Scalable collaborative targeted learning for high-dimensional data

Cheng J Ju, Susan Gruber, Samuel D Lendle, Antoine Chambaz, Jessica J Franklin, Richard Wyss, Sebastian Schneeweiss, Mark van Der Laan

HHS Public Access

Author manuscript

Stat Methods Med Res. Author manuscript; available in PMC 2018 August 11.

Published in final edited form as:

Stat Methods Med Res. 2019 February ; 28(2): 532–554. doi:10.1177/0962280217729845.

Scalable collaborative targeted learning for high-dimensional data

Cheng Ju[1], **Susan Gruber**[2], **Samuel D Lendle**[1], **Antoine Chambaz**[1,4], **Jessica M Franklin**[3], **Richard Wyss**[3], **Sebastian Schneeweiss**[3], and **Mark J van der Laan**[1]

[1]University of California, Berkeley, CA, USA

[2]Harvard Pilgrim Health Care Institute and Harvard Medical School, Boston, MA, USA

[3]Division of Pharmacoepidemiology and Pharmacoeconomics, Department of Medicine, Brigham and Womens Hospital and Harvard Medical School, Boston, MA, USA

[4]Modal'X, UPL, Univ Paris Nanterre, Nanterre, France

Abstract

Robust inference of a low-dimensional parameter in a large semi-parametric model relies on external estimators of infinite-dimensional features of the distribution of the data. Typically, only one of the latter is optimized for the sake of constructing a well-behaved estimator of the low-dimensional parameter of interest. Optimizing more than one of them for the sake of achieving a better bias-variance trade-off in the estimation of the parameter of interest is the core idea driving the general template of the collaborative targeted minimum loss-based estimation procedure. The original instantiation of the collaborative targeted minimum loss-based estimation template can be presented as a greedy forward stepwise collaborative targeted minimum loss-based estimation algorithm. It does not scale well when the number p of covariates increases drastically. This motivates the introduction of a novel instantiation of the collaborative targeted minimum loss-based estimation template where the covariates are pre-ordered. Its time complexity is $\mathcal{O}(p)$ as opposed to the original $\mathcal{O}(p^2)$, a remarkable gain. We propose two pre-ordering strategies and suggest a rule of thumb to develop other meaningful strategies. Because it is usually unclear a priori which pre-ordering strategy to choose, we also introduce another instantiation called SL-C-TMLE algorithm that enables the data-driven choice of the better pre-ordering strategy given the problem at hand. Its time complexity is $\mathcal{O}(p)$ as well. The computational burden and relative performance of these algorithms were compared in simulation studies involving fully synthetic data or partially synthetic data based on a real world large electronic health database; and in analyses of three real, large electronic health databases. In all analyses involving electronic health databases, the greedy collaborative targeted minimum loss-based estimation algorithm is unacceptably slow. Simulation studies seem to indicate that our scalable collaborative targeted minimum loss-based estimation and SL-C-TMLE algorithms work well. All C-TMLEs are publicly available in a Julia software package.

Corresponding author: Cheng Ju, University of California, Berkeley, CA, USA., jucheng1992@gmail.com.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Keywords

Observational study; propensity score; variable selection; targeted minimum loss-based estimation; high-dimensional data; electronic healthcare database

1 Introduction

The general template of collaborative double robust targeted minimum loss-based estimation (C-TMLE; "C-TMLE template" for short) builds upon the targeted minimum loss-based estimation (TMLE) template.[1,2] Both the TMLE and C-TMLE templates can be viewed as meta-algorithms which map a set of user-supplied choices/hyper-parameters (e.g., parameter of interest, loss function, submodels) into a specific machine-learning algorithm for estimation, that we call an instantiation of the template.

Constructing a TMLE or a C-TMLE involves the estimation of a nuisance parameter, typically an infinite-dimensional feature of the distribution of the data. For a plain TMLE estimator, the estimation of the nuisance parameter is addressed as an independent statistical task. In the C-TMLE template, on the contrary, the estimation of the nuisance parameter is optimized to provide a better bias-variance trade-off in the inference of the targeted parameter. The C-TMLE template has been successfully applied in a variety of areas, from survival analysis,[3] to the study of gene association[4] and longitudinal data structures,[5] to name just a few.

In the original instantiation of the C-TMLE template of van der Laan and Gruber,[2] that we henceforth call "the greedy C-TMLE algorithm", the estimation of the nuisance parameter aiming for a better bias-variance trade-off is conducted in two steps. First, a greedy forward stepwise selection procedure is implemented to construct a sequence of candidate estimators of the nuisance parameter derived by fitting a nested sequence of models. Second, cross-validation is used to select the candidate from this sequence which minimizes a criterion that incorporates a measure of bias and variance with respect to (w.r.t.) the targeted parameter (the algorithm is described in Section 4). The authors show that the greedy C-TMLE algorithm exhibits superior relative performance in analyses of sparse data, at the cost of an increase in time complexity. For instance, in a problem with p baseline covariates, one would construct and select from p candidate estimators of the nuisance parameter, yielding a time complexity of order $\mathcal{O}(p^2)$. Despite a criterion for early termination, the algorithm does not scale to large-scale and high-dimensional data. The aim of this article is to develop novel C-TMLE algorithms that overcome these serious practical limitations without compromising finite sample or asymptotic performance.

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

Because one may be reluctant to specify a single a priori pre-ordering of the candidate estimators of the nuisance parameter, we also introduce a SL-C-TMLE algorithm. It selects the best pre-ordering from a set of ordering strategies by Super Learning (SL).[6] SL is an example of ensemble learning methodology which builds a meta-algorithm for estimation out of a collection of individual, competing algorithms of estimation, relying on oracle properties of cross-validation.

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

The performance of the two scalable C-TMLE and SL-C-TMLE algorithms are compared with those of competing, well-established estimation methods: G-computation,[7] inverse probability of treatment weighting (IPTW),[8,9] augmented inverse probability of treatment weighted estimator (A-IPTW).[10–12] Results from unadjusted regression estimation of a point treatment effect are also provided to illustrate the level of bias due to confounding.

The article is organized as follows. Section 2 introduces the parameter of interest and a causal model for its causal interpretation. Section 3 describes an instantiation of the TMLE template. Section 4 presents the C-TMLE template and a greedy instantiation of it. Section 5 introduces the two proposed pre-ordered scalable C-TMLE algorithms, and SL-C-TMLE algorithm. Sections 6 and 7 present the results of simulation studies (based on fully or partially synthetic data, respectively) comparing the C-TMLE and SL-C-TMLE estimators with other common estimators. Section 8 presents and compares the empirical processing time of C-TMLE algorithms for different sample sizes and numbers of candidate estimators of the nuisance parameter. Section 9 compares the performance of the new C-TMLEs with standard TMLE on three real data sets. Section 10 is a closing discussion. The appendix presents a brief introduction to a Julia software that implements all the proposed C-TMLE algorithms.

2 The average treatment effect example

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

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

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

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

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

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

3 A TMLE instantiation for the ATE

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

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

where $H_0(A, W) = A/g_0(1, W) - (1 - A)/g_0(0, W)$. The notation $D^*(\bar{Q}_0, g_0)$ is slightly misleading: it suggests that $\bar{Q}_0$ and g_0 fully characterize $D^*(\bar{Q}_0, g_0)$ whereas the marginal distribution $P_{0,W}$ of W under P_0, which appears in $\Psi(P_0)$, is also needed. We nevertheless keep the notation as is for brevity. We refer the reader to Bickel et al.[13] for details about efficient influence curves.

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

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

where $H_g(A, W) = A/g(1, W) - (1 - A)/g(0, W)$. The augmented inverse probability of treatment weighted estimator (A-IPTW, or so called "DR IPTW")[14–16] and TMLE[1,17] are

two well-studied DR estimators. Taking the estimation of the ATE as an example, A-IPTW estimates $\Psi(P_0)$ by solving the EIC equation directly. Given two estimators $\bar{Q}_n$ and g_n of $\bar{Q}_0$ and g_0, setting

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

and solving (in ψ)

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

yield the A-IPTW estimator

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

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

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

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

III. **Building the so-called "clever covariate"**. Define $H_n(A,W)$ as in equation (1).

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

$$\bar{Q}_n^*(A,W) = \mathrm{expit}(\mathrm{logit}(\bar{Q}_n^0(A,W)) + \varepsilon_n H_n(A,W)) \quad (2)$$

V. V **Constructing the TMLE**. Evaluate

$$\psi_n^{\mathrm{TMLE}} = \frac{1}{n}\sum_{i=1}^{n}\left(\bar{Q}_n^*(1,W_i) - \bar{Q}_n^*(0,W_i)\right) \quad (3)$$

In steps I and II, it is highly recommended to avoid making parametric assumptions, as any parametric model is likely mis-specified. Relying on SL[6] is a good option. Step IV aims to reduce bias in the estimation of $\Psi(P_0)$ by enhancing the initial estimator derived from $\bar{Q}_n^0$ and the marginal empirical distribution of W as an estimator of its counterpart under P_0. It is dubbed a "fluctuation" step because it consists, here, in *(i)* building a parametric model through $\bar{Q}_n^0$ and *(ii)* finding the optimal fluctuation of $\bar{Q}_n^0$ in it w.r.t. the chosen loss function. In practice, bounded continuous outcomes and binary outcomes are fluctuated on the *logit* scale (hence the expression "logistic fluctuation") to ensure that bounds on the model space are respected.[18] In the context of the above TMLE algorithm, step IV consists in minimizing $\varepsilon \mapsto L_n(\bar{Q}_n^0(\varepsilon))$ over $\mathbb{R}$, where

$$L_n\left(\bar{Q}_n^0(\varepsilon)\right) = \sum_{i=1}^{n} \left(Y_i \log\left(\bar{Q}_n^0(\varepsilon)(A_i, W_i)\right) + (1 - Y_i) \log\left(1 - \bar{Q}_n^0(\varepsilon)(A_i, W_i)\right)\right) \quad (4)$$

is the empirical loss of $\bar{Q}_n^0(\varepsilon)$ given by equation (2) with $\boldsymbol{\varepsilon}$ substituted for $\boldsymbol{\varepsilon_n}$. Moreover, the fluctuation in step 4 is made in such a way that the EIC equation is solved: $\sum_i D^*(\bar{Q}_n^*, g_n)(O_i) = 0$, which justifies why $\bar{Q}_n^*$ is said to be "targeted" toward $\Psi(P_0)$. This is the key to the TMLE estimator being DR and asymptotically efficient under regularity conditions.[1]

Standard errors and confidence intervals (CIs) can be computed based on the variance of the influence curve. Proofs and technical details are available in the literature.[1,17]

4 The C-TMLE general template and its greedy instantiation for the ATE

When implementing an instantiation of the TMLE template, one relies on a single external estimate of the nuisance parameter, g_0 in the ATE example (see step 2 in Section 3). In contrast, an instantiation of the C-TMLE template involves constructing a series of nuisance parameter estimates and corresponding TMLE estimators using these estimates in the targeting step. Section 4.1 presents the C-TMLE general template and Section 4.2 its first instantiation, called the greedy C-TMLE algorithm.

4.1 The C-TMLE template

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

1. **Initialization.** Build an initial triplet $(g_{n,0}, \bar{Q}_{n,0}, \bar{Q}^*_{n,0})$ where $g_{n,0}$ estimates g_0 and $\bar{Q}_{n,0} = \bar{Q}_n^0$ and $\bar{Q}^*_{n,0}$ estimate $\bar{Q}_0$, the latter estimator being targeted toward $\Psi(P_0)$ for instance as in step 4 of the TMLE algorithm presented in Section 3.

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

2. **Deriving the next triplet.**

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

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

   c. For each j, build $\bar{Q}^{j,*}_{n,k}$ by fluctuating $\bar{Q}_{n,k}$ based on $g^j_{n,k}$ as in step 4 of the TMLE algorithm presented in Section 3 for instance.

   d. Find J such that the empirical loss (see (4) in Section 3 for an example) of $\bar{Q}^{J,*}_{n,k}$ equals the minimum among the empirical losses of $\bar{Q}^{j,*}_{n,k}$ $(1 \le j \le J_{n,k})$, then tentatively set $(g_{n,k}, \bar{Q}_{n,k}, \bar{Q}^*_{n,k}) = (g^J_{n,k}, \bar{Q}_{n,k}, \bar{Q}^{J,*}_{n,k})$.

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

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

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

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

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

As in step 1 of the TMLE instantiation presented in Section 3, we recommend relying on SL in step 1 of the above general template of C-TMLE. Two comments are in order regarding step 2. First, to achieve collaborative DR eventually, the sequence of estimators $(g_{n,k}: k)$ derived in steps 2b and 2d should be arranged in such a way that the estimator becomes increasingly nonparametric, with asymptotic bias and variance, respectively, decreasing and increasing, and so that $g_{n,k}$ converges (in k) to a consistent estimator of g_0.[1] One could for instance rely on a nested sequence of models, see Section 4.2. By doing so, the empirical fit for g_0 improves as k increases.[1,19] Second, if step 2f is carried out, then it necessarily holds

that the empirical risk of $\bar{Q}^*_{n,k}$ is smaller than that of $\bar{Q}^*_{n,k-1}$ the second time step 2e is undertaken, so the candidate triplet is accepted. In step 3, k_n is formally defined as

$$k_n = \underset{k}{\operatorname{argmin}} \left\{ \text{cvRisk}_k + \text{cvVar}_k + n \times \text{cvBias}_k^2 \right\}$$

where cvRisk_k, cvVar_k, cvBias_k are, respectively, given by

$$\sum_{v=1}^{V} \sum_{i \in \text{Val}(v)} \text{loss}(\bar{Q}^*_{n,k}(P^0_{nv}))(O_i),$$
$$\frac{1}{n} \sum_{v=1}^{V} \sum_{i \in \text{Val}(v)} D^*(\bar{Q}^*_{n,k}(P^0_{nv}), g_{n,k}(P^0_{n,v}))(O_i)^2,$$
$$\frac{1}{V} \sum_{v=1}^{V} [\Psi(\bar{Q}^*_{n,k}(P^0_{nv})) - \Psi(\bar{Q}^*_{n,k}(P_n))]$$

where $\Psi(\bar{Q}^*_{n,k}(P^0_{nv}))$ and $\Psi(\bar{Q}^*_{n,k}(P_n))$ are shorthand notation for equation (3) with $\bar{Q}^*_{n,k}(P^0_{nv})$ and $\bar{Q}^*_{n,k}(P_n)$ substituted for $\bar{Q}^*_n$, and where loss is the loss function used in step 2c to fluctuate $\bar{Q}_{n,k}$. That could be for instance the leastsquare loss function, in which case cvRisk_k would equal

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

In the two previous displays, $\text{Val}(v)$ is the set of indices of observations used for validation in the v-th fold, P^0_{nv} is the empirical distribution of the observations indexed by $i \notin \text{Val}(v)$, P_n is the empirical distribution of the whole data set, and $Z(P^0_{nv})$ (respectively, $Z(P_n)$) means that Z is fitted using P^0_{nv} (respectively, P_n). The penalization terms $\frac{1}{n}$, cvVar_k and cvBias_k robustify the finite sample performance when the positivity assumption is violated.[2]

The C-TMLE eventually defined in step 4 inherits all the properties of the plain TMLE estimator defined in equation (3).[2] It is DR and asymptotically efficient under appropriate regularity conditions. Porter et al.[20] discuss and compare TMLE and C-TMLE with other DR estimators, including A-IPTW.

Section 4.2 presents the first instantiation of the C-TMLE general template.

Algorithm 1

General Template of C-TMLE

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

4.2 The greedy C-TMLE algorithm

We refer to the first instantiation of the C-TMLE template as the greedy C-TMLE algorithm. It uses a forward selection algorithm to build the sequence of estimators of g_0 based on a nested sequence of models for g_0 that we call PS models. Let us describe the algorithm in the case that W consists of p covariates. The steps we refer to are those of the C-TMLE template of Section 4.1.

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

$$H_{g_{nk}}(A, W) = A/g_{n,k}(1 \mid W) - (1 - A)/g_{n,k}(0 \mid W) \quad (5)$$

leading to

$$\text{logit}(\bar{Q}_{n,k}^*(A, W)) = \text{logit}(\bar{Q}_{n,k}(A, W)) + \varepsilon_k H_{g_{nk}}(A, W) \quad (6)$$

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

Assume that $g_{n,1}, \ldots, g_{n,k-1}$ have already been derived by fitting PS models for g_0 where the ℓ th PS model is included (as a set) in the $(\ell + 1)$ th PS model because in the latter A is regressed on an intercept, the same $(\ell - 1)$ covariates as in the former *and* on an additional covariate (for each $1 \leq \ell \leq k$). To construct the $(k + 1)$ th PS model in step 2b, each covariate W_j $(1 \leq j \leq p$ such that W_j has not been included yet) is considered in turn as a candidate additional covariate added to the kth PS model to form the $(k + 1)$ th PS model. By fitting

the corresponding candidate $(k+1)$ th PS model, we obtain a candidate $g^j_{n,k}$. Step 2c consists in defining the corresponding $H_{g^j_{n,k}}$ and $\bar{Q}^{j,*}_{n,k}$ as in equations (5) and (6). To carry out step 2d, let the empirical risk of $\bar{Q}^{J,*}_{n,k}$ w.r.t. $\mathscr{L}$ be the smallest of the empirical risks of $\bar{Q}^{j,*}_{n,k}$ (for all considered js), let the $(k+1)$ th PS model be the one where W_J is added to the kth PS model, and set $(g_{n,k}, \bar{Q}_{n,k}, \bar{Q}^*_{n,k}) = (g^J_{n,k}, \bar{Q}_{n,k-1}, \bar{Q}^{J,*}_{n,k})$. Let $\mathscr{L}_k$ be the empirical risk of $\bar{Q}^*_{n,k}$ w.r.t. $\mathscr{L}$. In step 2e, we assess whether $\mathscr{L}_k \leq \mathscr{L}_{k-1}$ or not. If the inequality is met, then the candidate triplet is accepted. Otherwise, we reset $\bar{Q}_{n,k} = \bar{Q}^*_{n,k-1}$ and repeat steps 2c and 2d. It is then guaranteed that the empirical risk of $\bar{Q}^*_{n,k}$ w.r.t. $\mathscr{L}$ is smaller than $\mathscr{L}_{k-1}$, and the candidate triplet is accepted.

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

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

5 Scalable C-TMLE algorithms

Now that we have introduced the background on C-TMLE, we are in a position to present our scalable C-TMLE algorithm. Section 5.1 summarizes the philosophy of the scalable C-TMLE algorithm, which hinges on a data adaptively determined pre-ordering of the baseline covariates. Sections 5.2 and 5.3 present two such pre-ordering strategies. Section 5.4 discusses what properties a pre-ordering strategy should satisfy. Section 5.5 proposes a discrete Super Learner-based model selection procedure to select among a set of scalable C-TMLE estimators, which is itself a scalable C-TMLE algorithm. Finally, Section 5.6 sketches how to adapt scalable C-TMLEs to other estimation problems, with the example of the relative risk (RR).

5.1 Outline

A $\mathcal{O}(p^2)$ time complexity when there are p covariates is unsatisfactory for large-scale and high-dimensional data, a situation which is increasingly common in health care research. For example, the high-dimensional propensity score (hdPS) algorithm is a method to extract information from electronic medical claims data that produces hundreds or even thousands of candidate covariates, increasing the dimension of the data dramatically.[21]

In order to make it possible to apply C-TMLE algorithms to such data sets, we propose to add a new preordering procedure after the initial estimation of $\bar{Q}_0$ and before the stepwise

construction of the candidate $\bar{Q}^*_{n,0}, \bar{Q}^*_{n,1}, \ldots, \bar{Q}^*_{n,k}, \ldots$. We present two pre-ordering procedures in Sections 5.2 and 5.3. By imposing an ordering over the covariates, only one covariate is eligible for inclusion in the PS model at each step when constructing the next candidate $\bar{Q}^*_{n,k}$. In other words, $J_{n,k}$ equals 1 in steps 2b and 2c, and $J = j = 1$ in step 2d of the C-TMLE general template presented in Section 4.1. Therefore, the computational time of a scalable CTMLE algorithm w.r.t. p is $\mathcal{O}(\sum_{i=1}^{p} 1) = \mathcal{O}(p)$ time units (the $\mathcal{O}$ accounts for the cross-validation).

5.2 Logistic pre-ordering strategy

The logistic pre-ordering procedure is similar to step 2 of the C-TMLE general template specialized to the greedy C-TMLE algorithm of Section 4.2. However, instead of selecting one single covariate before going on, we use the empirical losses w.r.t. $\mathcal{L}$ to order the covariates by how much they can improve the predictive performance of $\bar{Q}^0_n$ (or, *heuristically, by their ability to reduce bias)*. More specifically, for each covariate W_k ($1 \leq k \leq p$), we construct an estimator $g_{n,k}$ of the conditional distribution of A given W_k only (one might also add W_k to a fixed baseline model); we define a clever covariate as in equation (5) using $g_{n,k}$ and fluctuate $\bar{Q}^0_n$ as in equation (6); we compute the empirical loss of the resulting $\bar{Q}^*_{n,k}$ w.r.t. $\mathcal{L}$, yielding $\mathcal{L}_k$. Finally, the covariates are ranked by increasing values of the empirical loss. This is summarized in Algorithm 2.

Algorithm 2

Logistic Pre-Ordering Algorithm

1. **for** each covariate W_k in W **do**
2. Construct an estimator $g_{n,k}$ of g_0 using a logistic model with W_k as predictor.
3. Define a clever covariate $H_{g_{n,k}}(A, W_k)$ as in (5).
4. Fit ε_k by regressing Y on $H_{g_{n,k}}(A, W_k)$ with i-specific offset/intercept $\text{logit}(\bar{Q}^0_n(A_i, W_{k,i}))$.
5. Define $\bar{Q}^*_{n,k}$ as in (6).
6. Compute the empirical loss $\mathcal{L}_k$ w.r.t. $\mathcal{L}$.
7. **end for**
8. Rank the covariates by increasing $\mathcal{L}_k$.

5.3 Partial correlation pre-ordering strategy

In the greedy C-TMLE algorithm described in Section 4.2, once k covariates have already been selected, the $(k+1)$ th is that remaining covariate which provides the largest reduction in the empirical loss w.r.t. $\mathcal{L}$. Heuristically, the $(k+1)$ th covariate is the one that best explains the residual between Y and $\bar{Q}^*_{n,k}$. Drawing on this idea, the partial correlation pre-ordering procedure ranks the p covariates based on how each of them is correlated with the residual between Y and *the initial* $\bar{Q}^0_n$ within strata of A. This second strategy is less

computationally demanding than the previous one because there is no need to fit any regression models, all one has to do is merely to estimate p partial correlation coefficients.

Let $\rho(X_1,X_2)$ denote the Pearson correlation coefficient between X_1 and X_2. Recall that the partial correlation $\rho(X_1,X_2|X_3)$ between X_1 and X_2 given X_3 is defined as the correlation coefficient between the residuals R_{X_1} and R_{X_2} resulting from the linear regression of X_1 on X_3 and of X_2 on X_3, respectively.[22] For each $1 \leq k \leq p$, we introduce $R = Y - \bar{Q}_n^0(A, W)$

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

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

Algorithm 3

Partial Correlation Pre-Ordering Algorithm

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

5.4 Discussion of the design of pre-ordering

Sections 5.2 and 5.3 propose two pre-ordering strategies. In general, a rule of thumb for designing a pre-ordering strategy is to rank the covariates based on the impact of each in reducing the residual bias in the target parameter which results from the initial estimator $\bar{Q}_n^0$ of $\bar{Q}_0$. In this light, the logistic ordering of Section 5.2 uses TMLE to reflect the importance of each variable w.r.t. its potential to reduce residual bias. The partial correlation ordering of Section 5.3 ranks the covariates according to the partial correlation of residual of the initial fit and the covariates, conditional on treatment.

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

5.5 Super learner-based C-TMLE algorithm

Here, we explain how to combine several C-TMLE algorithms into one. The combination is based on a (SL). SL is an ensemble machine learning approach that relies on cross-validation. It has been proven that a SL selector can perform asymptotically as well as an oracle selector under mild assumptions.[6,23,24]

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

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

Algorithm 4

Super Learner C-TMLE Algorithm

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

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

We compare the pre-ordered C-TMLE algorithms and SL-C-TMLE algorithm with greedy C-TMLE algorithm and other common methods in Sections 6 and 9.

5.6 Extending to other estimation problems

We have claimed that the scalable C-TMLEs presented so far, which are tailored to the estimation of the ATE, can be easily adapted to other estimation problems. Say for instance

that the RR is the target parameter: $\Psi'(P_0) = \mathbb{E}_0[\mathbb{E}_0(Y|A=1,W)]/\mathbb{E}_0[\mathbb{E}_0(Y|A=0,W)]$. Then it suffices to adapt the targeting step (6). We now define two clever covariates

$$H^0_{g_{nk}}(A,W) = -(1-A)/g_{n,k}(0,W)$$
$$H^1_{g_{nk}}(A,W) = A/g_{n,k}(1,W)$$

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

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

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

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

See Rose and van der Laan[25] for details.

6 Simulation studies on fully synthetic data

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

Denoting $\bar{Q}^0_n$ and g_n two initial estimators of $\bar{Q}_0$ and g_0, the unadjusted, G-computation/MLE, and IPTW estimators of the ATE parameter are given by equations (7) to (9)

$$\psi^{\text{unadj}}_n = \frac{\sum_{i=1}^n A_i Y_i}{\sum_{i=1}^n A_i} - \frac{\sum_{i=1}^n (1-A_i)Y_i}{\sum_{i=1}^n (1-A_i)} \quad (7)$$

$$\psi^{\text{MLE}}_n = \frac{1}{n}\sum_{i=1}^{n}(Q^0_n(1,W_i) - Q^0_n(0,W_i)) \quad (8)$$

$$\psi_n^{\mathrm{IPTW}} = \frac{1}{n}\sum_{i=1}^{n}\frac{(2A_i - 1)Y_i}{g_n(A_i, W_i)} \quad (9)$$

$$\psi_n^{\mathrm{A-IPTW}} = \frac{1}{n}\sum_{i=1}^{n}\frac{(2A_i - 1)}{g_n(A_i \mid W_i)}(Y_i - Q_n^0(W_i, A_i)) + \frac{1}{n}\sum_{i=1}^{n}(Q_n^0(1, W_i) - Q_n^0(0, W_i)) \quad (10)$$

The A-IPTW and TMLE estimators are presented in Section 3. The estimators yielded by the C-TMLE and scalable C-TMLE algorithms are presented in Sections 4.2 and 5.

In all simulation studies, the definitions of the TMLE (3), IPTW (9) and A-IPTW (10) estimators involve an estimator g_n of g_0 obtained by fitting a correctly specified, main terms logistic regression PS model. The definitions of the C-TMLEs also involve estimators obtained by fitting main terms logistic regression PS model but with an additional layer of variable selection.

The simulation studies of Sections 6.1 and 6.2 illustrate the relative performance of the estimators in scenarios with highly correlated covariates. These two scenarios are by far the most challenging settings for the greedy C-TMLE and scalable C-TMLE algorithms. The simulation studies of Section 6.3 and 6.4 illustrate performance in situations where instrumental variables (covariates predictive of the treatment but not of the outcome) are included in the true PS model. In these two scenarios, greedy C-TMLE and our scalable C-TMLEs are expected to perform better, if not much better, than other widely used doubly-robust methods.

6.1 Simulation study 1: low-dimensional, highly correlated covariates

In the first simulation study, data were simulated based on a data generating distribution published by Freedman and Berk[26] and further analyzed by Petersen et al.[27] A pair of correlated, multivariate normal baseline covariates (W_1, W_2) is generated as $(W_1, W_2) \sim N(\mu, \Sigma)$ where $\mu_1 = 0.5$, $\mu_2 = 1$ and $\Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$. The PS g_0 is given by

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

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

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

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

Bias, variance, and mean squared error (MSE) for all estimators across 1,000 simulated data sets are shown in Table 1. Box plots of the estimated ATE are shown in Figure 1.

When the model for $\bar{Q}_0$ was correctly specified, all estimators had very small bias. As Freedman and Berk[26] discussed, even when the correct PS model was used, near positivity violations could lead to finite sample bias for IPTW estimators.[27] Scalable C-TMLEs had smaller bias than the other DR estimators, but the distinctions were small.

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

6.2 Simulation study 2: highly correlated covariates

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

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

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

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

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

In this case, the true confounders are W_1, W_2, W_4, W_6, W_7. Covariate W_5 is most closely related to W_6. Covariate W_3 is mainly associated with W_7. Neither W_3 nor W_5 is a confounder (both of them are predictive of treatment A, but do not influence directly outcome Y). Including either one of them in the PS model should inflate the variance.[28]

As in Section 6.1, each of the estimators involving the estimation of $\bar{Q}_0$ was implemented twice: by fitting a model correctly specified for $\bar{Q}_0$, and by regressing Y on A only in a mis-specified linear model.

Table 2 demonstrates and compares performance across 1000 replications. Box plots of the estimated ATE are shown in Figure 2. When $\bar{Q}_0$ was estimated by fitting a correctly specified model, all estimators except the unadjusted estimator had small bias. The DR estimators had lower MSE than the inefficient IPTW estimator. When $\bar{Q}_0$ was estimated by fitting a mis-specified model, the A-IPTW and IPTW estimators were less biased than the C-TMLE estimators. The bias of the greedy C-TMLE was five times larger. However, all DR estimators had lower MSE than the IPTW estimator, with the TMLE outperforming the others.

6.3 Simulation study 3: binary outcome with instrumental variable

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

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

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

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

As in Sections 6.1 and 6.2, each of the estimators involving the estimation of $\bar{Q}_0$ was implemented twice: by fitting a model correctly specified for $\bar{Q}_0$, and by regressing Y on A only in a mis-specified linear model.

Table 3 demonstrates the performance of the estimators across 1000 replications. Figure 3 shows box plots of the estimates for the different methods across 1000 simulation, with a well-specified or mis-specified model for $\bar{Q}_0$.

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

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

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

1. Use a greedy C-TMLE algorithm to select the covariates.
2. Use main terms logistic regression with selected covariates for the PS model.
3. Compute IPTW using the estimated PS.

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

6.4 Simulation study 4: continuous outcome

In the fourth simulation, we assess the performance of C-TMLEs in a simulation scheme with a continuous outcome inspired by that of Gruber and van der Laan[29] (we merely increased the coefficient in front of W_1 to introduce a stronger positivity violation). We first independently draw $W_1, W_2, W_3, W_4, W_5, W_6$ from the standard normal law, then A given W with

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

and, finally Y given (A, W) from a Gaussian law with variance 1 and mean

$$\bar{Q}_0(A, W) = 0.5 \times W_1 - 8 \times W_2 + 9 \times W_3 - 2 \times W_5 + A$$

The initial estimator $\bar{Q}_n^0$ was built based on a linear regression model of Y on A, W_1, and W_2, thus partially adjusting for confounding. There was residual confounding due to W_3. There was also residual confounding due to W_1 and W_2 within at least one stratum of A, despite their inclusion in the initial outcome regression model.

Figure 4 reveals that the C-TMLEs performed much better than TMLE and A-IPTW estimators in terms of bias and standard error. This illustrates that choosing to adjust for less than the full set of covariates can improve finite sample performance when there are near positivity violations. In addition, Table 4 shows that the pre-ordered C-TMLEs out-performed the greedy C-TMLE. Although the greedy C-TMLE estimator had smaller bias, it had higher variance, perhaps due to its more data adaptive ordering procedure.

7 Simulation study on partially synthetic data

The aim of this section is to compare TMLE and all C-TMLEs using a large simulated data set that mimics a real-world data set. Section 7.1 starts the description of the data-generating scheme and resulting large data set. Section 7.2 presents the high-dimensional propensity score (hdPS) method used to reduce the dimension of the data set. Section 7.3 completes the description of the data-generating scheme and specifies how $\bar{Q}_0$ and g_0 are estimated. Section 7.4 summarizes the results of the simulation study.

7.1 Data-generating scheme

The simulation scheme relies on the Nonsteroidal anti-inflammatory drugs (NSAID) data set presented and studied in Schneeweiss et al.[21] and Rassen and Schneeweiss.[30] Its n=49,653 observations were sampled from a population of patients aged 65 years and older, and enrolled in both Medicare and the Pennsylvania Pharmaceutical Assistance Contract for the Elderly (PACE) programs between 1995 and 2002. Each observed data structure consists of a triplet (W, A, Y) where W is decomposed in two parts: a vector of 22 baseline covariates and a highly sparse vector of C =9,470 unique claims codes. In the latter, each entry is a nonnegative integer indicating how many times (mostly zero) a certain procedure (uniquely identified among C = 9,470 by its claims code) has been undergone by the corresponding patient. The claims codes were manually grouped into eight categories: ambulatory diagnoses, ambulatory procedures, hospital diagnoses, hospital procedures, nursing home diagnoses, physician diagnoses, physician procedures and prescription drugs. The binary indicator A stands for exposure to a selective COX-2 inhibitor or a comparison drug (a non-selective NSAID). Finally, the binary outcome Y indicates whether or not either a hospitalization for severe gastrointestinal hemorrhage or peptic ulcer disease complications including perforation in GI patients occurred.

The simulated data set was generated as in Gadbury et al.[31] and Franklin et al.[32] It took the form of n = 49,653 data structures (W_i, A_i, Y_i) where $\{(W_i, A_i) : 1 \leq i \leq n\}$ was extracted from the above real data set and where $\{Y_i : 1 \leq i \leq n\}$ was simulated by us in such a way that, for each $1 \leq i \leq n$, the random sampling of Y_i depended only on the corresponding (W_i, A_i). As argued in the aforementioned articles, this approach preserves the covariance structure of the covariates and complexity of the true treatment assignment mechanism, while allowing the true value of the ATE parameter to be known. In addition, we can control the bias in the unadjusted estimator by tuning the coefficients of the parametric data generating conditional distribution of Y given (A, W), if there exist covariates associated with the treatment mechanism.

7.2 High-dimensional propensity score method for dimension reduction

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

the hdPS method for dimension reduction prior to applying the TMLE and C-TMLE algorithms.

Introduced in Schneeweiss et al.,[21] the hdPS method was proposed to reduce the dimension in large electronic healthcare databases. It is increasingly used in studies involving such databases.[30,33–37]

The hdPS method essentially consists of two main steps: *(i)* generating so-called hdPS covariates from the claims data (which can increase the dimension) then *(ii)* screening the enlarged collection of covariates to select a small proportion of them (which dramatically reduces the dimension). Specifically, the method unfolds as follows[21]:

a. **Group by resource.** Group the data by resource in $\mathscr{C}$ groups

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

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

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

In the current example, we derived $\mathscr{C} = 8$ groups in step a. The groups correspond to the following categories: ambulatory diagnoses, ambulatory procedures, hospital diagnoses, hospital procedures, nursing home diagnoses, physician diagnoses, physician procedures and prescription drugs. See Schneeweiss et al.[21] and Patorno et al.[33] for other examples.

In step b, we chose J=50. The dimension of the claims data thus went from 9470 to 400.

In step c, we relied on the following estimate of the measure of the potential confounding impact introduced in Bross:[38] for hdPS covariate c^ℓ

$$\frac{\pi_n^\ell(1)(r_n^\ell - 1) + 1}{\pi_n^\ell(0)(r_n^\ell - 1) + 1} \quad (11)$$

where

$$\pi_n^{\ell}(a) = \frac{\sum_{i=1}^{n} \mathbf{1}\{c_i^{\ell} = 1, a_i = a\}}{\sum_{i=1}^{n} \mathbf{1}\{a_i = a\}} \quad (a = 0, 1)$$

$$r_n^{\ell} = \frac{p_n(1)}{p_n(0)} \quad \text{with}$$

$$p_n(c) = \frac{\sum_{i=1}^{n} \mathbf{1}\{y_i = 1, c_i^{\ell} = c\}}{\sum_{i=1}^{n} \mathbf{1}\{c_i^{\ell} = c\}} \quad (c = 0, 1)$$

A rationale for this choice can be found in Schneeweiss et al.,[21] where r_n^{ℓ} in equation (11) is replaced by $\max(r_n^{\ell}, 1/r_n^{\ell})$. As explained below we chose K=100. As a result, the dimension of the claims data was thus reduced to 100 from 9470.

7.3 Data-generating scheme (cont.) and estimating procedures

Let us resume here the presentation of the simulation scheme initiated in Section 7.1. Recall that the simulated data set is written as $\{(W_i, A_i, Y_i) : 1 \le i \le n\}$ where $\{W_i : 1 \le i \le n\}$ is the by-product of the hdPS method of Section 7.2 with J=50 and K=100 and $\{A_i : 1 \le i \le n\}$ is the original vector of exposures. It only remains to present how $\{Y_i : 1 \le i \le n\}$ was generated.

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

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

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

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

The resulting true value of the ATE is $\psi_0 = 0{:}21156$.

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

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

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

7.4 Results

Table 5 reports the point estimates for ψ_0 as derived by all the considered methods. It also reports the 95% CIs of the form $[\psi_n \pm 1.96\sigma_n/\sqrt{n}]$, where $\sigma_n^2 = n^{-1}\sum_{i=1}^{n} D^*(\bar{Q}_n, g_n)(O_i)^2$ estimates the variance of the efficient influence curve at the couple $(\bar{Q}_n, g_n)$ yielding ψ_n. We refer the interested reader to van der Laan and Rose[1] (Appendix 1) for details on influence curve based inference. All the CIs contained the true value of ψ_0. Table 5 also reports processing times (in seconds).

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

8 Time complexity

We study here the computational time of the pre-ordered C-TMLE algorithms. The computational time of each algorithm depends on the sample size n and number of covariates p. First, we set n=1000 and varied p between 10 and 100 by steps of 10. Second, we varied n from 1000 to 20,000 by steps of 1000 and set p = 20. For each (n, p) pair, the analysis was replicated 10 times independently, and the median computational time was reported. In every data set, all the random variables are mutually independent. The results are shown in Figure 5(a) and (b).

Figure 5(a) is in line with the theory: the computational time of the forward stepwise C-TMLE is $\mathcal{O}(p^2)$ whereas the computational times of the pre-ordered C-TMLE algorithms are $\mathcal{O}(p)$. Note that the pre-ordered C-TMLEs are indeed scalable. When n=1000 and p=100, all the scalable C-TMLE algorithms ran in less than 30 s.

Figure 5(b) reveals that the pre-ordered C-TMLE algorithms are much faster in practice than the greedy C-TMLE algorithm, even if all computational times are $\mathcal{O}(n)$ in that framework with fixed p.

9 Real data analyses

This section presents the application of variants of the TMLE and C-TMLE algorithms for the analysis of three real data sets. Our objectives are to showcase their use and to illustrate the consistency of the results provided by the scalable and greedy C-TMLE estimators. We thus do not implement the competing unadjusted, G-computation/MLE, IPTW and A-IPTW estimators (see the beginning of Section 6).

In Sections 6 and 7, we knew the true value of the ATE. This is not the case here.

9.1 Real data sets and estimating procedures

We compared the performance of variants of TMLE and C-TMLE algorithms across three observational data sets. Here are brief descriptions, borrowed from Schneeweiss et al.[21] and Ju et al..[37]

9.1.1 NSAID data set—Refer to Section 7.1 for its description.

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

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

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

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

Each data set is given by $\{(W_i, A_i, Y_i) : 1 \leq i \leq n\}$ where $\{W_i : 1 \leq i \leq n\}$ is the by-product of the hdPS method of Section 7.2 with J=100 and K=200 and $\{(A_i, Y_i) : 1 \leq i \leq n\}$ is the original collection of paired exposures and outcomes.

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

To save computational time, the C-TMLE algorithms relied on the same early stopping rule described in Section 7.3. The scalable SL-C-TMLE library included the two scalable pre-ordered C-TMLE algorithms and excluded the greedy C-TMLE algorithm.

9.2 Results on the NSAID data set

Figure 6 shows the point estimates and 95% CIs yielded by the different TMLE and C-TMLE estimators built from the NSAID data set.

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

9.3 Results on the NOAC data set

Figure 7 shows the point estimates and 95% CIs yielded by the different TMLE and C-TMLE estimators built on the NOAC data set.

We observe more variability in the results than in those presented in section 9.2.

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

9.4 Results on the Vytorin data set

Figure 8 shows the point estimates and 95% CIs yielded by the different TMLE and C-TMLEs built on the Vytorin data set.

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

10 Discussion

Robust inference of a low-dimensional parameter in a large semi-parametric model traditionally relies on external estimators of infinite-dimensional features of the distribution of the data. Typically, only one of the latter is optimized for the sake of constructing a well-behaved estimator of the low-dimensional parameter of interest. For instance, the targeted minimum loss (TMLE) estimator of the average treatment effect (ATE) (3) relies on an external estimator $\bar{Q}_n^0$ of the conditional mean $\bar{Q}_0$ of the outcome given binary treatment and baseline covariates, and on an external estimator g_n of the PS g_0. Only $\bar{Q}_n^0$ is optimized/updated into $\bar{Q}_n^*$ based on g_n in such a way that the resulting substitution estimator of the ATE can be used, under mild assumptions, to derive a narrow confidence interval with a given asymptotic level.

There is room for optimization in the estimation of g_0 for the sake of achieving a better bias-variance trade-off in the estimation of the ATE. This is the core idea driving the general C-TMLE template. It uses a targeted penalized loss function to make smart choices in determining which variables to adjust for in the estimation of g_0, only adjusting for variables

that have not been fully exploited in the construction of $\bar{Q}_n^0$, as revealed in the course of a data-driven sequential procedure.

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

The C-TMLE algorithms used in our data analyses have been implemented in Julia and are publicly available at https://lendle.github.io/TargetedLearning.jl/. We undertook five simulation studies. Four of them involved fully synthetic data. The last one involved partially synthetic data based on a real electronic health database and the implementation of a hdPS method for dimension reduction widely used for the statistical analysis of claims codes data. In Section 8, we compare the computational times of variants of C-TMLE algorithms. We also showcase the use of C-TMLE algorithms on three real electronic health database. In all analyses involving electronic health databases, the greedy C-TMLE algorithm was unacceptably slow. Judging from the simulation studies, our scalable C-TMLE algorithms work well, and so does the SL-C-TMLE algorithm.

This article focused on ATE with a binary treatment. In future work, we will adapt the theory and practice of scalable C-TMLE algorithms for the estimation of the ATE with multi-level or continuous treatment by employing a working marginal structural model. We will also extend the analysis to address the estimation of other classical parameters of interest.

Acknowledgments

The authors are grateful for the excellent suggestions of the associate editor and reviewers. They proved very useful and led to a much better version of the article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This project is supported by NIH grant R01 AI074345-08, PCORI contract ME-1303-5638, and the project Labex MME-DII (ANR11-LBX-0023-01).

References

1. van der Laan, MJ, Rose, S. Targeted learning: causal inference for observational and experimental data. New York, NY: Springer Science & Business Media; 2011.
2. van der Laan MJ, Gruber S. Collaborative double robust targeted maximum likelihood estimation. Int J Biostat. 2010; 6
3. Stitelman OM, Wester CW, De Gruttola V, et al. Targeted maximum likelihood estimation of effect modification parameters in survival analysis. Int J Biostat. 2011; 7

4. Wang H, Rose S, van der Laan MJ. Finding quantitative trait loci genes with collaborative targeted maximum likelihood learning. Stat Probabil Lett. 2011; 81:792–796.
5. Stitelman OM, van der Laan MJ. Collaborative targeted maximum likelihood for time to event data. Int J Biostat. 2010; 6
6. van der Laan MJ, Polley EC, Hubbard AE. Super learner. Stat Appl Genetics Mol Biol. 2007; 6
7. Robins JM. A new approach to causal inference in mortality studies with sustained exposure periods - application to control of the healthy worker survivor effect. Math Model. 1986; 7:1393–1512.
8. Hernan MA, Brumback B, Robins JM. Marginal structural models to estimate the causal effect of zidovudine on the survival of HIV-positive men. Epidemiology. 2000; 11:561–570. [PubMed: 10955409]
9. Robins, JM. Statistical models in epidemiology, the environment, and clinical trials. New York, NY: Springer; 2000. Marginal structural models versus structural nested models as tools for causal inference; 95–133.
10. Robins JM, Rotnitzky A. Comment on the Bickel and Kwon article, 'Inference for semiparametric models: Some questions and an answer'. Statistica Sinica. 2001; 11:920–936.
11. Robins JM, Rotnitzky A, van der Laan M. Comment on "On Profile Likelihood" by S.A. Murphy and A.W van der Vaart. J Am Stat Assoc – Theory Meth. 2000; 450:431–435.
12. Robins, J. Robust estimation in sequentially ignorable missing data and causal inference models. Proceedings of the American Statistical Association: Section on Bayesian Statistical Science; 8–12 August 1999; 6–10.
13. Bickel, PJ, Klaassen, CA, Ritov, Y. , et al. Efficient and adaptive estimation for semiparametric models. Springer-Verlag; 1998.
14. Robins JM, Rotnitzky A, Zhao LP. Estimation of regression coefficients when some regressors are not always observed. J Am Stat Assoc. 1994; 89:846–866.
15. Robins JM, Hernan MA, Brumback B. Marginal structural models and causal inference in epidemiology. Epidemiology. 2000; 11:550–560. [PubMed: 10955408]
16. van der Laan, MJ, Robins, JM. Unified methods for censored longitudinal data and causality. Springer Science & Business Media; 2003.
17. van der Laan MJ, Rubin D. Targeted maximum likelihood learning. Int J Biostat. 2006; 2
18. Gruber S, van der Laan MJ. A targeted maximum likelihood estimator of a causal effect on a bounded continuous outcome. Int J Biostat. 2010; 6
19. Gruber S, van der Laan MJ. An application of collaborative targeted maximum likelihood estimation in causal inference and genomics. Int J Biostat. 2010; 6
20. Porter KE, Gruber S, van der Laan MJ, et al. The relative performance of targeted maximum likelihood estimators. Int J Biostat. 2011; 7
21. Schneeweiss S, Rassen JA, Glynn RJ, et al. High-dimensional propensity score adjustment in studies of treatment effects using health care claims data. Epidemiology. 2009; 20:512. [PubMed: 19487948]
22. Hair, JF, Black, WC, Babin, BJ. , et al. Multivariate data analysis. Vol. 6. Upper Saddle River, NJ: Pearson Prentice Hall; 2006.
23. van der Laan, MJ; Dudoit, S. [accessed January 2016] Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: finite sample oracle inequalities and examples. U.C. Berkeley Division of Biostatistics Working Paper Series. 2003. Working Paper 130, http://works.bepress.com/sandrine_dudoit/34/
24. van der Vaart AW, Dudoit S, Laan MJ. Oracle inequalities for multi-fold cross validation. Stat Decis. 2006; 24:351–371.
25. Rose, S, van der Laan, MJ. Targeted learning. New York, NY: Berlin Heidelberg Springer; 2011. Understanding tmle; 83–100.
26. Freedman DA, Berk RA. Weighting regressions by propensity scores. Eval Rev. 2008; 32:392–409. [PubMed: 18591709]
27. Petersen ML, Porter KE, Gruber S, et al. Diagnosing and responding to violations in the positivity assumption. Stat Meth Med Res. 2012; 21:31–54.

28. Brookhart MA, Schneeweiss S, Rothman KJ, et al. Variable selection for propensity score models. Am J Epidemiol. 2006; 163:1149–1156. [PubMed: 16624967]
29. Gruber, S, van der Laan, MJ. Targeted learning. New York, NY: Berlin Heidelberg Springer; 2011. C-tmle of an additive point treatment effect; 301–321.
30. Rassen JA, Schneeweiss S. Using high-dimensional propensity scores to automate confounding control in a distributed medical product safety surveillance system. Pharmacoepidemiol Drug Safe. 2012; 21:41–49.
31. Gadbury GL, Xiang Q, Yang L, et al. Evaluating statistical methods using plasmode data sets in the age of massive public databases: an illustration using false discovery rates. PLoS Genet. 2008; 4:e1000098. [PubMed: 18566659]
32. Franklin JM, Schneeweiss S, Polinski JM, et al. Plasmode simulation for the evaluation of pharmacoepidemiologic methods in complex healthcare databases. Computat Stat Data Anal. 2014; 72:219–226.
33. Patorno E, Glynn RJ, Hernández-Díaz S, et al. Studies with many covariates and few outcomes: selecting covariates and implementing propensity-score–based confounding adjustments. Epidemiol. 25:268–278.
34. Franklin JM, Eddings W, Glynn RJ, et al. Regularized regression versus the high-dimensional propensity score for confounding adjustment in secondary database analyses. Am J Epidemiol. 2015; 187:651–659.
35. Toh S, García Rodríguez LA, Hernán MA. Confounding adjustment via a semi-automated high-dimensional propensity score algorithm: an application to electronic medical records. Pharmacoepidemiol Drug Safe. 2011; 20:849–857.
36. Kumamaru H, Gagne JJ, Glynn RJ, et al. Comparison of high-dimensional confounder summary scores in comparative studies of newly marketed medications. J Clin Epidemiol. 2016; 76:200–208. [PubMed: 26931292]
37. Ju C, Combs M, Lendle SD, et al. Propensity score prediction for electronic healthcare databases using super learner and high-dimensional propensity score methods. 2017
38. Bross I. Misclassification in 2×2 tables. Biometrics. 1954; 10:478–486.

Appendix 1. C-TMLE software

A flexible Julia software package implementing all C-TMLE algorithms described in this article is publicly available at https://lendle.github.io/TargetedLearning.jl/. The website contains detailed documentation and a tutorial for researchers who do not have experience with Julia.

In addition to the two pre-ordering methods described in Section 5, the software accepts any user-defined ranking algorithm. The software also offers several options to decrease the computational time of the scalable C-TMLE algorithms. The “Pre-Ordered” search strategy has an optional argument k which defaults to 1. At each step, the next k available ordered covariates are added to the model used to estimate g_0. Large k can speed up the procedure when there are many covariates. However, this approach is prone to over-fitting, and may miss the optimal solution.

An early stopping criteria that avoids computing and cross-validating the complete model containing all p covariates can also save unnecessary computations. A “patience” argument accelerates the training phase by setting the number of steps to carry out after having found a local optimum. To prepare Section 7.1, argument “patience” was set to 10. More details are provided in that section.

a. Well-specified model for $\overline{Q_0}$.

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

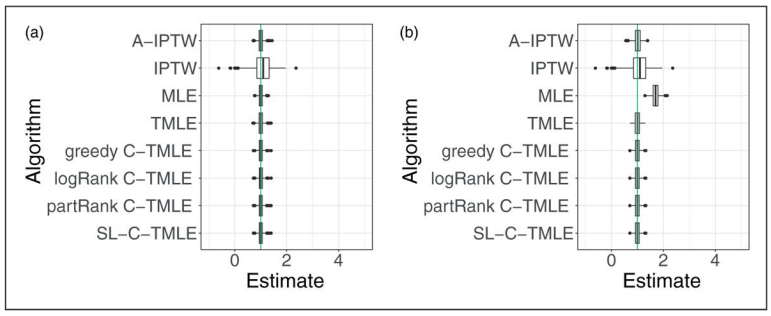

Figure 1.

Simulation 1: Box plot of the ATE estimates with well/mis-specified models for $\bar{Q}_0$. The green lines indicate the true parameter value. (a) Well specified model for $\bar{Q}_0$. (b) Mis-specified model for $\bar{Q}_0$.

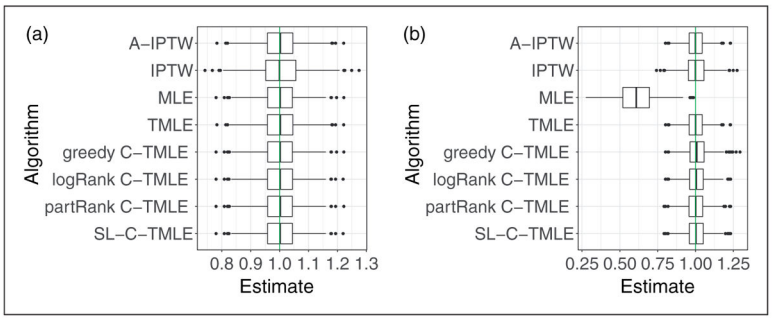

Figure 2.

Simulation 2: Box plot of the ATE estimates with well/mis-specified models for $\bar{Q}_0$. The green line indicates the true parameter value. (a) Well specified model for $\bar{Q}_0$. (b) Mis-specified model for $\bar{Q}_0$.

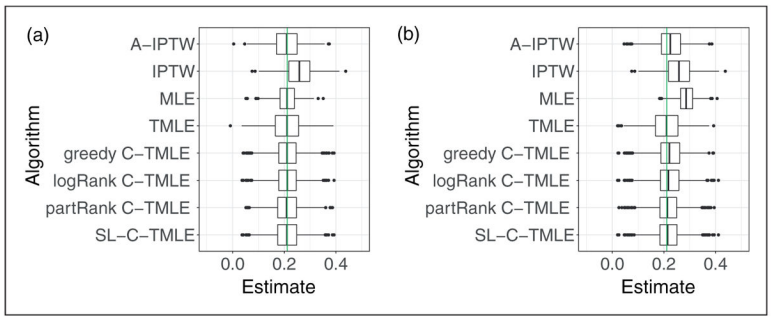

Figure 3.

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

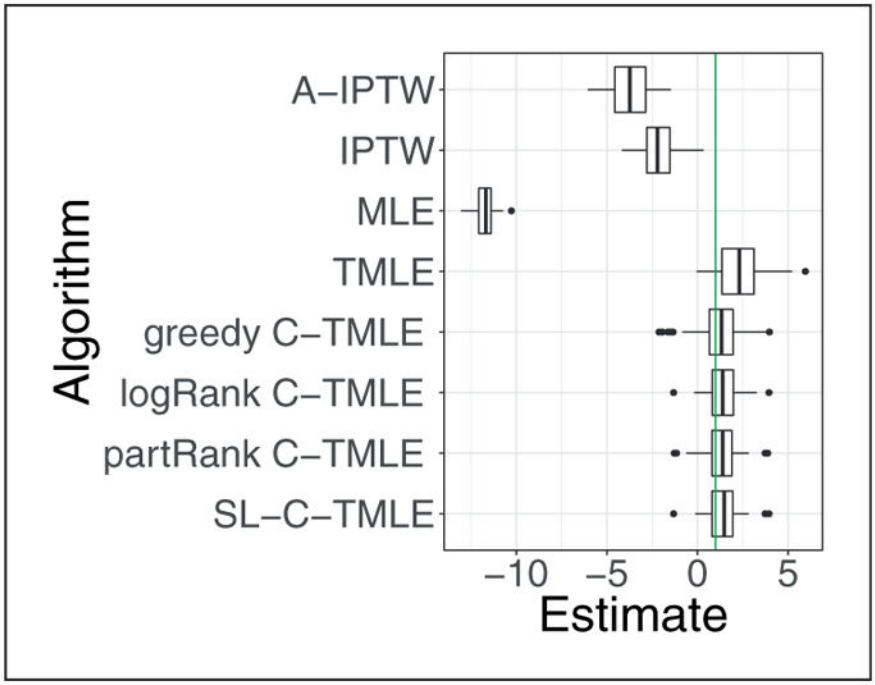

Figure 4.

Simulation 4: Box plot of the ATE estimates with mis-specified model for $\bar{Q}_0$. (a) Median computational time (across 10 replications for each point), with $n = 1,000$ fixed and p varying. (b) Median computational time (across 10 replications for each point), with varying n and fixed $p = 20$.

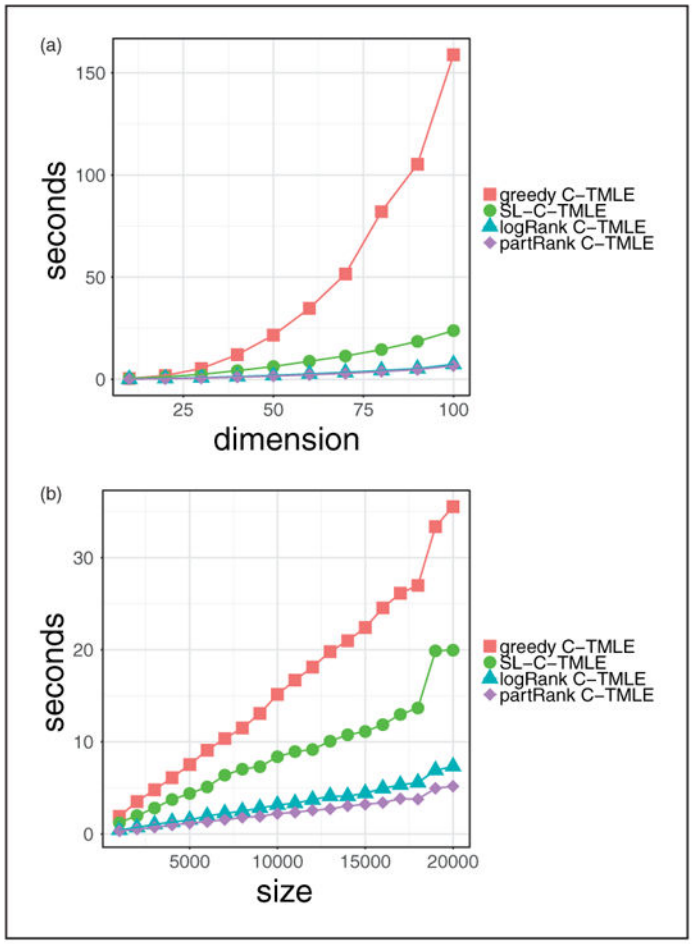

Figure 5.

Computational times of the C-TMLE algorithms with greedy search and pre-ordering. (a) Median computational time (across 10 replications for each point), with n=1, 000 fixed and p varying and (b) Median computational time (across 10 replications for each point), with varying n and fixed p=20.

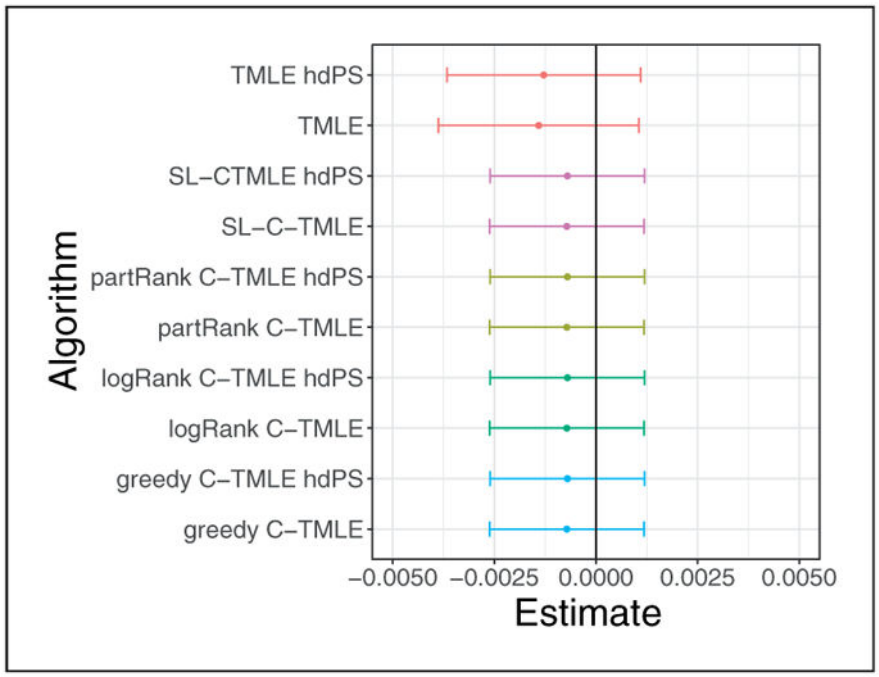

Figure 6.

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

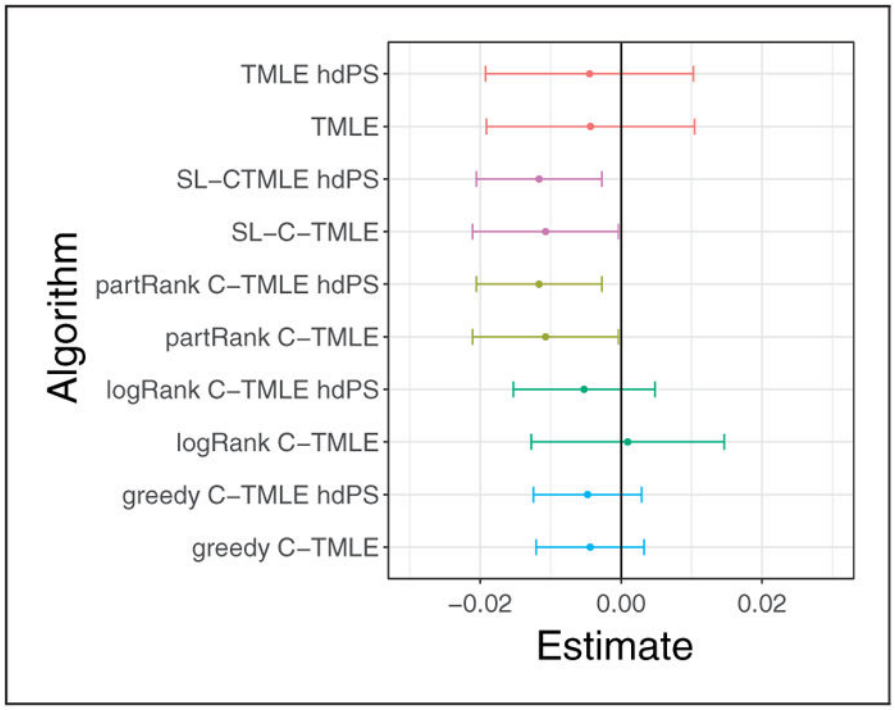

Figure 7.
Point estimates and 95% CIs yielded by the different TMLE and C-TMLEs built on the NOAC data set.

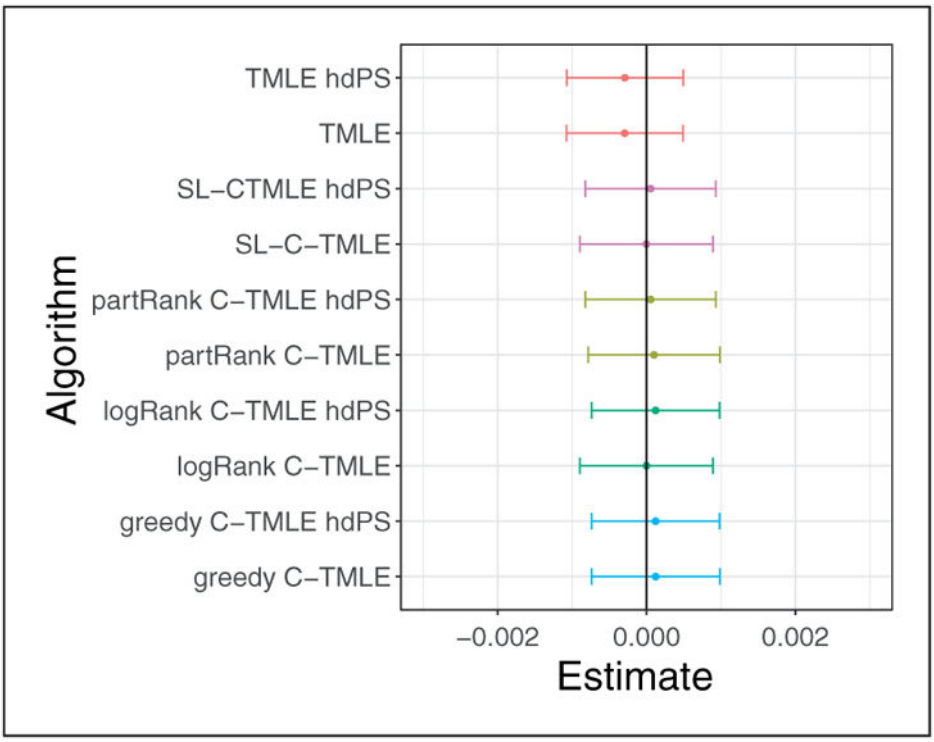

Figure 8.

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

Table 1

Simulation study 1 – performance of the various estimators across 1000 simulated data sets of sample size 1000.

	Well-specified model for $\bar{Q}_0$			Mis-specified model for $\bar{Q}_0$		
	bias (10^{-3})	**se (10^{-2})**	**MSE (10^{-3})**	**bias (10^{-3})**	**se (10^{-2})**	**MSE (10^{-3})**
Unadj	2766.8	22.60	7706.3	2766.8	22.61	7706.3
A-IPTW	0.7	9.54	9.1	10.8	13.52	18.4
IPTW	75.9	34.91	127.5	75.9	34.91	127.5
MLE	1.0	8.20	6.7	699.4	13.96	508.6
TMLE	0.6	9.55	9.1	1.3	11.05	12.2
greedy C-TMLE	0.8	8.91	7.9	0.4	10.41	10.8
logRank C-TMLE	0.1	8.94	8.0	0.4	10.41	10.8
partRank C-TMLE	0.3	8.94	8.0	0.4	10.41	10.8
SL-C-TMLE	0.1	9.07	8.2	0.4	10.41	10.8

Table 2

Simulation study 2 – performance of the various estimators across 1000 simulated data sets of sample size 1000.

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

Table 3

Simulation study 3 – performance of the various estimators across 1000 simulated data sets of sample size 10,000.

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

Table 4

Simulation study 4 – performance of the various estimators across 1000 simulated data sets of sample size 1000.

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

Note: Omitted in the table, the performance of the unadjusted estimator was an order of magnitude worse than the performance of the other estimators.

Table 5

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

	Model for $\bar{Q}_0$	Estimate	CI	Processing time
TMLE	Well specified	0.202	(0.193, 0.212)	0.6s
	Mis-specified	0.203	(0.193, 0.213)	0.6s
C-TMLE, Greedy	Well specified	0.205	(0.196, 0.213)	618.7s
	Mis-specified	0.214	(0.205, 0.223)	1101.2s
C-TMLE, logistic ordering	Well specified	0.205	(0.196, 0.213)	57.4s
	Mis-specified	0.211	(0.202, 0.219)	125.6s
C-TMLE, partial correlation ordering	Well specified	0.205	(0.197, 0.213)	22.5s
	Mis-specified	0.211	(0.202, 0.219)	149.0s
SL-C-TMLE	Well specified	0.205	(0.197, 0.213)	69.8s
	Mis-specified	0.211	(0.202, 0.219)	264.3s