, 5, 10, 20$

With the non-forgiving second-order remainder term of blip variance, we expect both bootstrap and Wald-type interval to perform poorly at a limited sample size 100. Under a data generating distribution with a small sectional variation norm (number of modes), bootstrap can achieve near nominal coverage. When the blip function becomes more complex, the coverages of both methods start to decline, although bootstrap always has better coverage than Wald-type. At sample size 500, more asymptotic kicks in and the bootstrap coverage

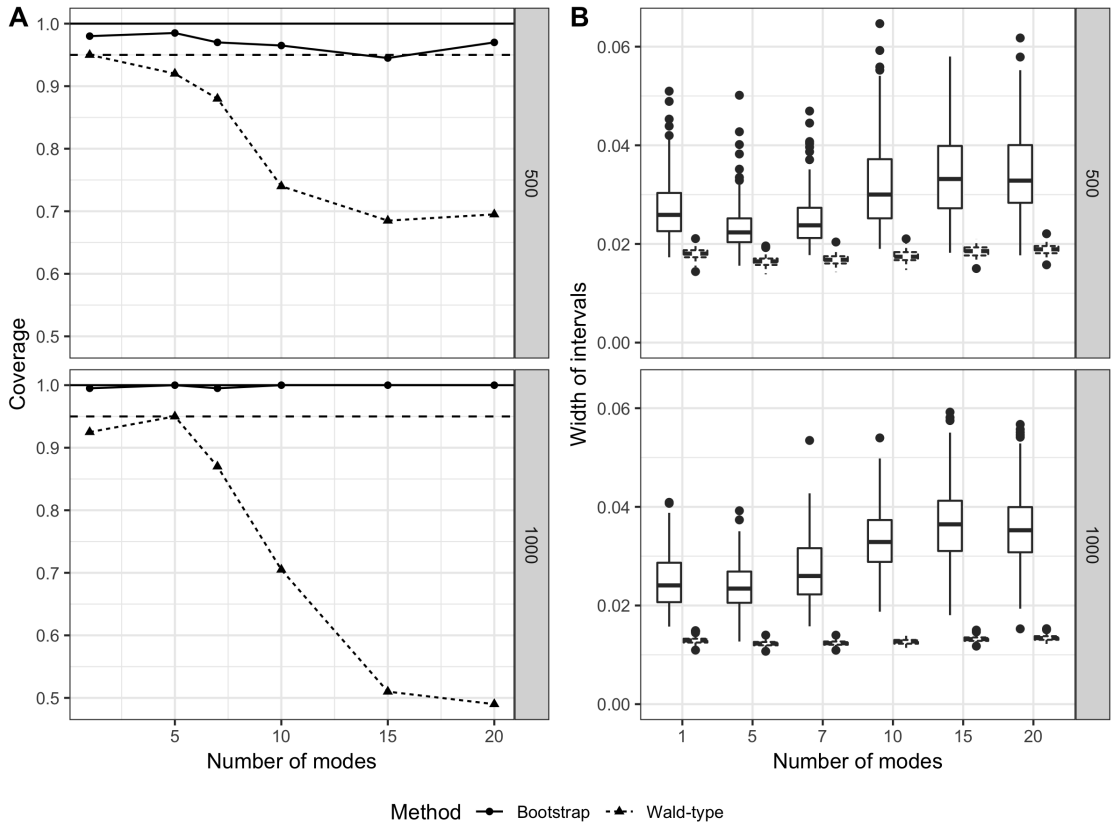

Figure 2.7: Results for blip variance parameter comparing our bootstrap method and classic Wald-type method as a function of the number of modes in true blip function $f(W)$ (sectional variation norm). Panel A is the coverage of the intervals, where dashed line indicate 95% nominal coverage. Panel B is the widths of the intervals. Within each panel, the upper plot is under sample size 500 and the lower plot is under sample size 1000.

quickly achieves nominal, while Wald-type fails to have nominal coverage at a larger number of modes.

2.5 Application

We evaluate the finite sample coverage of the proposed bootstrap method on two publicly available datasets: the UCI salary data and the MNIST image data.

UCI salary data

The UCI salary data [47] collects the annual salaries of 848 professors from the University of California at Irvine in 2007 as well as their demographic and work information. We study

the ATE of gender on salary, controlling for their education history, ethnicity, and years of university service. Although we know gender cannot be intervened and thus the parameter is not an interesting causal parameter, we treat ATE as a statistical parameter to illustrate our bootstrap method. We use HAL regression to fit the outcome regression and use HAL logistic regression to fit the propensity score, both controlling for all pre-treatment variables. We apply the modified HAL-TMLE bootstrap in Section 2.2.

We evaluate the coverage of the bootstrap interval and that of the Wald-type interval from HAL-TMLE on subsamples of the dataset with different sizes. Since we do not know the actual parameter value, we reserve a large and separate test set of the data, apply HAL-TMLE on the test set, and treat the estimated parameter on the test set as the ground truth of the parameter value. We also compute the width of the two intervals for reference. We repeat the procedure by subsampling the training set into sizes of 50, 100, 200 and repeat each setting 1000 times to compute coverage of intervals.

Results are displayed in Figure 2.8. The results indicate that the bootstrap interval maintains the nominal coverage at different sizes of the training set, while the Wald-type interval under-covers across all sample sizes. The widths of the bootstrap intervals are slightly wider than those of the Wald-type intervals.

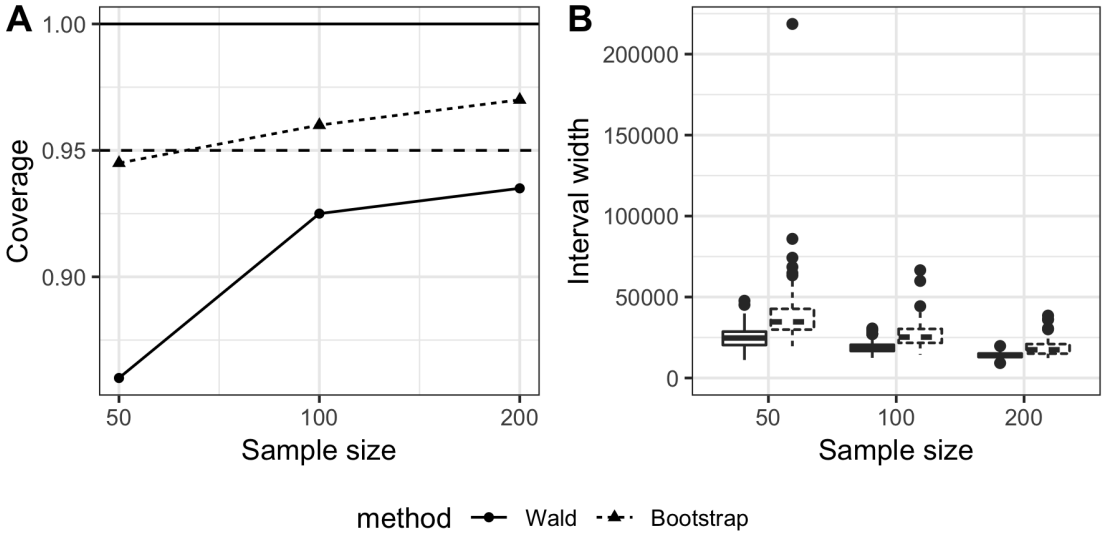

Figure 2.8: Results for UCI salary dataset comparing our bootstrap method and classic Wald-type method as a function of the subsample size. Plot A is the coverage of the intervals, where dashed line indicate 95% nominal coverage. Plot B is the widths of the intervals.

MNIST image summary statistics

The MNIST data [35] consists of 60,000 images of handwritten digits. We transform each image into a real-valued scalar $X \in (0, 1)$ representing the proportion of the image pixels

which are occupied by writing. We treat these univariate features X drawn i.i.d. from a 1-dimensional population density $p(x)$, and we analyze the average density value from data.

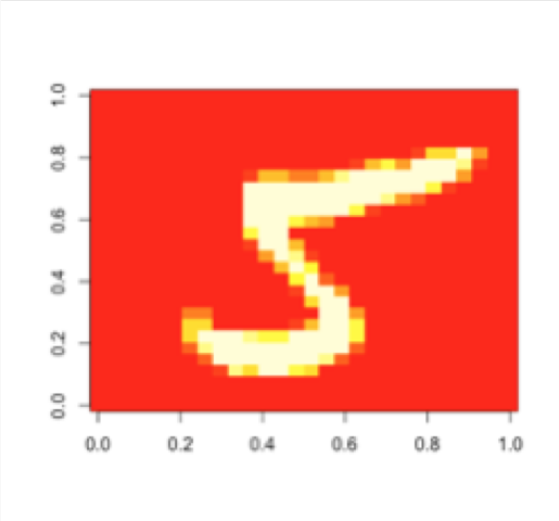

Figure 2.9: An example MNIST image of digit 5. By counting how many pixels in the image are covered by writing, the summary statistic of this image is roughly 10%

We use the HAL density learner as implemented in Section 2.3 to estimate the 1-dimensional density and perform targeting to construct the HAL-TMLE estimator. Similar to our procedure in the salary data analysis, we evaluate the coverage and width of our proposed bootstrap interval and compare with the Wald-type interval from HAL-TMLE. Similar to the previous data analysis, we hold out a large test set of size 10,000 and treat the HAL-TMLE applied on the massive test set as the ground truth parameter value in our evaluation. We repeat the procedure by subsampling the training set into different sizes and repeat each setting 1000 times to compute coverage of intervals.

Figure 2.10 depicts the results. The bootstrap interval coverage is better than Wald-type and closer to nominal 95% at all sample sizes. As sample size increases, bootstrap quickly becomes nominal, while Wald-type coverage converges to nominal more slowly. The widths of the bootstrap intervals are slightly wider than those of the Wald-type intervals.

2.6 Discussion

The article investigated bootstrap inference of HAL-TMLE. The finite-sample confidence interval is constructed by grabbing empirical quantiles of the bootstrap distribution. We proposed a 'plateau tracking' method to approximate the optimal tuning parameter of the highly-adaptive LASSO so that the bootstrap confidence interval coverage is optimized. In the case of non-doubly robust target parameter when the point estimate has finite sample

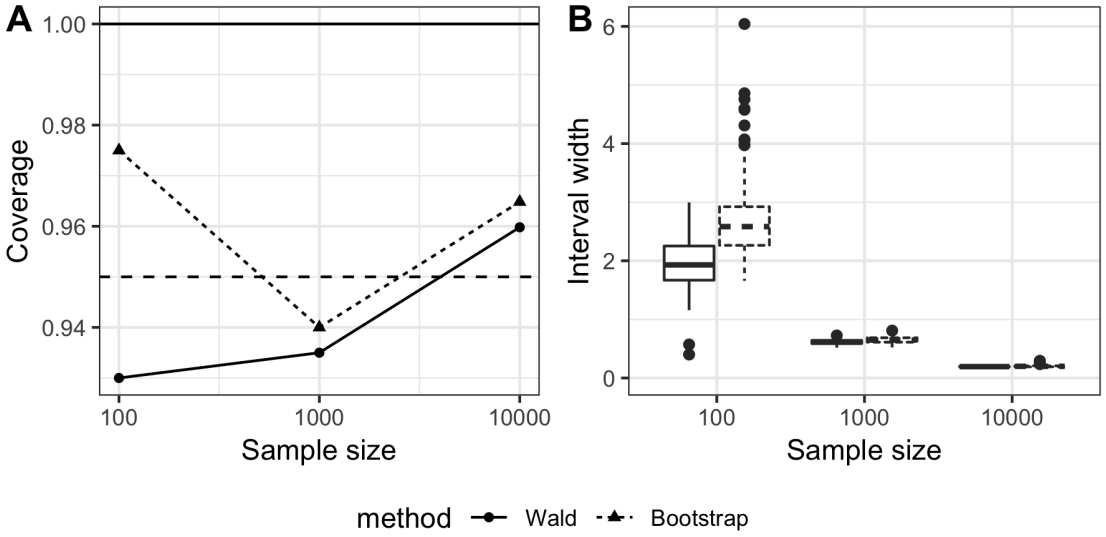

Figure 2.10: Results for MNIST dataset comparing our bootstrap method and classic Wald-type method as a function of the subsample size. Plot A is the coverage of the intervals, where dashed line indicate 95% nominal coverage. Plot B is the widths of the intervals.

bias, we proposed a scale adjustment post-processing method such that the bootstrap confidence intervals are at least as wide as the Wald-type interval that we would conventionally compute. We presented exact formulations and simulations for average treatment effect, average density value, and blip variance parameter and showed that our bootstrap confidence intervals have optimal coverage that is robust to sizeable sectional variation norm of the distribution function. We expect our work to be important for both randomized trials and observational studies and feel our finite-sample valid non-parametric bootstrap method fills an important gap in the machine learning theory literature.

There are some important future directions to this research. We will apply the method to explore effects in a real data application. It will also be useful to develop a theory that explains the connection between our 'plateau tracking' method and the tuning parameter selection method for non-pathwise differentiable parameter CV-TMLE [20], where the optimal tuning parameter is chosen at the point where the derivative of the estimator equals the derivative of the estimator standard error, because the two methods both look for tuning parameter based on the second moment of the parameter. We also wish to explore applying this method to other TMLE applications in addition to the three examples we present in this article, such as longitudinal TMLE studies or TMLE of high dimensional parameters where the second order remainder term can easily outnumber the first order term.

Chapter 3

Efficient Causal Inference Based on the Highly Adaptive Lasso: Undersmoothing and Targeted HAL

3.1 Introduction

Nonparametric structural causal models provide statistical models for the data generating distribution and allow the formal definition of causal impact of an intervention on an outcome of interest. Formal identification results establish non-testable assumptions that allows one to identify the causal quantity of interest as an estimand of the data distribution. Once we accept this estimand as a best or perfect approximation of the causal quantity of interest, we are left with a pure statistical estimation problem of learning the estimand based on knowing that the true data distribution falls in a specified infinite dimensional statistical model.

Semiparametric efficiency theory teaches us that the nuisance parameter regression need to be root-n consistent, combined with solving the additional critical efficient score equation, to make the statistical estimator efficient [46]. Seminal work by [1] proposed highly-adaptive LASSO (HAL) algorithm and showed it is a regression that will guarantee this rate in the asymptotic, if the true function is cadlag and has finite sectional variation norm. Once such rate is achieved, the second order remainder has root-n rate. The researcher can then perform estimating equation method or targeted maximum likelihood estimator (TMLE) to solve the efficient influence curve (EIC) equation (estimating equation), so that the final estimator is asymptotically linear (and efficient) [32].

In this article, we present two alternatives to solving the EIC equation by using HAL: targeted HAL and under-smoothed HAL. Targeted HAL augments an additional covariate in the design matrix for the regression and takes advantage of the linear regression property to solve the EIC equation. Under-smoothed HAL selects the L_1-norm of the vector of coefficients associated with the collection of 0-th order spline basis functions larger than the value from cross-validation. [29] showed sufficient conditions so that it will solve EIC

equation for any desired pathwise differentiable statistical parameter, while preserving root-n consistency from HAL. Our contribution is to propose an automated tuning method called "multi-task tuning" that satisfies these conditions, which uses a family of proxy tasks to tune the L_1-norm. We show that the methods perform better than plug-in HAL regression (HAL-MLE) and TMLE using HAL regression as initial estimator (HAL-TMLE) in scenarios that are do not favor EE and TMLE method [18].

Organization of the paper We set up the statistical estimation problem in section 3.2 and review the highly-adaptive LASSO in section 3.2. And then present targeted HAL in section 3.2 and under-smoothed HAL in section 3.2. In section 3.3, we use two simulations to illustrate the finite sample performance of the proposed methods.

3.2 Methodology

Statistical Estimation of Target Parameter

Suppose we observe $O_1, \ldots, O_n \sim_{iid} P_0 \in \mathcal{M}$, where O is a Euclidean random variable of dimension k with support $\mathcal{O}$ contained in $[0, \tau_o] \subset \mathbb{R}^k$. Let $Q : \mathcal{M} \to Q(\mathcal{M}) = \{Q(P) : P \in \mathcal{M}\}$ be a functional parameter of the data distribution. It is assumed that there exists a loss function $L(Q)$ so that $P_0 L(Q(P_0)) = \min_{P \in \mathcal{M}} P_0 L(Q(P))$, where we use the notation $Pf \equiv \int f(o) dP(o)$. Thus, $Q(P)$ can be defined as the minimizer of the risk function $Q \to PL(Q)$ over all Q in the parameter space. Let $d_0(Q, Q_0) \equiv P_0 L(Q) - P_0 L(Q_0)$ be the loss-based dissimilarity, which for most loss functions behaves as a square of an $L^2(P)$-type norm (e.g., Kullback-Leibler divergence for the log-likelihood loss). We assume that $M_{20} \equiv \sup_{P \in \mathcal{M}} P_0 \{L(Q(P)) - L(Q_0)\}^2 / d_0(Q(P), Q_0) < \infty$ and $M_1 \equiv \sup_{o \in \mathcal{O}, P \in \mathcal{M}} \mid L(Q(P))(o) \mid < \infty$. These latter two assumptions are sufficient to guarantee good theoretical behavior of cross-validation-based estimator selection. In particular, these assumptions provide conditions whereby the a cross-validation-selected estimator is asymptotically equivalent with an oracle selector [24].

Let $\Psi : \mathcal{M} \to \mathbb{R}$ represent the statistical target parameter of interest, so that $\Psi(P_0)$ is the estimand we aim to learn. We assume that Ψ is pathwise differentiable at $P \in \mathcal{M}$ in the sense that $\frac{d}{d\epsilon} \Psi(P_\epsilon)\big|_{\epsilon=0} = PD(P)S$ for a rich collection of submodels $\{P_\epsilon : \epsilon\}$ through P at $\epsilon = 0$ with score S. If the gradient $D(P)(O)$ is chosen to be a score itself (or an arbitrarily fine approximation of a score), then it is called the canonical gradient, which we denote by $D^*(P)$. As above, let $Q : \mathcal{M} \to Q(\mathcal{M}) = \{Q(P) : P \in \mathcal{M}\}$ be a functional parameter such that $\Psi(P) = \Psi_1(Q(P))$ for some Ψ_1: we will abuse notation, and simply use $\Psi(Q)$ and $\Psi(P)$ interchangeably. Let $G : \mathcal{M} \to \mathcal{G}$ be a functional nuisance parameter so that the canonical gradient $D^*(P)$ only depends on P through $(Q(P), G(P))$. Let $R_2(P, P_0) = \Psi(P) - \Psi(P_0) + P_0 D^*(P)$ be the exact second-order remainder for the target parameter expansion. This remainder $R_2(P, P_0)$ only involves differences between (Q, G) and (Q_0, G_0) so that we will use notation $D^*(P) = D^*(Q(P), G(P))$ and $R_2(P, P_0) = R_2(Q, G, Q_0, G_0)$.

Consider that for a plug-in estimator $\Psi(Q_n)$ of $\Psi(Q_0)$,

$$\Psi(Q_n) - \Psi(Q_0) = (P_n - P_0)D^*(Q_n, G_0) - P_n D^*(Q_n, G_0) + R_2(Q_n, G_0, Q_0, G_0).$$

Assuming that $\{D^*(Q, G) : Q, G\}$ falls in a class of cadlag functions with a universal bound on the sectional variation norm (which is, importantly, a Donsker class), using empirical process theory we can establish a simple $L^2(P_0)$-consistency $P_0\{D^*(Q_n, G_0) - D^*(Q_0, G_0)\}^2 \to_p 0$ implies $(P_n - P_0)D^*(Q_n, G_0) = (P_n - P_0)D^*(Q_0, G_0) + o_P(n^{-1/2})$ [48]. In addition, the above stated convergence $d_0(Q_n, Q_0) = o_P(n^{-1/2})$ will generally imply (under a strong positivity assumption) that $R_2(Q_n, G_0, Q_0, G_0) = o_P(n^{-1/2})$. In so-called double-robust causal inference or censored data problems the second-order remainder only involves cross-terms like $(Q_n - Q_0)(G_n - G_0)$ so that we even have $R_2(Q_n, G_0, Q_0, G_0) = 0$ [34]. Thus,

$$\Psi(Q_n) - \Psi(Q_0) = P_n D^*(Q_0, G_0) - P_n D^*(Q_n, G_0) + o_P(n^{-1/2}).$$

The only remaining obstacle in proving efficiency of the HAL-MLE is that we need $P_n D^*(Q_n, G_0) = o_P(n^{-1/2})$. We can show that this can be proven under two fundamental conditions: 1) the loss function $L(Q)$ must generate the canonical gradient as a score; 2) C_n must be selected "large enough". We discuss these two conditions in Section 3.2.

Canonical gradient of target parameter in tangent space of loss function: We assume that the loss function $L(Q)$ is such that there exists a class of submodels $\{Q_\epsilon^h : \epsilon\} \subset Q(\mathcal{M})$, indexed by a choice h,through Q at $\epsilon = 0$, so that for any $G \in \mathcal{G}$, one of these h-specific submodels generates a score that equals the canonical gradient $D^*(Q, G)$ at (Q, G):

$$\frac{d}{d\epsilon} L(Q_\epsilon^h)\Big|_{\epsilon=0} = D^*(Q, G).$$

Since the canonical gradient is an element of the tangent space and thereby typically a score of a submodel, this generally holds for Q defined as the density of P and the log-likelihood loss $L(Q) = -\log Q$. However, for any Q so that $\Psi(P)$ depends on P only through Q there are typically more direct loss functions $L(Q)$, so that the loss-based dissimilarity $d_0(Q, Q_0) = P_0 L(Q) - P_0 L(Q_0)$ directly measures a dissimilarity between Q and Q_0, for which this condition holds as well.

HAL-MLE

Parameter space for functional parameter Q: Cadlag and uniform bound on sectional variation norm. We assume that the parameter space $Q(\mathcal{M}) = \{Q(P) : P \in \mathcal{M}\}$ is a collection of multivariate real-valued cadlag functions on a cube $[0, \tau] \subset \mathbb{R}^k$ with finite sectional variation norm $\|Q(P)\|_v^* < C^u$ for some $C^u < \infty$ [13, 33, 28]: i.e., for all P, $Q(P)$ is a k-variate real-valued cadlag function on $[0, \tau] \subset \mathbb{R}^k_{\geq 0}$ with $\|Q(P)\|_v^* < C^u$, where the sectional variation norm is defined by

$$\|Q\|_v^* \equiv Q(0) + \sum_{s \subset \{1,\ldots,k\}} \int_{[0_s, \tau_s]} | \, dQ_s(u_s) \, | \, .$$

For a given subset $s \subset \{1, \ldots, k\}$, $Q_s : (0_s, \tau_s] \to \mathbb{R}$ is defined by $Q_s(x_s) = Q(x_s, 0_{-s})$. That is, Q_s is the s-specific section of Q which sets the coordinates in the complement of subset $s \subset \{1, \ldots, k\}$ equal to 0. For a given vector $x \in [0, \tau]$, we define $x_s = (x(j) : j \in s)$. Sometimes, we will also use the notation $x(s)$ for x_s.

Note also that $[0, \tau] = \{0\} \cup (\cup_s (0_s, \tau_s])$ is partitioned in the singleton $\{0\}$, the s-specific left-edges $(0_s, \tau_s] \times \{0_{-s}\}$ of cube $[0, \tau]$, and, in particular, the full-dimensional inner set $(0, \tau]$ (corresponding with $s = \{1, \ldots, k\}$). Therefore, the above sectional variation norm equals the sum over all subsets s of the variation norm of the s-specific section over its s-specific edge. It is also important to note that any cadlag function Q with finite sectional variation norm can be represented as

$$Q(x) = Q(0) + \sum_{s \subset \{1,\ldots,k\}} \int_{(0_s, x_s]} dQ_s(u_s).$$

That is, $Q(x)$ is a sum of integrals up to x_s over the s-specific edges with respect to the measure generated by the corresponding s-specific section Q_s. Thus, we refer to Q_s both as a cadlag function and as a measure. We note that this representation represents Q as an infinitesimal linear combination of indicator basis functions $x \to \phi_{s,u_s}(x) \equiv I(x_s \geq u_s)$ indexed by knot-point u_s with coefficient $dQ_s(u_s)$:

$$Q(x) = Q(0) + \sum_{s \subset \{1,\ldots,k\}} \int \phi_{s,u_s}(x) dQ_s(u_s).$$

Note that the L_1-norm of the coefficients in this representation is precisely the sectional variation norm $\|Q\|_v^*$.

Let $\mathcal{Q}(C^u) = \{Q \in D[0, \tau] : \|Q\|_v^* < C^u\}$ be the class of cadlag functions with sectional variation norm bounded by C^u, which is thus the parameter space for Q. Let $C_0 \equiv \|Q_0\|_v^*$ be the sectional variation norm of the true Q_0, and let C^u be an upper bound guaranteeing that $C_0 < C^u$. For a data adaptive selector C_n, we define the HAL-MLE as

$$Q_n \equiv \arg \min_{Q \in \mathcal{Q}(C_n)} P_n L(Q). \tag{3.1}$$

We will restrict the minimization to Q for which for all subsets $s \subset \{1, \ldots, k\}$, $dQ_s(u_s)$ is a discrete measure with a finite support $\{z_{s,j} : j = 1, \ldots, n_s\}$, where this support is chosen fine enough so that its resulting bias is negligible. Typically, one can actually prove that the unrestricted HAL-MLE (3.1) is attained at a discrete Q_n. Generally, if O includes observing X where $L(Q)(O)$ depends on Q through $Q(X)$, we recommend to select the support of dQ_s as a subset (or whole set) of the observed data $X_i(s)$, $i = 1, \ldots, n$. The above representation for functions in $D[0, \tau]$ shows that all such discrete Q are represented by a finite dimensional linear combination of basis functions indexed subset s and knotpoint $z_{s,j}$. Therefore, in this case the HAL-MLE can be represented as $Q_n = \sum_{s,j \in \mathcal{J}_n(s)} \beta_n(s,j) \phi_{s,j}$, where

$$\beta_n \equiv \arg \min_{\beta, \|\beta\|_1 \leq C_n} L\left(\sum_{s,j \in \mathcal{J}_n(s)} \beta(s,j) \phi(s,j) \right),$$

and $\mathcal{J}_n(s)$ is the collection of support points of the s-specific section $Q_{n,s}$ of Q_n.

The data adaptive selector C_n defining the L_1-norm restriction will be selected larger or equal than the cross-validation selector

$$C_{n,cv} = \arg\min_C \frac{1}{V} \sum_{v=1}^{V} P_{n,v}^1 L(\hat{Q}_C(P_{n,v})) \ ,$$

where $P_{n,v}^1, P_{n,v}$ are the empirical distributions of the validation and training sample, respectively, corresponding with the v-th sample split in a typical V-fold cross-validation scheme. Here $\hat{Q}_C(P_{n,v})$ is the HAL-MLE applied to the training sample corresponding with the v-th sample split. For any selector $C_n \leq C^u < \infty$ for which $P(C_n > C_0) \to 1$, we have that $d_0(Q_n, Q_0) = o_P(n^{-1/2-\alpha(k)})$ for $\alpha(k) = 1/(2(k+2))$ [28]. In particular, we have this rate of convergence for the cross-validation selector, which is optimal for estimation of Q_0 as a whole.

HAL-TMLE for the Treatment-specific Mean

Consider a finite dimensional local least favorable model $\{Q_{n,\epsilon} : \epsilon\} \subset Q(\mathcal{M})$ through Q_n at $\epsilon = 0$ so that the linear span of the components of $\frac{d}{d\epsilon} L_1(Q_{n,\epsilon})$ at $\epsilon = 0$ includes $D^*(Q_n, G_n)$. Let $Q_n^* = Q_{n,\epsilon_n}$ for $\epsilon_n = \arg\min_\epsilon P_n L_1(Q_{n,\epsilon})$. We assume that this one-step TMLE Q_n^* already satisfies

$$r_n \equiv P_n D^*(Q_n^*, G_n) = o_P(n^{-1/2}). \tag{3.2}$$

Since $d_{01}(Q_n, Q_0) = o_P(n^{-1/2})$ we will have that $\epsilon_n = o_P(n^{-1/4})$, and ϵ_n solves its score equation $\frac{d}{d\epsilon_n} P_n L_1(Q_{n,\epsilon_n}) = 0$, which, in first order, equals its score equation $P_n D^*(Q_{n,\epsilon_n}, G_n)$ at $\epsilon = 0$ (with a second order remainder $O(\epsilon_n^2) = o_P(n^{-1/2})$). This basic argument allows one to prove that (3.2) holds under the assumption $d_{01}(Q_n, Q_0) = o_P(n^{-1/2})$ and regularity conditions, as formally shown in the Appendix of [28]. The HAL-TMLE of ψ_0 is the plug-in estimator $\psi_n^* = \Psi(Q_n^*)$.

Example: (Treatment-specific mean)

Let $O = (W, A, Y) \sim P_0$, where $Y \in \{0, 1\}$ and $A \in \{0, 1\}$ are binary random variables. Let (A, W) have support in $[0, \tau] \in \mathbb{R}^k$, where various of its components are discrete and thereby supported on a finite grid within $[0, \tau]$. Let $\bar{G}(W) = E_P(A \mid W)$ and $\bar{Q}(A, W) = E_P(Y \mid A, W)$. Assume the positivity assumption $\bar{G}_0(W) > \delta > 0$ for some $\delta > 0$; $\bar{Q}_0$ and $\bar{G}_0$ are cadlag functions with $\|\bar{Q}_0\|_v^* \leq C^u$ and $\|\bar{G}_0\|_v^* \leq C_2^u$ for some finite constants C^u, C_2^u; $\delta < \bar{Q}_0 < 1 - \delta$ for some $\delta > 0$. This defines the statistical model $\mathcal{M}$ for P_0.

Let $\Psi : \mathcal{M} \to \mathbb{R}$ be defined by $\Psi(P) = E_P E_P(Y \mid W, A = 1)$. For simplicity, we focus on estimation of this treatment specific mean, but the presentation trivially generalizes to the average treatment effect (ATE) $\Psi(P) = E_P E_P(Y \mid W, A = 1) - E_P E_P(Y \mid$

$A = 0, W$). Let $\tilde{Q} = (Q_W, \bar{Q})$, where Q_W is the probability distribution of W. Note that $\Psi(P) = \Psi(\tilde{Q}) = Q_W \bar{Q}(\cdot, 1)$. We have that Ψ is pathwise differentiable at P with canonical gradient given by $D^*(\tilde{Q}, G) = A/\bar{G}(W)(Y - \bar{Q}(A, W)) + \bar{Q}(1, W) - \Psi(\tilde{Q})$. Let $L(\bar{Q})(O) = -\{Y \log \bar{Q}(A, W) + (1 - Y) \log(1 - \bar{Q}(A, W))\}$ be the log-likelihood loss for $\bar{Q}$, and note that by the above bounding assumptions on $\bar{Q}$, we have that this loss function has finite bounds $M_1 < \infty$ and $M_{20} < \infty$. Let $D_1^*(\bar{Q}, \bar{G}) = A/\bar{G}(Y - \bar{Q})$ be the $\bar{Q}$-component of the canonical gradient, $D_2^*(\tilde{Q}) = \bar{Q}(1, W) - \Psi(Q)$ the Q_W-component, and note that $D^*(\tilde{Q}, G) = D_1^*(\bar{Q}, G) + D_2^*(\tilde{Q})$. We have $\Psi(\tilde{Q}) - \Psi(\tilde{Q}_0) = -P_0 D^*(\tilde{Q}, G) + R_{20}(\bar{Q}, \bar{G}, \bar{Q}_0, \bar{G}_0)$, where

$$R_2(\bar{Q}, \bar{G}, \bar{Q}_0, \bar{Q}_0) = P_0 \frac{\bar{G} - \bar{G}_0}{\bar{G}} (\bar{Q} - \bar{Q}_0).$$

We have $\sup_{P \in \mathcal{M}} \|D^*(\tilde{Q}(P), G(P))\|_v^* < C(C^u, C_2^u)$ for some finite constant C implied by the universal bounds (C^u, C_2^u) on the sectional variation norm of $\bar{Q}, \bar{G}$. We also note that, using Cauchy-Schwarz inequality, $R_{20}(\bar{Q}, \bar{G}, \bar{Q}_0, \bar{G}_0) \leq \frac{1}{\delta} \|\bar{Q} - \bar{Q}_0\|_{P_0} \|\bar{G} - \bar{G}_0\|_{P_0}$, where $\|f\|_{P_0}^2 = \int f^2(o) dP_0(o)$.

The least favorable submodel through $\bar{Q}_n$ is given by

$$\text{logit}\bar{Q}_{n,\varepsilon} = \text{logit}\bar{Q}_n + \varepsilon C(G_n), \tag{3.3}$$

where $C(G_n)(A, W) \triangleq A/G_n(W)$. Let $\varepsilon_n \triangleq \arg\min_\varepsilon P_n L_1(Q_{n,\varepsilon})$, which is thus computed with a simple univariate logistic regression MLE, using as off-set $\text{logit}\bar{Q}_n$. This defines the TMLE $\bar{Q}_n^* = \bar{Q}_{n,\epsilon_n}$. Recall that $Q_{W,n}$ is already an NPMLE so that a TMLE-update based on a log-likelihood loss and local least favorable submodel (i.e., with score $\bar{Q}_n(W) - \Psi(Q_n)$, will not change this estimator. Let $Q_n^* = (Q_{W,n}, \bar{Q}_n^*)$. The HAL-TMLE of ψ_0 is the plug-in estimator $\psi_n^* \triangleq \Psi(Q_n^*) = \frac{1}{n} \sum_{i=1}^n \bar{Q}_n^*(1, W_i)$.

Targeted HAL

For a large number of target parameters (such as average treatment effect, treatment effect among the treated, counterfactual mean outcome in a longitudinal trial), the least favorable submodel can be expressed as a generalized linear model with a univariate outcome, regressed onto a clever covariate $C(G_n)$ with an offset (for example, (3.3) for the treatment-specific mean). Targeted HAL append the clever covariate vector $C(G_n)$ into the HAL design matrix $\vec{\phi}_{s,j}$ (basis-expanded). When we run the LASSO regression, we do not put penalization on this special column, which guarantees our regression to solve the estimating equation without doing an additional targeting step. Targeted HAL takes the following form:

$$\bar{Q}_n = \sum_{s,j \in \mathcal{J}_n(s)} \beta_n(s, j) \phi_{s,j} + \gamma_n C(G_n),$$

where

$$(\beta_n, \gamma_n) \equiv \arg \min_{\beta, \|\beta\|_1 \leq C_n, \gamma} L \left(\sum_{s,j \in \mathcal{J}_n(s)} \beta(s,j)\phi(s,j) + \gamma C(G_n) \right),$$

To show that this Targeted HAL solves the estimating equation, we can use the property of OLS, where

$$\sum_{i=1}^{n} (C(G_n)(O_i)\hat{r}_i) = \sum_{i=1}^{n} (C(G_n)(O_i)(Y_i - \bar{Q}_n(A_i, W_i))) = \sum_{i=1}^{n} D^*(\bar{Q}_n, \bar{G})(O_i) = 0.$$

Targeted HAL has the same assumptions and asymptotic efficiency of HAL-TMLE, while fitting a single outcome regression.

Under-smoothed Highly Adaptive LASSO

Theorem 1 in [29] showed that undersmoothing HAL-MLE will result in efficient plug-in estimator for all pathwise differentiable parameters if

$$\min_{s,j \in \mathcal{J}_n(s)} P_0 \phi_{s,j} = o_P \left(n^{-1/2+\alpha/4} \right). \tag{3.4}$$

Here we propose one criterion to select the tuning parameter of HAL called "multi-task tuning". Our proposed selector operates under the setting where the target parameter of interest is known (same as when we apply HAL-TMLE), instead of trying to under-smooth the HAL for any arbitrary target parameter. Given the parameter Ψ (and its corresponding EIC D^*), we (automatically) come up with a large family of tasks $P_n D_j^*(Q_{n,\lambda}, G_n)$ that are "section-specific EIC equations", where $D_j^*(.)(O) = D^*(.)(O)\phi_j(O)$ is the j-specific efficient influence curve for basis ϕ_j. Solving this family of EIC equations implicitly correspond to solving the estimating equation for a family of "section-specific target parameters". We tune L_1-norm (larger than the cross-validation chosen value) such that the all of the set of "section-specific EIC equations" are approximately solved (with tolerance level carefully chosen as a function of sample size guided from semiparametric efficiency theory). The criteria cover a large family of data structures and statistical parameters, and we demonstrate using ATE parameter as follows.

Algorithm 4: Under-smoothed HAL for ATE

1 Perform HAL regression for the $\bar{Q}_0$ function using cross-validation tuning of HAL hyper-parameter (sectional variation norm) C, giving us $C_{n,cv}$. On the training data and under $C = C_{n,cv}$, denote the set of spline bases with non-zero coefficients as J;

2 For our target parameter Ψ. We propose a new family of target parameters $\Psi_j(Q) = E[(\bar{Q}(1,W) - \bar{Q}(0,W))\phi_j(W)], j \in J$ as the surrogate tasks to perform undersmoothing. The undersmoothing criteria is

$$C^* = \max_{j \in J} \arg \max_{C \geq C_{n,cv}} |P_n D_j^*(Q_{n,C}, G_n)| \leq C\sigma_{0,j}/\sqrt{n}/\log(n),$$

where $D_j^*(.)(A,W) = D^*(.)(A,W)\phi_j(W)$ is the j-specific efficient influence curve and $\sigma_{0,j}$ is estimated by the standard error of D_j^* in sample:
$\sigma_{n,j} = \sqrt{P_n[D_j^*(Q_{n,C}, G_n)^2] - [P_n D_j^*(Q_{n,C}, G_n)]^2}$;

3 Note that there is a multiplicative constant C that controls the finite sample behavior of multi-task tuning. We recommend choosing the constant C not too small. Empirical study shows that the performance of the whole pipeline is not sensitive to too large a C, but too small a C can make the learner have a large variance in finite sample. The choice of C will not affect its asymptotic performance;

4 This defines our under-smoothed HAL using multi-task tuning: $\bar{Q}_{n,C^*}$;

5 Optionally, one can perform a targeting step on top of $\bar{Q}_{n,C^*}$ towards the main target parameter of interest Ψ;

We use the simulation study to show that "multi-task tuning" satisfies the sufficient condition (3.4) defined in Theorem 1 of [29].

Theorem 3. *For an undersmoothed HAL $\bar{Q}_{n,C^*}$ that satisfies*

$$\max_{j \in J(\bar{Q}_{n,C_{n,cv}})} P_n \phi_j (Y - \bar{Q}_{n,C^*}) \leq \sigma_{n,j}/\sqrt{n}/\log n. \tag{3.5}$$

Assume that we can choose α_j such that

$$\| \sum_{j \in J} \alpha_j \phi_j - \frac{2A-1}{G_n} \|_{P_0} = O_P(n^{-\frac{1}{4}}) \tag{3.6}$$

and that

$$\{ (\sum_{j \in J} \alpha_j \phi_j - \frac{2A-1}{G_n})(Y - \bar{Q}_{n,C^*}) : \vec{\alpha} \} \tag{3.7}$$

is P_0-Donsker, which holds when $\bar{Q}_0$, $\bar{Q}_{n,C^*}$ and G_n are functions with finite sectional variation norm.

Then

$$P_n \frac{2A-1}{G_n}(Y - \bar{Q}_{n,C^*}) = O_P(n^{-\frac{1}{2}}) \tag{3.8}$$

Proof: For any $\sum_{j\in J}\alpha_j\phi_j$ such that $\sum_{j\in J}|\alpha_j|\leq M$, we have

$$\begin{aligned}P_n(\sum_{j\in J}\alpha_j\phi_j)(Y-\bar{Q}_{n,C^*}) &= \sum_{j\in J}\alpha_j P_n\phi_j(Y-\bar{Q}_{n,C^*})\\ &\leq \sum_{j\in J}|\alpha_j||P_n\phi_j(Y-\bar{Q}_{n,C^*})|\\ &\leq \sum_{j\in J}|\alpha_j|\sigma_{n,j}/\sqrt{n}/\log n\\ &\leq M\sigma_{n,max}/\sqrt{n}/\log n\end{aligned}$$

so that

$$P_n(\sum_{j\in J}\alpha_j\phi_j)(Y-\bar{Q}_{n,C^*}) = O_P(n^{-\frac{1}{2}}). \tag{3.9}$$

We now are left to show that

$$P_n(\sum_{j\in J}\alpha_j\phi_j - H_n)(Y-\bar{Q}_{n,C^*}) = O_P(n^{-\frac{1}{2}}), \tag{3.10}$$

where $H_n = \frac{2A-1}{G_n}$, so that $P_nH_n(Y-\bar{Q}_{n,C^*}) = O_P(n^{-\frac{1}{2}})$ (the efficient influence curve equation is solved).

Assume we can choose α_j such that (3.6) is satisfied, we have (by denoting $\tilde{H}_n = \sum_{j\in J}\alpha_j\phi_j$)

$$\begin{aligned}P_n(\tilde{H}_n - H_n)(Y-\bar{Q}_{n,C^*}) &= (P_n-P_0)[(\tilde{H}_n-H_n)(Y-\bar{Q}_{n,C^*})]+\\ &\quad P_0[(\tilde{H}_n-H_n)(Y-\bar{Q}_{n,C^*})]\\ &= P_0[(\tilde{H}_n-H_n)(\bar{Q}_0-\bar{Q}_{n,C^*})]+O_P(n^{-\frac{1}{2}}) &(3.11)\\ &\leq \|\tilde{H}_n-H_n\|_{P_0}\|\bar{Q}_0-\bar{Q}_{n,C^*}\|_{P_0}+O_P(n^{-\frac{1}{2}}) &(3.12)\\ &= O_P(n^{-\frac{1}{2}})+O_P(n^{-\frac{1}{2}}) &(3.13)\\ &= O_P(n^{-\frac{1}{2}}),\end{aligned}$$

where (3.11) is by the property of Donsker class (3.7), (3.12) is by Cauchy-Schwartz inequality, and (3.13) is due to (3.6) and that $\|\bar{Q}_0-\bar{Q}_{n,C^*}\|_{P_0} = O_P(n^{-\frac{1}{4}})$ (undersmoothed HAL preserves the rate of the HAL). □

Theorem 3 proves that the undersmoothed HAL is efficient for the target parameter Ψ by performing the 'multi-task tuning', as long as the clever covariates can be expressed as a linear combination of the basis in the set J with finite L-1 norm on the coefficients. The general implication of this Theorem is that any target parameter whose clever covariates can be expressed as such a linear combination can be estimated with the undersmoothed HAL plug-in estimator efficiently.

3.3 Simulation

We evaluate the proposed estimators via two simulations. In the first data generating distribution, we mimic a case where the positivity assumption holds and where there is a non-linear relationship between the outcome, the treatment, and the baseline covariate. In the second simulation, we simulate the often cited Kang & Schafer design [18] where there is violation of posivitiy assumption and strong interaction between the baseline covariates. In addition to the targeted HAL and under-smoothed HAL methods we proposed, we also include in comparison the plug-in estimator HAL-MLE and the classic HAL-TMLE [32]. We evaluate the bias, variance, mean-squared error, the EIC equation value, as well as the sampling distribution of the estimates.

In the first simulation, we draw 1000 samples of size $n \in \{100, 500, 1000\}$ from the following data generating distribution. We simulate a univariate $W \sim N(0, 4^2, -10, 10)$ i.i.d. from a truncated normal distribution with mean equals 0, standard deviation 4, bounded within $[-10, 10]$. A binary treatment $A \sim \text{Bernoulli}(p(W))$ is a Bernoulli binary random variable, with a probability $p(W)$ as a function of W, given by

$$p(W) = \begin{cases} 0.3, & \text{if } 0.3 + 0.1W\sin(0.1W) < 0.3 \\ 0.7, & \text{if } 0.3 + 0.1W\sin(0.1W) > 0.7 \\ 0.3 + 0.1W\sin(0.1W), & \text{otherwise} \end{cases}.$$

Continuous outcome $Y \sim N(3\sin(0.5W) + A, 1)$ has the conditional expectation equal a sinusoidal function of W.

As predicted by theory, all methods except HAL-MLE are root-n consistent in this data generating distribution. the MSE of targeted HAL, HAL-TMLE, undersmoothed HAL (targeted) and undersmoothed HAL are all very similar in the asymptotic. Note that undersmoothed HAL (not targeted) is already asymptotically efficient and normal, and additional targeting further improves its performance.

In the second simulation, we draw 1000 samples of size $n \in \{100, 500, 1000\}$ from the following data generating distribution. We let $W_1 = 4Z - 2$, where Z was drawn from a Beta(0.85,0.85) distribution. W_2 was independently drawn from a Bernoulli(0.5) distribution. Given $W = (w_1, w_2)$ we drew A from a Bernoulli distribution with the probability $A = 1$ equal to $\bar{G}_0(w_1, w_2) = \text{expit}(w_1 - 2w_1w_2)$. Given $A = a$, and $W = (w_1, w_2)$ we drew Y from a Normal($\bar{Q}_0(w_1, w_2)$, 0.33^2) distribution with $\bar{Q}_0(w_1, w_2) = \text{expit}(w_1 - 2w_1w_2)$. The true ATE is zero and the true propensity score falls between (0.023, 0.921).

In this design, HAL-TMLE and targeted HAL (based on using inverse propensity score weighting) are no longer root-n consistent for the target parameter, but under-smoothed HAL is still root-n consistent and is approaching the efficiency bound at the fastest rate. The bias, variance and MSE of under-smoothed HAL all outperform HAL-MLE, which used to be the best performing method in this scenario. The targeted variant of under-smoothed HAL has less bias, larger variance and overall larger MSE in finite sample. In larger sample sizes,

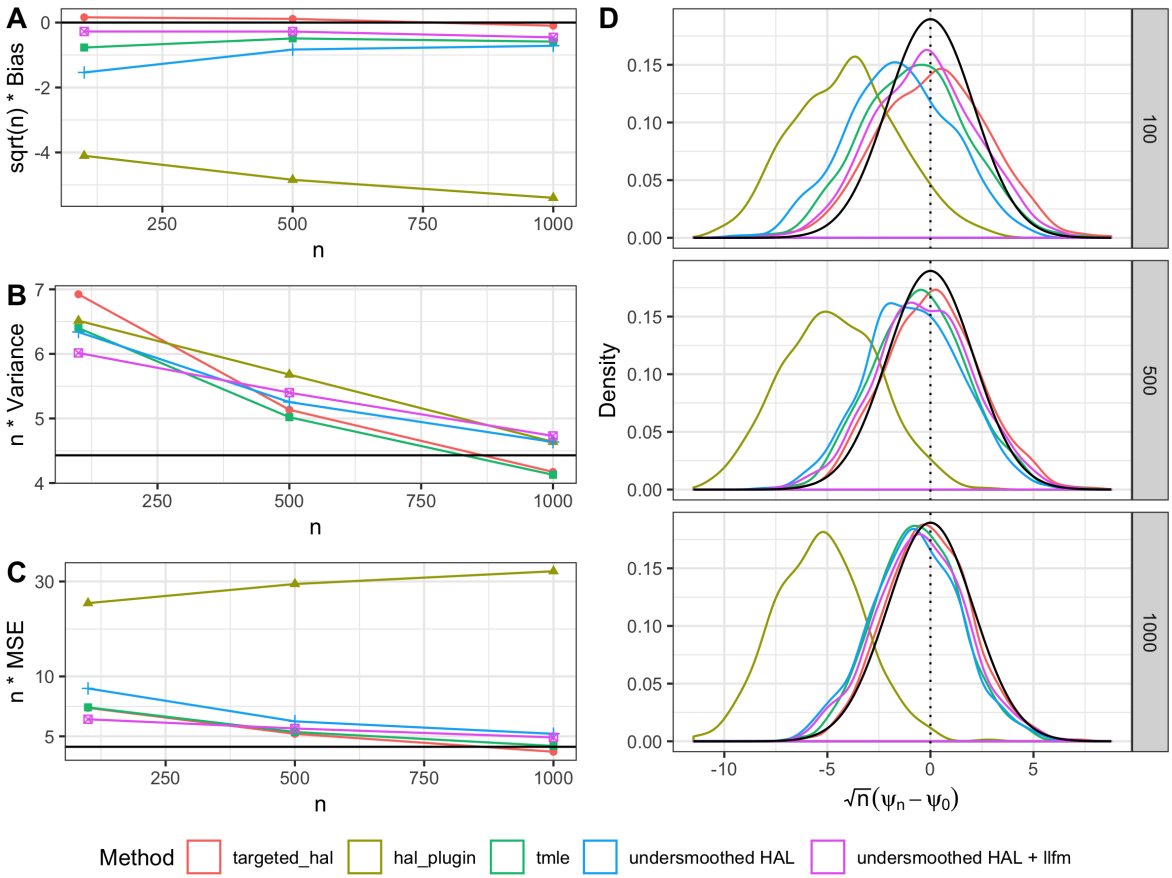

Figure 3.1: Results for simulation 1 comparing targeted HAL, under-smoothed HAL, under-smoothed HAL (plus targeting), HAL-TMLE and HAL-MLE. Each panel displays a different performance metric. Panel A: $\sqrt{n}$ times bias of the estimators. Panel B: n times Variance of the estimators. Panel C: n times MSE. Panel D: Kernel density estimates of sampling distributions using a Gaussian kernel and Silverman's rule of thumb bandwidth (Silverman, 1986). The black lines in the variance and MSE plots denote the efficiency bound. The reference sampling distribution for the estimators is a mean-zero Normal distribution with this variance (in black).

the additional targeting step is almost not updating the under-smoothed HAL, resulting in almost identical performance.

The surprisingly good performance of under-smoothed HAL in both scenarios, one which favors inverse probability weighting estimators and one which violates the assumption of inverse probability weighting, indicate that the under-smoothed HAL method is not another trade-off between fully solving the estimating equation v.s. using plug-in estimator. The majority of benefit comes from the first under-smoothing step which is very robust to positivity violation. The optional targeting step can affect finite sample performance but will

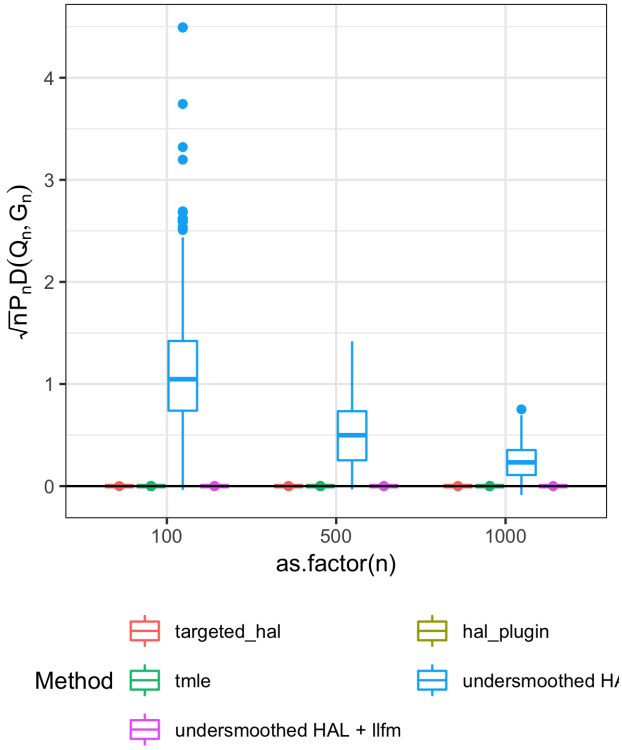

Figure 3.2: Scaled empirical average of efficient influence curve from targeted HAL, under-smoothed HAL, under-smoothed HAL (plus targeting) and HAL-TMLE. Computed under simulation 1.

not affect its asymptotic.

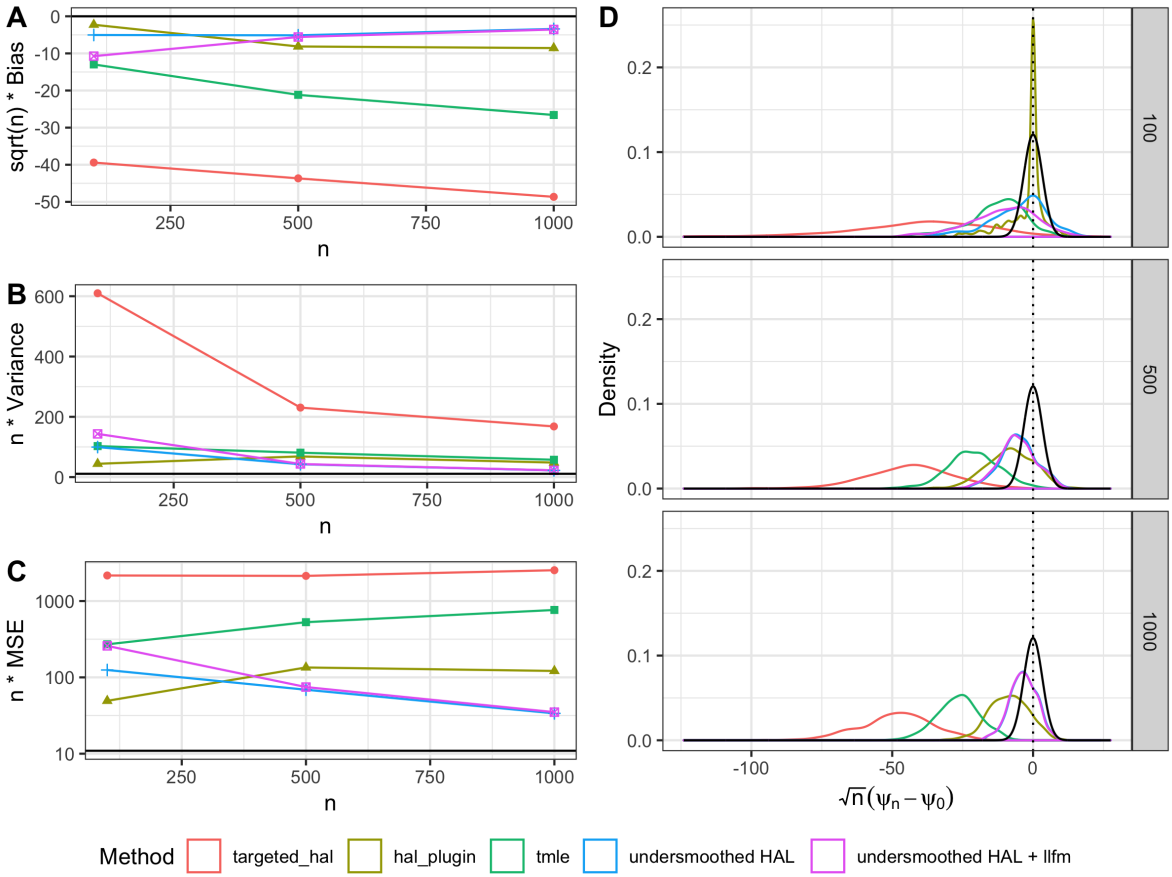

Figure 3.3: Results for simulation 2 comparing targeted HAL, under-smoothed HAL, under-smoothed HAL (plus targeting), HAL-TMLE and HAL-MLE. Each panel displays a different performance metric. Panel A: $\sqrt{n}$ times bias of the estimators. Panel B: n times Variance of the estimators. Panel C: n times MSE. Panel D: Kernel density estimates of sampling distributions using a Gaussian kernel and Silverman's rule of thumb bandwidth (Silverman, 1986). The black lines in the variance and MSE plots denote the efficiency bound. The reference sampling distribution for the estimators is a mean-zero Normal distribution with this variance (in black).

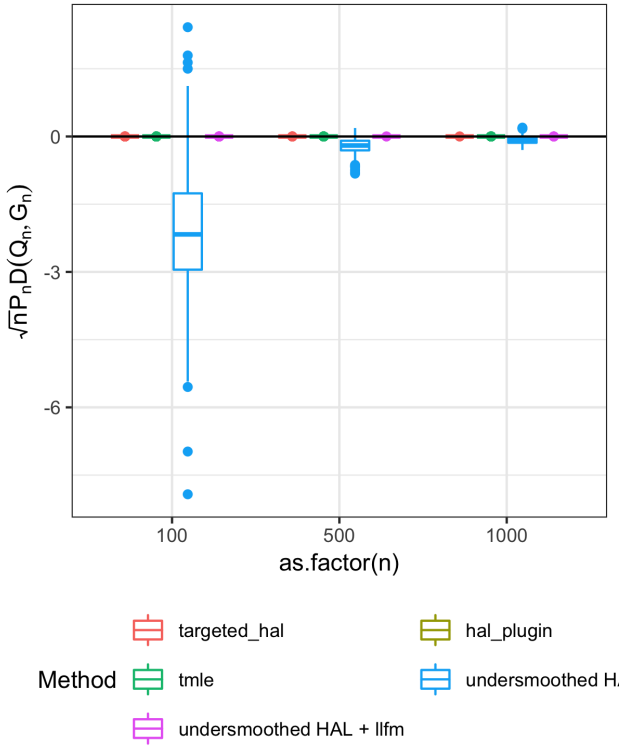

Figure 3.4: Scaled empirical average of efficient influence curve from targeted HAL, under-smoothed HAL, under-smoothed HAL (plus targeting) and HAL-TMLE. Computed under simulation 2.

Bibliography

[1] David Benkeser and Mark van der Laan. "The highly adaptive lasso estimator". In: *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*. IEEE. 2016, pp. 689–696.

[2] Peter J Bickel, Friedrich Götze, and Willem R van Zwet. "Resampling fewer than n observations: gains, losses, and remedies for losses". In: *Selected works of Willem van Zwet*. Springer, 2012, pp. 267–297.

[3] Peter J Bickel et al. *Efficient and Adaptive Estimation for Semiparametric Models*. Vol. 4. Johns Hopkins University Press Baltimore, 1993.

[4] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[5] Weixin Cai and Mark J van der Laan. *MOSS: One-step TMLE for survival analysis*. R package version 1.1.2. 2018. URL: `https://github.com/wilsoncai1992/MOSS`.

[6] Weixin Cai and Mark van der Laan. "Nonparametric Bootstrap Inference for the Targeted Highly Adaptive LASSO Estimator". In: *submitted to The international journal of biostatistics* (2019).

[7] Weixin Cai and Mark van der Laan. *TMLEbootstrap: HAL-TMLE bootstrap in R*. 2018. URL: `https://github.com/wilsoncai1992/TMLEbootstrap`.

[8] Jeremy Coyle and Mark J van der Laan. "Targeted Bootstrap". In: *Targeted Learning in Data Science*. Springer, 2018, pp. 523–539.

[9] M. Davies and M.J. van der Laan. *Sieve Plateau Variance Estimators: A New Approach to Confidence Interval Estimation for Dependent Data*. Tech. rep. U.C. Berkeley Division of Biostatistics Working Paper Series, 2014. URL: `http://biostats.bepress.com/ucbbiostat/paper322/`.

[10] Brad Efron. "Bootstrap Methods: Another Look at the Jackknife". In: *The Annals of Statistics* (1979), pp. 1–26.

[11] Jerome H Friedman et al. "Multivariate adaptive regression splines". In: *The Annals of Statistics* 19.1 (1991), pp. 1–67.

[12] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*. 2016, pp. 1050–1059.

[13] R.D. Gill, M.J. van der Laan, and J.A. Wellner. "Inefficient estimators of the bivariate survival function for three models". In: *Annales de l'Institut Henri Poincare* 31 (1995), pp. 545–597.

[14] Richard D Gill, Mark J van der Laan, and James M Robins. "Coarsening at random: Characterizations, conjectures, counter-examples". In: *Proceedings of the First Seattle Symposium in Biostatistics*. Springer. 1997, pp. 255–294.

[15] Peter Hall. *The bootstrap and Edgeworth expansion*. Springer Science & Business Media, 2013.

[16] Trevor J Hastie. "Generalized additive models". In: *Statistical models in S*. Routledge, 2017, pp. 249–307.

[17] Alan E Hubbard, Mark J van der Laan, and James M Robins. "Nonparametric locally efficient estimation of the treatment specific survival distribution with right censored data and covariates in observational studies". In: *IMA Volumes in Mathematics and Its Applications* 116 (2000), pp. 135–178.

[18] Joseph DY Kang, Joseph L Schafer, et al. "Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data". In: *Statistical science* 22.4 (2007), pp. 523–539.

[19] Robert A Kyle et al. "A long-term study of prognosis in monoclonal gammopathy of undetermined significance". In: *New England Journal of Medicine* 346.8 (2002), pp. 564–569.

[20] Mark J van der Laan, Aurélien Bibaut, and Alexander R Luedtke. "CV-TMLE for Nonpathwise Differentiable Target Parameters". In: *Targeted Learning in Data Science*. Springer, 2018, pp. 455–481.

[21] Mark J van der Laan and Susan Gruber. "One-step targeted minimum loss-based estimation based on universal least favorable one-dimensional submodels". In: *The International Journal of Biostatistics* 12.1 (2016), pp. 351–378.

[22] Mark J van der Laan and Susan Gruber. "Targeted minimum loss based estimation of an intervention specific mean outcome". In: (2011).

[23] 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).

[24] 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).

[25] Mark J van der Laan and James M Robins. *Unified Methods for Censored Longitudinal Data and Causality*. Springer Science & Business Media, 2003.

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

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

[28] Mark van der Laan. "A Generally Efficient Targeted Minimum Loss Based Estimator based on the Highly Adaptive Lasso". In: *The international journal of biostatistics* 13.2 (2017).

[29] Mark van der Laan. "Efficient Estimation of Pathwise Differentiable Target Parameters with the Undersmoothed Highly Adaptive Lasso". In: *technical report* (2019+).

[30] Mark van der Laan. "Finite Sample Inference for Targeted Learning". In: *arXiv preprint arXiv:1708.09502* (2017).

[31] Mark van der Laan and Susan Gruber. "One-Step Targeted Minimum Loss-based Estimation Based on Universal Least Favorable One-Dimensional Submodels". In: *The International Journal Of Biostatistics* 12.1 (2016), pp. 351–378.

[32] M.J. van der Laan. "A Generally Efficient Targeted Minimum Loss Based Estimator". In: *International Journal of Biostatistics* (2017), pp. 1106–1118.

[33] M.J. van der Laan. *Causal Effect Models for Intention to Treat and Realistic Individualized Treatment Rules*. Technical Report 203. Division of Biostatistics, University of California, Berkeley, 2006.

[34] M.J. van der Laan and J.M. Robins. *Unified Methods for Censored Longitudinal Data and Causality*. Berlin Heidelberg New York: Springer, 2003.

[35] Yann LeCun. "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/* (1998).

[36] Jonathan Levy et al. "A Fundamental Measure of Treatment Effect Heterogeneity". In: *arXiv preprint arXiv:1811.03745* (2018).

[37] Stephan Mandt, Matthew D Hoffman, and David M Blei. "Stochastic gradient descent as approximate Bayesian inference". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 4873–4907.

[38] Kelly Moore and Mark J van der Laan. "Application of time-to-event methods in the assessment of safety in clinical trials". In: *Design and Analysis of Clinical Trials with Time-to-Event Endpoints. Taylor & Francis* (2009), pp. 455–482.

[39] John Ashworth Nelder and Robert WM Wedderburn. "Generalized linear models". In: *Journal of the Royal Statistical Society: Series A (General)* 135.3 (1972), pp. 370–384.

[40] 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.

[41] Maya Petersen et al. "Targeted maximum likelihood estimation for dynamic and static longitudinal marginal structural working models". In: *Journal of causal inference* 2.2 (2014), pp. 147–185.

[42] James M. Robins and Andrea Rotnitzky. "Recovery of information and adjustment for dependent censoring using surrogate markers". In: *AIDS Epidemiology: Methodological Issues*. Ed. by Nicholas P. Jewell, Klaus Dietz, and Vernon T. Farewell. Boston, MA: Birkhäuser Boston, 1992, pp. 297–331. ISBN: 978-1-4757-1229-2. DOI: 10.1007/978-1-4757-1229-2_14. URL: https://doi.org/10.1007/978-1-4757-1229-2_14.

[43] Paul R Rosenbaum and Donald B Rubin. "The central role of the propensity score in observational studies for causal effects". In: *Biometrika* 70.1 (1983), pp. 41–55.

[44] Ori Stitelman and Mark J van der Laan. "Collaborative targeted maximum likelihood for time to event data". In: *The International Journal of Biostatistics* (2010).

[45] Linh Tran et al. "Robust variance estimation and inference for causal effect estimation". In: *arXiv preprint arXiv:1810.03030* (2018).

[46] Anastasios Tsiatis. *Semiparametric theory and missing data*. Springer Science & Business Media, 2007.

[47] Irvine Academic Personnel University of California. *Pay Equity Study*. http://ap.uci.edu/programs/payequity/. Accessed: 2018-12-06.

[48] A. van der Vaart and J. Wellner. *Weak Convergence and Empirical Processes*. Springer-Verlag, New York, 1996.

[49] Stefan Wager, Trevor Hastie, and Bradley Efron. "Confidence intervals for random forests: The jackknife and the infinitesimal jackknife". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1625–1651.

[50] Wenjing Zheng and Mark J van der Laan. "Asymptotic theory for cross-validated targeted maximum likelihood estimation". In: (2010).