Blip Variance, A Measure of Treatment Precision

Jonathan Levy, Mark van der Laan, Alan Hubbard, Romain Pirracchio

August 28, 2017

Abstract

In this paper we offer an efficient, non-parametric way to assess treatment reliability via the blip function, $B(W)$, the average treatment effect for a randomly drawn strata, W. We can ask the two main questions of any function of a random variable: What are its mean and variance? The mean gives the more easily estimable average treatment effect where as blip variance measures reliability of treatment or the extent of effect modification. The square root of blip variance places it on the scale of the average treatment effect and thus gives a good measure of how much precision in treatment can be gained in assigning treatments based on the covariates. In short, blip variance provides us with a measure of clinical treatment effect heterogeneity which, is useful for doctors as to the reliability of a static intervention. Through extensive simulations we will verify some of the theoretical properties of the estimator established in this paper, as well as demonstrate potentially huge advantages provided by the targeted learning (Laan and Rose 2011) framework in this estimation problem. We also offer a brief demonstration on a publicly available data set, where we employ newly developed software to perform the estimation. We also offer on https://github.com/jlstiles/Simulations easy instructions on how to reproduce all the results presented in this paper, organized by section.

Introduction and layout of this paper

This paper is on part of a github repository available at https://github.com/jlstiles/Simulations, where the reader can easily download an R package including the paper and all functions used in the simulations and analysis along with a readme file with very simple and easy instructions, organized by section, on how to reproduce the results here in. In section 1 we will give a basic motivation and background for estimating this parameter and set up the estimation problem in section 2. Section 3 will cover the estimation methodology and the exact one-step TMLE algorithm (Laan and Gruber 2016) employed in this paper. In section 3 we will also cover ensemble learning and CV-TMLE (Zheng and Laan 2010). Section 4 will verify the theoretical properties of our estimator through two types of simulations. Particularly we will show the virtues of ensemble learning, which will include the highly adaptive lasso thus guaranteeing an asymptotically efficient non-parametric estimator (Laan 2016) for the accompanying TMLE. We then provide an example on a publicly available data set, followed by concluding remarks as to where estimating the blip variance fits in the broader picture of analyzing heterogeneous effects as well as the unique difficulties in estimating this particular parameter. The Appendix will contain a derivation of the efficient influence curve for blip variance as well as computation of second order remainder terms after going through some basic efficiency theory to set up the proofs. The Appendix includes a derivation for how to obtain inference based on logistic regression plug-in estimates which, is a useful tool for comparison with TMLE employed in this paper. We offer a new R package, gentmle, (Coyle and Levy 2017), included in the package, Simulations in the github repository, that performs the targeting portion of the estimating procedure to be detailed later.

keywords: Blip Function, Condition Average Treatment Effect, Cross-validation, Efficient Influence Curve, Ensemble Learning, Machine Learning, Oracle Inequality, Simultaneous Confidence Bounds, SuperLearner, Targeted minimum loss estimation, Targeted Learning, TMLE, gentmle.

1 Background and Motivation

We shall define, Y_a as a random variable which, is a counterfactual outcome for a population under the intervention to set treatment to a as per the Neyman-Rubin causal model (Pearl 2000). The random variable $Y_1 - Y_0$ takes the values -1, 0 or 1. In this paper we consider the average and variance of the related **blip function**, $\mathbb{E}[(Y_1 - Y_0)|W]$. The parameter, $var(Y_1 - Y_0)$ is not identifiable from the observed data with confounders. In the case of a randomized clinical trial Ding, Feller and Miratrix (Ding et al. 2016), construct a Fisher randomization test as to whether $var(Y_1 - Y_0) = 0$ which, is formulated as assuming a null hypothesis $Y_1 = Y_0 + \tau$, where τ is the nuisance paramter, average treatment effect. The method centers around computing a maximum p-value across possible values of τ. This hypothesis test has desirable qualities of not relying on asymptotics or any modeling assumptions beyond the fact treatment is randomly assigned.

An outgrowth of the approach is to extend the hypothesis test to the case of a stratified randomized experiment to test whether $var(Y_1 - Y_0) = 0$ over all pre-specified strata. As noted in (Bitler et al. 2014) establishing rough mean effects subgroups can miss detecting treatment effect heterogeneity for often such subgroups are too rough. Additionally, rejecting the 0 null hypothesis on data set is not necessarily helpful for a clinician who assigns treatment based on confounders. In this paper we aim to develop a one-step targeted maximum likelihood estimator (Laan and Gruber 2016) for estimating the variance of the blip function, $var(\mathbb{E}[Y_1 - Y_0]|W)$ as well as the causal risk difference (mean of the blip function), giving individual or simultaneous confidence intervals that cover with a specified significance level. This gives a clinician a useful measure of what a patient can expect from treatment in addition to the average effect.

2 Defining the Estimation Problem

2.1 Full Data Statistical Model and the link to the Observed Data

Our full data, including unobserved measures, is assumed to be generated according to the following structural equations. We can assume a joint distribution, $U = (U_W, U_A, U_Y) \sim P_U$, an unknown distribution of unmeasured variables. In the time ordering of occurrence,$W = f_W(U_W)$ where W is a vector of confounders, $A = f_A(U_A, W)$ where A is a binary treatment, $Y = f_Y(U_Y, W, A)$ where Y is the outcome, either binary or continuous. This defines a distribution, $P_{U,X}$ where $(U, X) = (U_W, U_A, U_Y, W, A, Y) \sim P_{U,X}$.

Y_a is a random outcome under $P_{U,X}$ where we intervene on the structural equations to set treatment to a. ie, $Y_a = f_Y(U_Y, a, W)$. The full model, $\mathcal{M}^F$ consists of all possible P_{UX}, which can be considered nonparametric. We could also have some knowledge about the treatment mechanism but the analysis will remain the same. The observed data model, $\mathcal{M}$, is linked to $\mathcal{M}^F$ in that we observe $X = (W, A, Y) \sim P$ where $X = (W, A, Y)$ is generated by P_{UX} according to the structural equations above. Our true observed data distribution, P_0 is an element of $\mathcal{M}$. Since we assume a nonparametric full model, $\mathcal{M}$ is also nonparametric.

2.1.1 Parameter of Interest and Identification

We define the blip function as $B_{P_{UX}}(W) = \mathbb{E}_{P_{UX}}[Y_1|W] - \mathbb{E}_{P_{UX}}[Y_0|W]$,where we specify the distribution P_{UX} to mean the full-data model blip function. Our parameter of interest is a mapping from $\mathcal{M}^F$ to R^2 defined by $\Psi^F(P_{UX}) = (\mathbb{E}_{P_{UX}} B_{P_{UX}}, var_{P_{UX}} B_{P_{UX}})$. This two dimensional parameter gives the causal risk difference and a measure of how much the treatment effect varies from strata to strata. Too high a variance for say, a treatment beneficial on average, would be cause for developing more precision in the treatment.

We will impose the randomization assumption on our full data model: $Y_a \perp A|W$, thus giving us $\mathbb{E}_{P_{UX}}[Y_a|W] = \mathbb{E}_P[Y|A = a, W]$. Define $B_P(W) = \mathbb{E}_P[Y|A = 1, W] - \mathbb{E}_P[Y|A = 0, W]$ and we see $B_{P_{UX}}(W) = B_P(W)$ and we can identify the parameter of interest as a mapping from the observed data model, $\mathcal{M}$ to $\mathbb{R}^2$ via $\Psi(P) = (\mathbb{E}_P B_P, var_P B_P) = \Psi(P^F_{UX})$

3 Estimation Methodology

We refer the reader to Targeted Learning Appendix (Laan and Rose 2011) as well as (Laan 2016; Laan and Gruber 2016; Laan and Rubin 2006) for a more detailed look at the theory of TMLE and the use of targeted learning that comprises our general technique. The efficient influence curve at a distribution, P, for the parameter mapping, Ψ, is a function of the observed data, $O \sim P$, notated as $D^\star_\Psi(P)(O)$. The efficient influence curve is the central object in the methodology for its variance gives the generalized Cramer-Rao lower bound for the variance of any regular asymptotically linear estimator of Ψ (Vaart 2000). When Ψ is understood we will leave it out of the notation. We also note, in our general discussion, we consider a d-dimensional parameter and corresponding d-dimensional efficient influence curve, $D^\star(P) = (D^\star_1(P), ..., D^\star_d(P))$.

3.1 TMLE Conditions and Efficiency

We will employ the notation, $P_n f$ to be the empirical average of function, f, and Pf to be $\mathbb{E}_P f$. We can note the following 2^{nd} order expansion, $\Psi(P) - \Psi(P_0) = (P - P_0)D^\star(P) + R_2(P, P_0)$. Now, define a valid loss function, $L(P)(O)$, which is a function of the observed data, O, and indexed at the distribution on which it is defined, P. The reader may consider log-likelihood as the standard example, for a binary outcome, which is also valid for a continuous outcome scaled between 0 and 1. TMLE maps an initial estimate, $P_n^0 \in \mathcal{M}$, of the true data generating distribution, $P_0 \in \mathcal{M}$, to $P_n^\star \in \mathcal{M}$ such that $P_n L(P_n^\star) \leq P_n L(P_n^0)$ and such that $P_n D^\star(P_n^\star) = 0_{d\times 1}$. Now the second order expansion becomes $\Psi(P_n^\star) - \Psi(P_0) = (P_n - P_0)D^\star(P_n^\star) + R_2(P_n^\star, P_0)$.

3.1.1 Asymptotic Efficiency of TMLE

Define the usual norm $\|f\|_{L^2(P)} = \sqrt{\mathbb{E}_P f^2}$. Assume the following:

1. $D_j^\star(P_n^\star)$ is in a P-Donsker class for all j. This condition can be dropped in the case of using CV-TMLE (Zheng and Laan 2010). We show the advantages to CV-TMLE in our simulations.

2. $R_{2,j}$ is $o_p(1/\sqrt{n})$ for all j.

3. $D_j^\star(P_n^\star) \overset{L^2(P_0)}{\longrightarrow} D_j^\star(P_0)$ for all j.

then $\sqrt{n}(\Psi(P_n^\star) - \Psi(P_0)) \overset{D}{\Longrightarrow} N[0, cov(D^\star(P_0)]$. Let $\widehat{\sigma}(\cdot)$ be the sample standard deviation operator. Since the efficient influence curve has variance that is the lower bound for any regular asymptotically linear estimator, this result shows our plug-in TMLE estimates and CI's given by

$$\Psi_j(P_n^\star) \pm z_\alpha * \frac{\widehat{\sigma}(D_j^\star(P_n^\star))}{\sqrt{n}}$$

will each be asymptotically efficient at significance level, $1 - \alpha$, where $Pr(|Z| \leq z_\alpha) = \alpha$ for Z standard normal (Laan and Rubin 2006). However, for multiple testing, we often want to provide confidence intervals that simultaneously cover all the coordinates of $\Psi(P_0)$ at a given significance level. We also note that if the TMLE conditions hold for the initial estimate, P_n^0, then they will hold for the updated model, $P_n^\star$ (Laan 2016).

3.1.2 Simultaneous Confidence Bounds

The following is an added benefit of having the efficient influence curve at hand for we can account for correlated estimates in a tighter manner than a simple bonferroni correction. Consider $Z_n \sim N(0_{d\times 1}, \Sigma_n)$ where Σ_n is the sample correlation matrix of $D^\star(P_n^\star)$. Let $q_{n,\alpha}$ be the α^{th} quantile of $max(|Z_{n,1}|, ..., |Z_{n,d}|)$. Let $Z \sim N(0_{d\times 1}, \Sigma)$ where Σ is the correlation matrix of $D^\star(P_0)$. Let q_α be the α^{th} quantile of $max(|Z_1|, ..., |Z_d|)$, ie the α^{th} quantile of the max number of standard deviations over the coordinates of Z. Then a straightforward application of the continuous mapping theorem (Vaart and Wellner 1996) tells us $q_{n,\alpha} \longrightarrow q_\alpha$. This then implies the confidence intervals given by

$$\Psi_j(P_n^\star) \pm q_{n,\alpha} * \frac{\widehat{\sigma}(D_j^\star(P_n^\star))}{\sqrt{n}}$$

will asymptotically cover all coordinates, Ψ_j of Ψ, simultaneously at the significance level, $1-\alpha$.

3.1.3 TMLE is a Plug-in Estimator

We greatly reduce the dimension of the estimation problem by always using the empirical distribution to estimate the distribution of the confounders, W, and plug in estimates of other relevant parts of the distribution to obtain our parameter estimates. The plug-in quality of TMLE gives it more stable performance in finite samples for it will always respect the natural bounds of the parameter.

3.2 Mapping P_n^0 to $P_n^\star$: The Targeting Step

The preceding section sketched the framework by which TMLE provides asymptotically efficient estimators for nonparametric models. Here we will explain how TMLE maps an initial estimate P_n^0 to $P_n^\star$, otherwise known as the targeting step. P_n^0 is considered to be the initial estimate for the true distribution, P_0. We will consider a binary outcome, Y, or in the case of a continuous outcome, we scale the outcome as $\frac{Y-a}{b-a}$ where the range is $[a, b]$ and then apply the same algorithm.

Definition 3.1. We can define a locally least favorable submodel (lfm) (Laan and Gruber 2016) of an estimate, P_n^0, of the true distribution as

$$\{P_n^\epsilon \text{ s.t } \frac{d}{d\epsilon} P_n L(P_n^\epsilon)\bigg|_{\epsilon=0} = \|P_n D^\star(P_n^0)\|_2, \epsilon \in [-\delta, \delta]\} \tag{1}$$

where $\|\cdot\|_2$ is the euclidean norm and we consider a $d-dimensional$ parameter mapping $\Psi : \mathcal{M} \longrightarrow \mathbb{R}^d$.

This only slightly differs from that offered by Mark van der Laan (Laan and Gruber 2016) in that we can define an lfm with only a single epsilon, which makes the ensuing discussion more fluid. The reader may note this lfm is what is employed in the R package gentmle (Coyle and Levy 2017) when specifying the approach as "line". We can then define the universal least favorable submodel (ulfm) in terms of the lfm if we use the difference equation $P_n(L(P_n^{dt}) - L(P_n^0)) \approx \|P_n D^\star(P_n^0)\|_2 dt$. More generally, if we take $t = m \times dt$ for an arbitrarily small, dt, we obtain $(L(P_n^{t+dt}) - L(P_n^t)) \approx \|P_n D^\star(P_n^t)\|_2 dt$. This recursive process yields the integral equation:
$P_n(L(P_n^\epsilon) - L(P_n^0)) = \int_0^\epsilon \|P_n D^\star(P_n^t)\|_2 dt$. We have thusly just sketched the construction of a universal least favorable submodel, the existence of which was established in Mark van der Laan's work (Laan and Gruber 2016). Again the reader may specify approach as "recursive" to perform a one-step TMLE using the gentmle R package (Coyle and Levy 2017).

Definition 3.2. A Universal Least Favorable Submodel (ulfm) of P_n^0 satisfies

$$\frac{d}{d\epsilon} P_n L(P_n^\epsilon) = \|P_n D^\star(P_n^\epsilon)\|_2 \ \forall \epsilon \in (-\delta, \delta)$$

When the empirical loss is minimized at a given ϵ, we will have solved, $\|P_n D^\star(P_n^\epsilon)\|_2 = 0$. Therefore, the loss is decreased and both influence curve equations are solved simultaneously with a single ϵ in one step. Specifically, $P_n D_j^\star(P_n^\star) = 0$ for all j. Thus $P_n^\star = P_n^\epsilon$ and we have defined the required TMLE mapping.

Remark on the one-step TMLE

We note that the one-step TMLE differs from the previous iterative version (Laan, 2011) in that iteration requires fluctuations that might go too far for each step, particularly in the case when the initial fit, P_n^0, is far from the truth. We therefore might not get a minimal change in the initial fit to reduce the bias, resulting in extra variance. More work is needed to compare the one-step TMLE with the iterative TMLE in finite samples. In our simulations, we found no appreciable difference.

3.3 Targeting Causal Risk Difference and Blip Variance

Since we have defined the universal least favorable submodel directly from the locally least favorable one, all that is left for our particular case, is to define our locally least favorable model. This will then give rise to the algorithm we provide at the end of this section. Consider the parameter mapping $\Psi : \mathcal{M} \longrightarrow \mathbb{R}^2$ where $\Psi(P) = (\mathbb{E}_P B_P(W), var_P(B_P(W)))$ where $B_P = \mathbb{E}_P[Y|A=1,W] - \mathbb{E}_P[Y|A=0,W]$. **The efficient influence curve** for the first component, the causal risk difference, $\Psi_1(P)$ is given by the well-known formula

$$D_1^\star(P) = D_{1,1}^\star(P) + D_{1,2}^\star(P) = \frac{2A-1}{g(A|W)}(Y - \bar{Q}(A,W)) + B_P(W) - \Psi_2(P)$$

where $D_{1,2}^\star(P) = B_P(W) - \Psi_1(P)$. The efficient influence curve for the blip variance, the second component of $\Psi(P)$, is given by

$$D_2^\star(P) = D_{2,1}^\star(P) + D_{2,2}^\star(P) = 2(B_P(W) - \mathbb{E}_P B_P)\frac{2A-1}{g(A|W)}(Y - \bar{Q}(A,W)) + (B_P(W) - \mathbb{E}_P B_P)^2 - \Psi_2(P)$$

derived in the appendix of the this paper. Also note $D_{2,2}^\star(P) = (B_P(W) - \mathbb{E}_P B_P)^2 - \Psi_2(P)$. It is convenient to define $\bar{Q}_0(A,W) \equiv Pr(Y=1|A,W) = \mathbb{E}_{P_0}[Y|A,W]$. We also have for the density, p_0 of distribution, P_0, the factorization.

$$\begin{aligned} p_0(W,A,Y) &= p_Y(Y|A,W)p_A(A|W)p_W(W) \\ &= p_Y(Y|A,W)g_0(A|W)q_W(W) \\ &= \bar{Q}_0(A,W)^Y(1-\bar{Q}_0)(A,W)^{1-Y}g_0(A|W)q_W(W) \end{aligned}$$

$p_{n,Y}^0$ is the initial estimate of the true density p_Y. We estimate the treatment mechanism, g_0, (as in the case of an observational study, for example) with initial estimate g_n^0. TMLE is a plug-in estimator, using the empirical distribution of W, q_n, to estimate p_W. Therefore $q_n D_{j,2}^\star(P_n^\star) = 0 \ \forall j$ and $\|P_n(D^\star(P_n^\star)))\|_2 = \|P_n(D_{1,1}^\star(P_n^\star), D_{2,1}^\star(P_n^\star))\|_2$. Since the parameter mapping does not depend on g we therefore only need to define a locally least favorable submodel of the initial estimate of the outcome density, $p_{n,Y}^0(Y|A,W) = \bar{Q}_n^0(A,W)^Y(1-\bar{Q}_n^0(A,W))^{1-Y}$ (Laan and Rose 2011). If the parameter mapping depended on g, then we would define the submodel in terms of $p_{n,A}^0$ as well, as is the case with say, average treatment effect among the treated (Laan and Gruber 2016) or a stochastic intervention with an unknown treatment mechanism (Muñoz ID 2012). Here, we will fix g_n^0 as g_n and as always, fix q_n. Our initial density estimate is $p_n^0(W,A,Y) = p_{n,Y}^0(Y|A,W)g_n(A|W)q_n(W)$ and we define a submodel, $P_{n,Y}^\epsilon$ of $P_{n,Y}^0$ as follows, using the inner product notation induced by the euclidean norm:

$$\bar{Q}_n^{\epsilon}(A,W) = expit\left(logit(\bar{Q}_n^0(A,W) - \epsilon\left\langle H(A,W), \frac{P_n D^{\star}(P_n^0)}{\|P_n D^{\star}(P_n^0)\|_2}\right\rangle_2\right)$$

Note that $H(A,W) = (H_1(A,W), H_2(A,W)$ has components $H_1(A,W) = \frac{2A-1}{g_n(A|W)}$ and $H_2(A,W) = 2(B_{P_n^0}(W) - P_n B_{P_n^0})\frac{2A-1}{g_n(A|W)}$ .We also remind the reader that $B_P(W)$ is the blip function with respect to the distribution, P. So we can consider the submodel of P_n^0 defined via:

$$p_{n,\epsilon}^0 = \bar{Q}_n^{\epsilon}(A,W)^Y(1-\bar{Q}_n^{\epsilon}(A,W))^{1-Y}g_n(A|W)q_n(W)$$

We can then easily verify $\{P_{n,\epsilon}^0|\epsilon \in [-\delta,\delta]\}$ satisfies definition 5.1 and hence, we have defined the universal least favorable model by definition 5.2. Now we are ready to provide the algorithm to form a one-step TMLE estimate of our two dimensional parameter.

3.4 One-step TMLE Algorithm for Blip Variance and Causal Risk Difference

The previous section just defined an algorithm to fluctuate the initial estimate of the outcome model in such a way that the efficient influence curve equation is solved for both parameters, enabling a plug-in estimator that is asymptotically efficient. We can note this algorithm can be easily generalized to any number of parameters, such as those in a longitudinal marginal structural model or a whole survival curve (Laan, 2015). We can also scale it back to one parameter. Here we will explicitly write out the algorithm for our particular case. We also refer the reader to a new R package, gentmle (Coyle and Levy 2017), which automates this procedure after initial estimates have been obtained.

Initialization

We start a recursive process with $\bar{Q}_n^0$, our initial prediction of the outcome model, found using the data-adaptive ensemble machine learning package, SuperLearner (Polley, 2009). We also use Superlearner to estimate the treatment mechanism, g_0, with g_n, which happens to stay fixed for targeting this parameter of interest. Compute the negative log-likelihood loss under the outcome model, $\bar{Q}_n^0$:

$$P_n L(P_n^0) = -\frac{1}{n}\sum_{i=1}^{n}\left[Y_i \text{log}\bar{Q}_n^0(A_i,W_i) + (1-Y_i)\text{log}(1-\bar{Q}_n^0(A_i,W_i))\right] = L_0 \text{ our starting loss}$$

Compute $H_1^0(A,W) = \frac{2A-1}{g_n(A|W)}$ and $H_2^0(A,W) = 2\left(B_n^0(W) - P_n B_n^0\right)\left(\frac{2A-1}{g_n(A|W)}\right)$ and note H_1 will stay fixed for the entire process in this case. Compute $\|P_n D^\star(P_n^0)\|_2$. In the ensuing algorithm use a tiny positive increment, $d\epsilon$. The increment size should be as small as possible. The authors have found 0.0001 to be small enough so that going smaller makes no difference.

The Targeting Step

step 2: If $|P_n D_j^\star(P_n^m)| < \hat{\sigma}(D_j^\star(P_n^m))/n$ for $j \in \{1,2\}$ then $P_n^\star = P_n^m$ and go to step 4. This insures that we stop the process once the bias is second order. Note, $\hat{\sigma}(\cdot)$ refers to the sample standard deviation operator. Recursions after this occurs are not fruitful. If $|P_n D_j^\star(P_n^m)| > 1/n$, then $m = m+1$ and go to step 3.

step 3

Define the following recursion, using euclidean inner product notation, $\langle \cdot, \cdot \rangle_2$:

$$\bar{Q}_n^m = expit\left(logit(\bar{Q}_n^{m-1}) - d\epsilon\left\langle (H_1^{m-1}(A,W), H_2^{m-1}(A,W)), \frac{P_n(D_1(P_n^{m-1}), D_2(P_n^{m-1}))}{\|P_n(D_1(P_n^{m-1}), D_2(P_n^{m-1}))\|_2}\right\rangle_2\right) \tag{2}$$

where

- $B_n^{m-1} = \bar{Q}_n^{m-1}(1,W) - \bar{Q}_n^{m-1}(0,W)$
- $H_1^{m-1}(A,W) = \left(\frac{2A-1}{g_n(A|W)}\right)$
- $H_2^{m-1}(A,W) = 2\left(B_n^{m-1}(W) - P_n B_n^{m-1}\right)\left(\frac{2A-1}{g_n(A|W)}\right)$

This recursively defines the distribution, P_n^m, via its density:

$$p_n^m = \bar{Q}_n^m(A,W)^Y(1-\bar{Q}_n^m(A,W))^{1-Y} g_n(A|W) q_n(W)$$

Compute $L_m = -P_n\left[Y\log\bar{Q}_n^m + (1-Y)\log(1-\bar{Q}_n^m)\right]$. If $L_m \leq L_{m-1}$ then return to step 2. Otherwise, $P_n^\star = P_n^{m-1}$ and continue to step 4.

step 4

Our estimate is $\hat{\Psi}(P_n) = \Psi(P_n^\star)$ which is really only dependent on $\bar{Q}_n^\star$ and the empirical distribution.

3.5 SuperLearner: Making Initial Predictions $\bar{Q}_n^0$ and g_n

Targeted learning (Laan and Rose 2011) features the use of data adaptive prediction methods optimized by the R package, SuperLearner (Polley, LeDell, et al. 2017) or by other ensemble learning packages, such as H2O (LeDell 2017). SuperLearner initial predictions using V-fold cross-validation, converge to the oracle predictor (minimum risk predictor) from the collection of algorithms at hand (Laan and Dudoit 2003). For our parameter, we wish to give the best fit (minimum loss) for $\bar{Q}(A, W)$ for the expected outcome, Y, given A and W and often, as in the case of an observational study, for the propensity score, $g(A|W)$. The power of targeted learning lies in the fact we only need worry about predicting as well as possible our initial estimates and then TMLE will handle getting the inference right via the sample standard deviation of the efficient influence, which has the unique advantage of being both cost free and valid in the presence of data adaptive methods. We refer the reader to (Laan, Polley, et al. 2007) for more detail on the SuperLearner algorithm.

Superlearner, which picks the best single algorithm in the library, has risk that converges to the oracle selector at rate $O\left(log\left(k(n)\right)/n\right)$ where $k(n)$ is the number of candidate algorithms, under very mild assumptions on the library of estimators (Laan, Polley, et al. 2007). For the optimal convex combination of algorithms that forms the SuperLearner predictor in our simulations and often in practice, $k(n)$ is the number of grid points for the convex combination. This allows the number of grid points to grow at a polynomial rate with respect to sample size up to a constant factor.

How SuperLearner Works in Brief

The following is description of V-fold cross-validation employed by SuperLearner. Here we consider V=10. For each training set T used in our CV-TMLE (9/10 of the data) we split it into 10 folds, each consisting of a training set (9/10 of T) and validation set (the remaining 1/10 of T). The 10 disjoint validation sets comprise the entire set, T, so we end up with a prediction for each algorithm over the entire set, T. Then a linear regression with each of these predictions as covariates, is fit to minimize the loss (non negative least squares) with the constraint that all the coefficients sum to 1 and are greater than 0. SuperLearner tables in this paper give the coefficients given to each algorithm, averaged over all 10 folds employed in the CV-TMLE (Zheng and Laan 2010) in section 7 or averaged over all of the simulations.

3.6 CV-TMLE recommendation

Up to this point we have outlined the TMLE algorithm performed on initial estimates of the treatment mechanism and outcome model. The donsker condition in section 3.1.1 can be avoided if we use a CV-TMLE (Zheng and Laan 2010). The donsker condition depends on the initial estimates (Laan 2016) if we use TMLE and not cv-TMLE, it is important SuperLearner initial estimates do not defy this condition. If overly adaptive learners are placed in the SuperLearner library, such as Ranger (Wright and Ziegler 2017), SuperLearner can overfit. However, with cv-TMLE, the SuperLearner predictions are all done on V (normally 10) validation folds so the targeting step is not performed on "inbred" data or data from which the initial predictions were fit. Therefore, if we satisfy the other conditions in 3.1.1, we obtain efficient normally distributed sampling distributions.

Fully nested CV-TMLE takes ten times as long to run as TMLE but is a useful insurance against wild results. We provide in case 2b below, a simulation that shows the advantage of CV-TMLE. The reader may visit https://github.com/jlstiles/Simulations where a readme file is displayed with simple instructions on how to run CV-TMLE as performed in section 7.

4 Simulations

We performed two different kinds of simulations, the first primarily to verify the remainder conditions in the theory of TMLE (condition 2, section 3.1.1). The rest were more standard simulations to mimic what happens with real data. The big question for non-parametric estimation in our context is, we might throw the world of clever prediction at a problem but then how do we get inference or a measure of uncertainty, especially when non-parametrically bootstrapping is expensive or, more crucially, not theoretically valid? Targeted learning provides the answer.

references for table and chart making

The authors would like to express gratitude to Mark Hlavac creator of the stargazer R package (Hlavac 2015) for ease in making tables and Hadley Wickham for ease in creating plots here-in (Wickham 2009).

Inference

Inference for all TMLE's used the sample standard deviation efficient influence curve approximation to form confidence intervals as per section 3. For logistic regression plug-in estimators of blip variance, named ”init ests LR” in the simulations, confidence bands were formed by using the delta method and the influence curve for the beta coefficients for intercept, main terms and interactions (see appendix B for the derivation and https://github.com/jlstiles/Simulations for running the code to compute these confidence intervals). SuperLearner initial estimates had no accompanying measure of uncertainty since there is little theory for such because SuperLearner uses a combination of different data adaptive estimators. Such underscores the importance of why we need the targeting step or TMLE theory.

Blip variance TMLE as opposed to doubly robust TMLE for ATE

TMLE enjoys doubly robust estimation for average treatment effect but such is not true for blip variance. For a doubly robust estimator for say, average treatment effect, precisely we need the product of L^2 covergence rates of treatment mechanism estimates and outcome model predictions to be $o(n^{-0.5})$. For estimating the blip variance, there is no such doubly robust estimator as both the treatment mechanism and the outcome model need to be specified at an $n^{0.25}$ rate (see Appendix A). Thus the clear difference between the TMLE for blip variance and the TMLE for average treatment effect, assuming the treatment mechanism is either known or correctly specified at L^2 rate of $n^{-0.5}$, is for blip variance, we would still need to estimate the outcome model at L^2 covergence rate of $n^{-0.25}$, particularly the pesky $\mathbb{E}_0(B_n(W) - B_0(W))^2$ remainder term is our concern here since we need to estimate the blip function at L_2 convergence rate of $n^{-0.25}$ rate (see Appendix A).

Use of One-Step TMLE vs Iterative

We note to the reader, that all TMLE sampling distributions for blip variance, whether employing a simultaneous TMLE estimator for blip variance and average treatment effect or whether estimating blip variance separately, had near identical sampling distributions. Also, whether we used the one-step TMLE or iterative did not affect the sampling distributions in any appreciable way.

SuperLearner Details

We refer the reader to the online supplementary materials for more complete details of the SuperLearner results. It is notable that any SuperLearner library containing highly adaptive lasso (Laan 2016) will yield asymptotically efficient estimates for TMLE. However, in finite samples we want a variety of algorithms to pick up the truth in different parts of the covariate space.

SuperLearner Library 1, termed SL1, Avoiding Overfitting

This library will be indicated by ”SL1” in the simulation results.

1. SL.gam3 a gam (Hastie 2017) using degree 3 smoothing splines, screening main terms, top 10 correlated variables with the outcome and top 6.

2. SL.glmnet_1, SL.glmnet_2 and SL.glmnet_3 (Friedman et al. 2010) performed a lasso, equal mix between lasso and ridge penalty and ridge regressions.

3. nnetMain_screen.Main (Venbles and Ripley 2002) is a neural network with decay = 0.1 and size = 5 using main terms.

4. earthMain (Milborrow 2017) is data adaptive penalized regression spline fitting method. They allow for capturing the subtlety of the true functional form. We allowed degree = 2, which is interaction terms with the default penalty = 3 and a minspan = 10 (minimum observations between knots).

5. SL.glm (R Core Team 2017) logistic regression and we used main terms, top 6 correlated variables with outcome and top 10 as well as a standard glm with main terms and interactions (glm_mainint_screen.Main)

6. SL.stepAIC (R Core Team 2017) uses Akaike criterion in forward and backward step regression

7. SL.hal is the highly adaptive lasso (Benkeser and Laan 2016), which guarantees the necessary L_2 rates of convergence and therefore, if included in the SuperLearner library, guarantees asymptotic efficiency (ref Mark). hal output is guaranteed to be of finite sectional variation norm and thus is guaranteed to be donsker or not overfit the data.

8. SL.mean returns the mean outcome for assurance against overfitting

9. rpartPrune (Therneau et al. 2017) is recursive partitioning with cp = 0.001 (must decrease the loss by this factor) minsplit = 5 (min observations to make a split), minbucket = 5 (min elements in a terminal node)

SuperLearner Library 2 termed SL2, More Aggressive, overfits a little

This library will be indicated by "SL2" in the simulation results. This library is identical to Library 1, except we added the following learners, which were tuned to maximize cross validated loss on a few draws from case 2a data generating distribution. Thus these additions do not severely overfit, in general.

1. SL.ranger (Wright and Ziegler 2017) mtry = 3, num.trees=2500 and the minimum leaf size was set to 10.

2. SL.xgboost (Chen et al. 2017) xgbFull had max_depth=1, minobspernode=3,shrinkage=.001,ntrees=10000. xgbMain used a depth of 4 on main terms only with same shrinkage and minobspernode but only 2500 trees.

SuperLearner Library Used for Treatment Mechanism

Whenever we mis-specify the treatment mechanism (cases 3 and 4) we used the following library to recover the true treatment mechanism: nnetMain (Venbles and Ripley 2002)(see above), SL.mean (see above), SL.hal (Benkeser and Laan 2016) , SL.glm (R Core Team 2017), SL.glm.interaction (R Core Team 2017), SL.step.interaction (R Core Team 2017) and SL.earth (Milborrow 2017), which uses default settings, allows interactions and calculates the minspan internally, as per Friedman's MARS paper section 3.8 with alpha = 0.05.

4.1 Simulations with Controlled Noise

TMLE enjoys doubly robust estimation for average treatment effect but such is not true for blip variance. For a doubly robust estimator for say, average treatment effect, precisely we need the product of L^2 covergence rates of treatment mechanism estimates and outcome model predictions to be $o(n^{-0.5})$. For estimating the blip variance, there is no such doubly robust estimator as both the treatment mechanism and the outcome model need to be specified at an $n^{0.25}$ rate (see Appendix A). Thus the clear difference between the tmle for blip variance and the tmle for average treatment effect, assuming the treatment mechanism is either known or correctly specified at L^2 rate of $n^{-0.5}$, is for blip variance we would still need to estimate the outcome model at L^2 covergence rate of $n^{-0.25}$. The pesky $\mathbb{E}_0(B_n(W) - B_0(W))^2$ remainder term does not go away without a fight so that is our main concern beyond the norm.

Instead of drawing W then A and then Y under a data generating distribution and then trying to recover the truth with various predictors or SuperLearner as we do later, we directly add heteroskedastic noise to $\bar{Q}_0$ in such a way that the conditions of TMLE hold and then use the noisy estimate as the initial estimate in the TMLE process. This does not necessarily match what happens in practice because the noise we add is not related to the noise in the draw of Y given A and W. However, it is a valid way to directly test the conditions of TMLE in that we can control the noise so that the TMLE conditions hold and watch the asymptotics at play. We also note that we will assume g_0 is known as the other second order terms for blip variance, involving bias in estimating g_0, are dependent on double robustness in the same way as for the average treatment effect, for which the properties of tmle are already well-known (Laan and Rose 2011).

4.1.1 Drawing covariates

$W_1 \sim uniform[-3, 3]$, $W_2 \sim binomial(1, .5)$, $W_3 \sim N[0, 1]$ and $W_4 \sim N[0, 1]$

4.1.2 The truth

1. We define $g_0(A|W) = expit(.5 * (-0.8 * W1 + 0.39 * W2 + 0.08 * W3 - 0.12 * W4 - 0.15))$, which is the true density of A given W.

2. $\mathbb{E}_0[Y|A, W] = \bar{Q}_0(A, W) = expit(.2 * (.1 * A + 2 * A * W1 - 10 * A * W2 + 3 * A * W3 + W1 + W2 + .4 * W3 + .3 * W4))\}$which defines the density of Y given A and W for a binary outcome.

3. defining the blip as $B(W) = \mathbb{E}_0[Y|A = 1, W] - \mathbb{E}_0[Y|A = 0, W]$ we have $\Psi(P_0) = var_0(B(W)) = 0.06356135$. This is a substantial blip variance to avoid getting near the parameter boundary at 0.

Below we illustrate the process for one simulation. For each sample size, n, we performed the simulation 1000 times.

1. define $bias(A, W, n) = 1.5n^{rate}(-.2 + 1.5A + 0.2W_1 + W_2 - AW_3 + W_4)$

2. define heteroskedasticity: $\sigma(A,W,n) = 0.8n^{rate}|3.5 + 0.5W_1 + 0.15W_2 + 0.33W_3W_4 - W_4|$

3. define $b(A,W,n,Z) = bias(A,W,n) + Z \times \sigma(A,W,n)$ where Z is standard normal

4. draw $\{Z_i\}_{i=1}^n$ and $\{X_i\}_{i=1}^n$ each from standard normals

5. $\bar{Q}_n^0(1,W_i) = expit\left(logit\left(\bar{Q}_0(1,W_i)\right) + b(1,W_i,n,Z_i)\right)$

6. $\bar{Q}_n^0(0,W_i) = expit\left(logit\left(\bar{Q}_0(0,W_i)\right) + 0.5b(1,W_i,n,Z_i) + \sqrt{0.75}b(0,W_i,n,X_i)\right)$

7. $\bar{Q}_n^0(A,W) = A * \bar{Q}_n^0(1,W) + (1-A)\bar{Q}_n^0(0,W)$

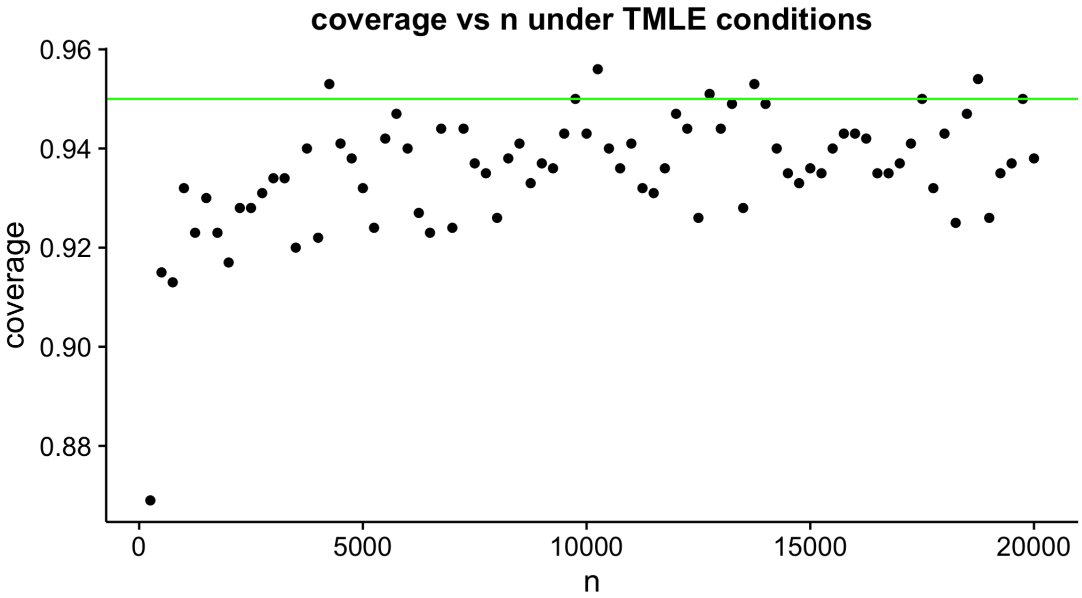
We note that we placed correlated noise on the true $\bar{Q}_0(1,W)$ and $\bar{Q}_0(0,W)$ so as to make the blip "estimates" of similar noise variance as the initial "estimates" for $\bar{Q}_0(A,W)$.

Satisfying TMLE conditions

By a Taylor series expansion about the truth, it is easy to see the above procedure will satisify the remainder term conditions of 3.1.1 if the *rate* is less than -0.25. We have that $\bar{Q}_n^0(1,W) = \bar{Q}_0(1,W) + \bar{Q}_0(1,W)(1 - \bar{Q}_0(1,W))b(1,W,n,Z) + O(b^2(1,W,n,Z))$ and likewise for $\bar{Q}_n^0(0,W)$ and thus trivially, $\sqrt{\mathbb{E}_0\left(B_n^0(W) - B_0(W)\right)^2}$ is of order n^{rate}. As previously mentioned, we need not worry about any second order terms but $\mathbb{E}_0\left(B_n^0(W) - B_0(W)\right)^2$ because we are using the true g_0. Condition 1 is easily satisfied. Condition 3, the donsker condition, is satisfied since our "estimated" influence curve, $D^*(\bar{Q}_n^0, g_0)$, depends on a fixed function of A and W with the addition of independently added random normal noise. Thus we are drawing values from a fixed function with random noise already added to it.

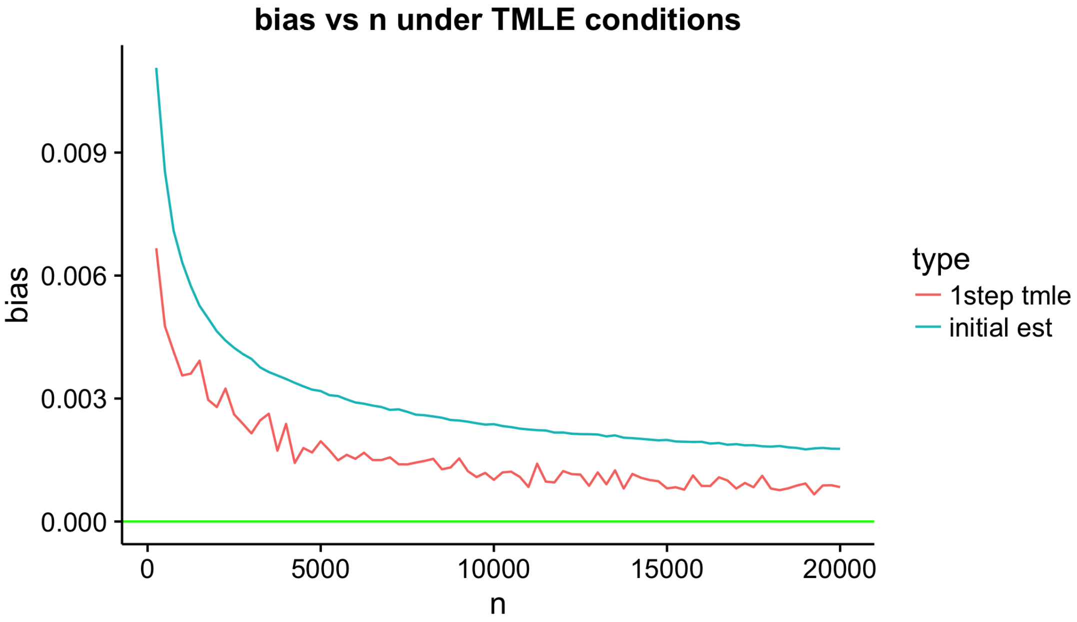
The simulation results below are in alignment with the theory established for the tmle estimator of blip variance.

Figure 1



blip bias has L2 norm = O_p(n^{-1/3}), o_p(n^{-.25}).
n increases by 250 for each point moving to the right, starting at n=250

Figure 2



TMLE performs debiasing at all sample sizes

Figure 3

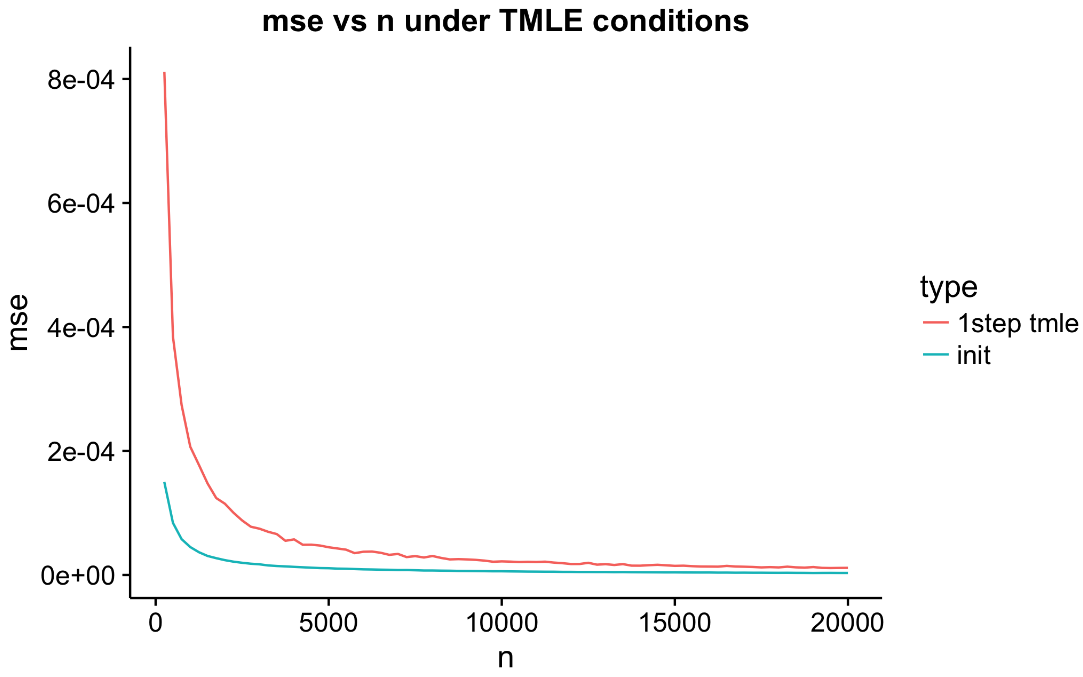


Figure 4

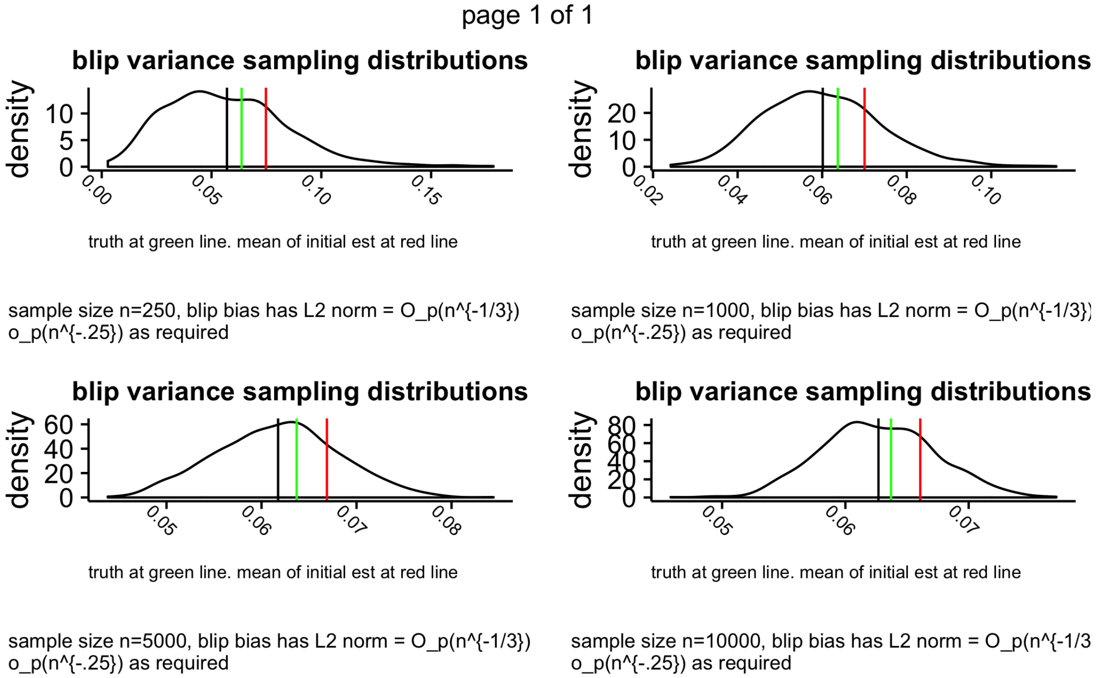
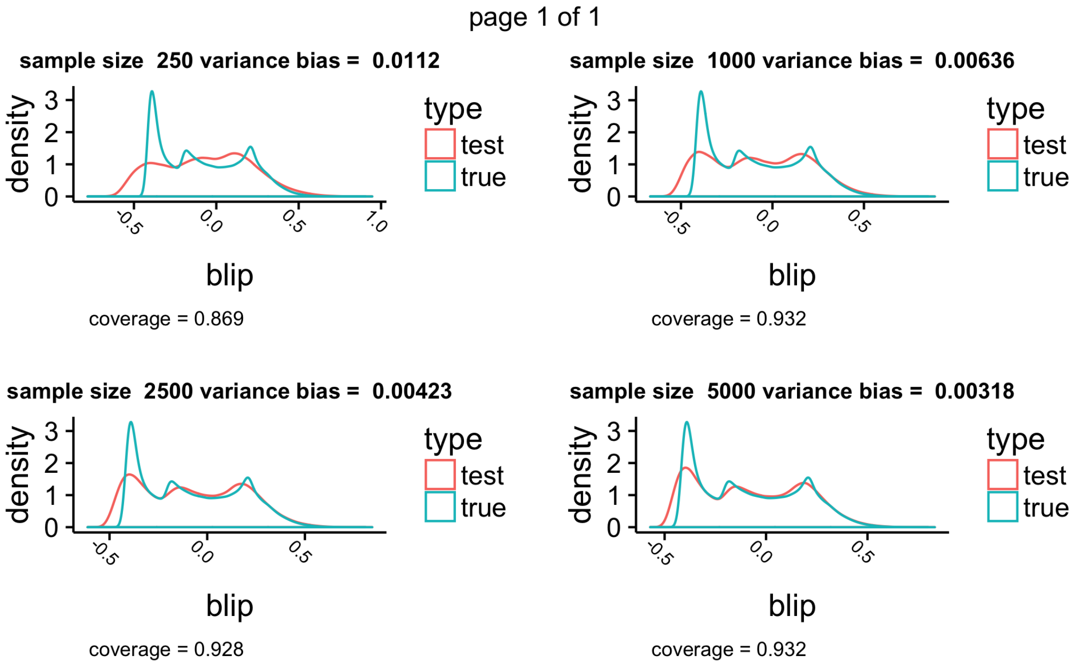


Figure 5

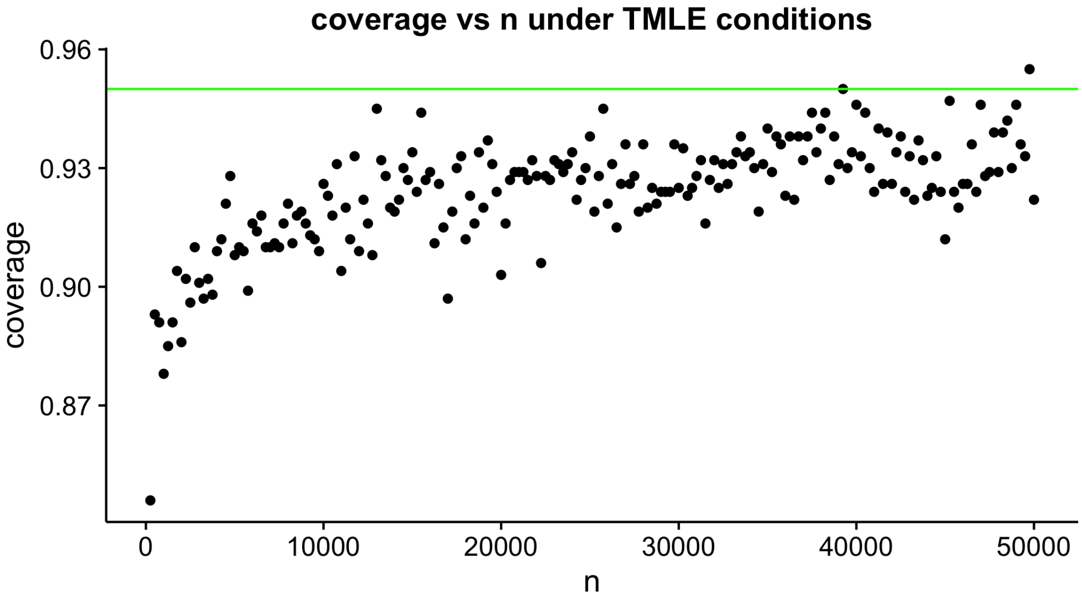


4.1.3 Different DGP with $var_n - var_0 < 0$

The only thing changed here is the bias function, which manufactured negatively biased initial estimates: $bias(A, W, n) = -n^{rate}(.2 + 1.5A + 0.2W_1 + W_2 - AW_3^2 + W_4)$

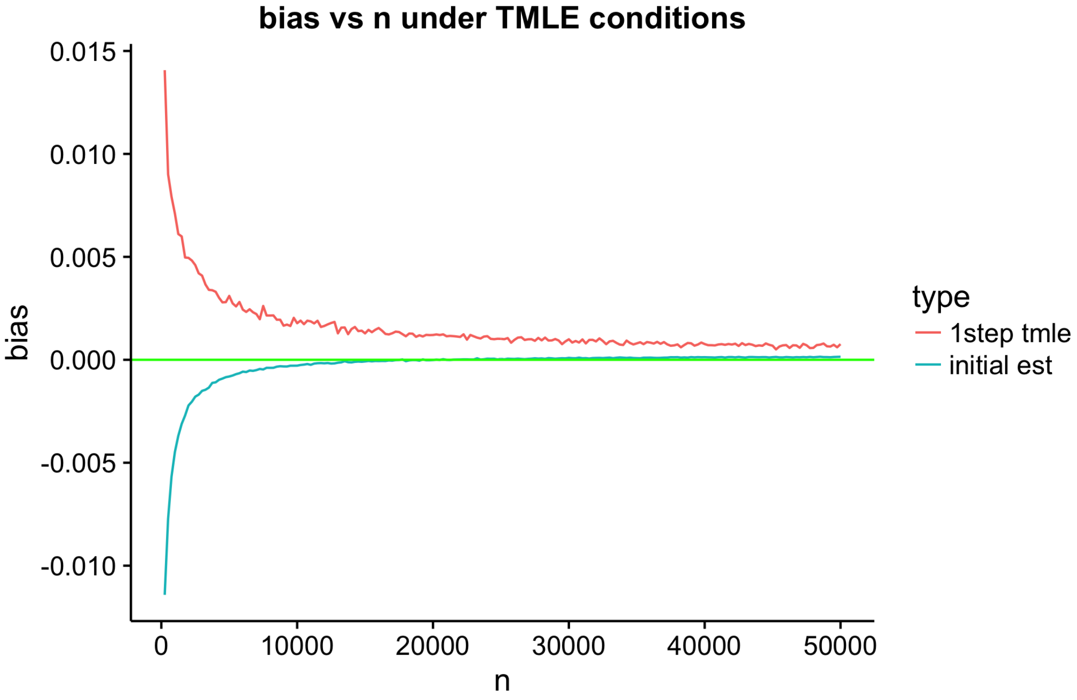
TMLE seemed to make the initial underestimation a little worse in terms of the bias because with tmle, the bias is related to the second order remainder term, which, although is second order, might induce bias.

Figure 6



blip bias has L2 norm = O_p(n^{-1/3}), o_p(n^{-.25}).
n increases by 250 for each point moving to the right, starting at n=250

Figure 7



TMLE does not necessarily debias if 2nd order term is bigger than the bias of initial

Figure 8

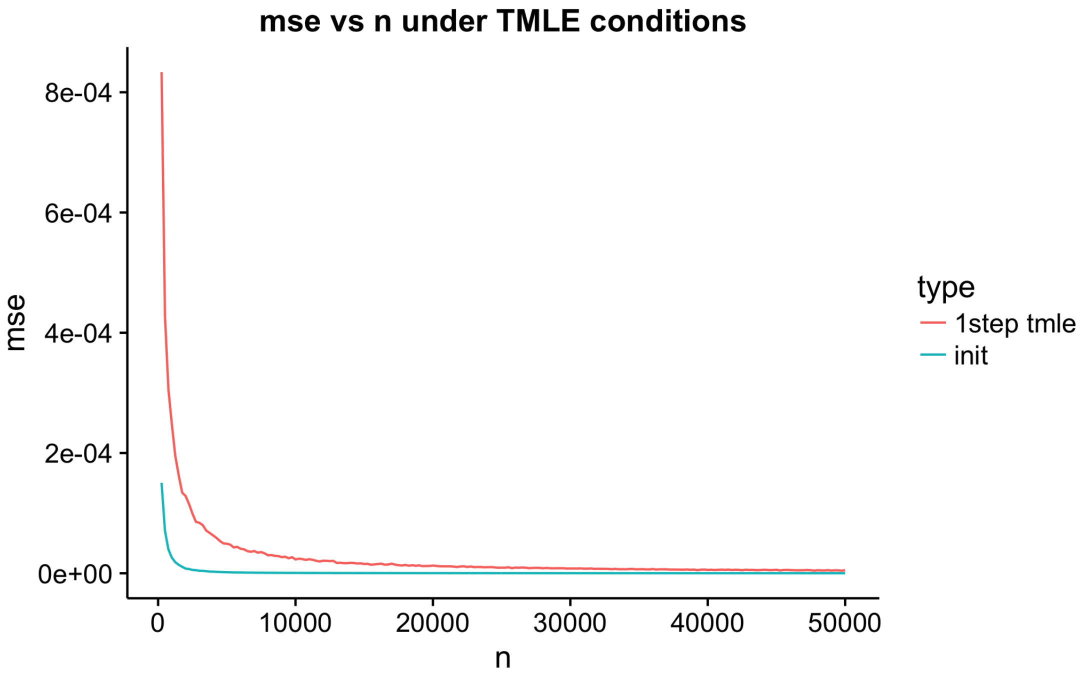


Figure 9

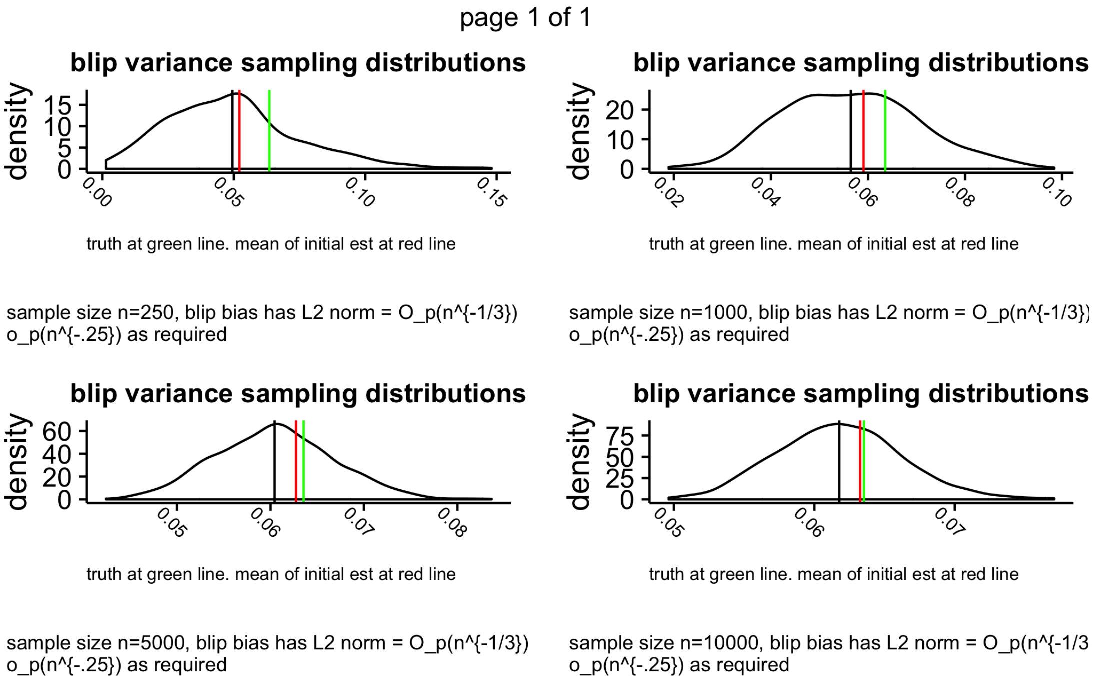
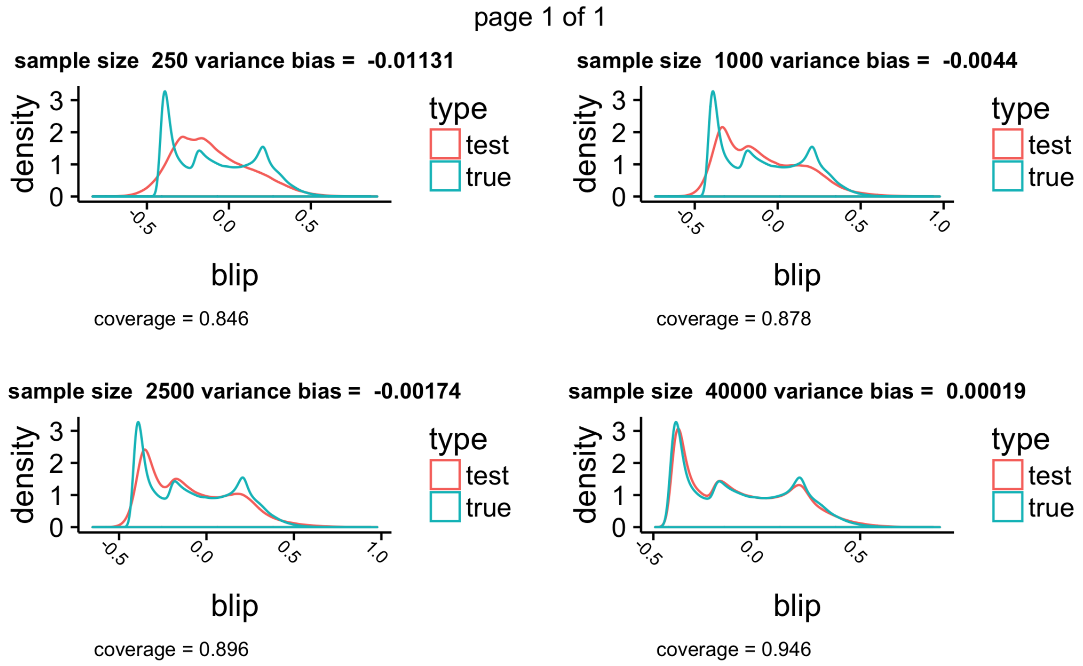


Figure 10



4.2 Simulations to mimic "real" scenarios

We will stick with binary outcome and treatment, though the results will be the same for continuous outcome.

Unless otherwise noted, sample size n = 1000 and the number of simulations = 1000

Generating Covariates, W:

Throughout the simulations we generated the covariates as follows: W_1 = n uniform[-3,3], W_2 = n random normals, W_3 = n random normals, W_4 = n random normals

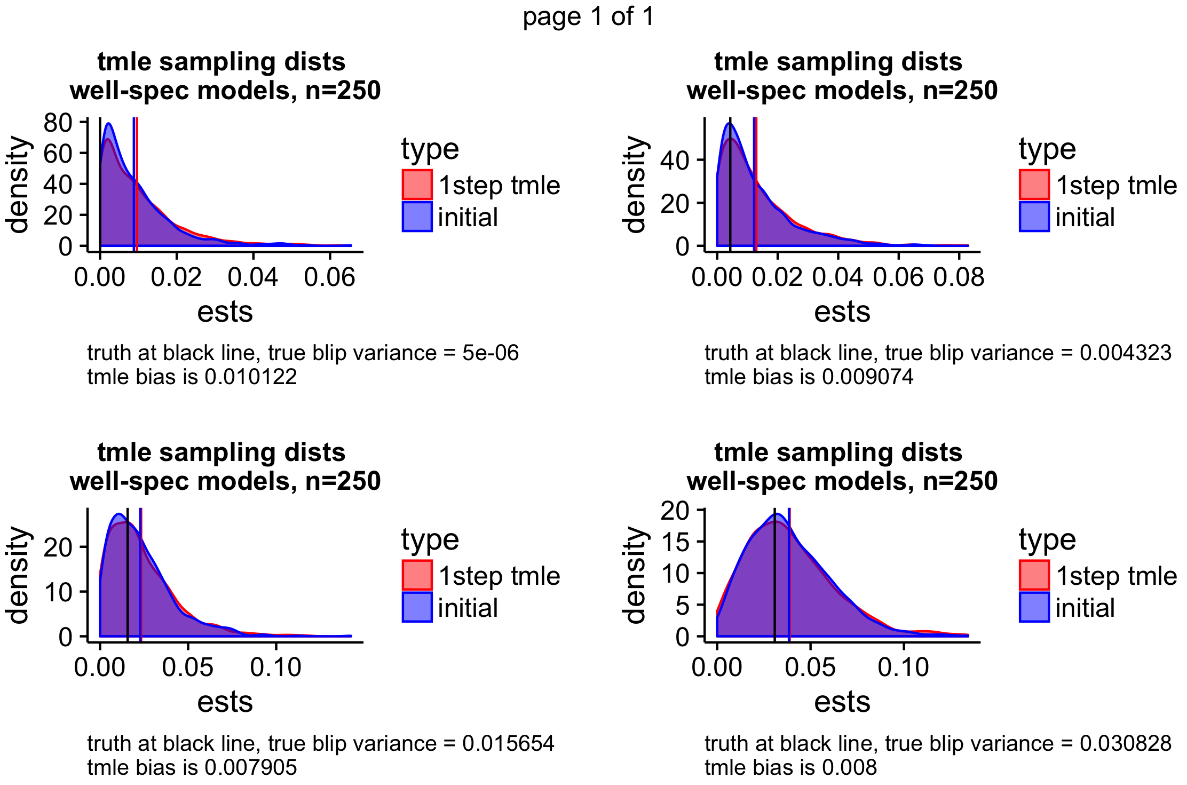
4.3 Well-specified Initial Estimates and TMLE

"Well-specified" means we are fitting a well-specified (functional form is correct for both outcome model, $\mathbb{E}[Y|A,W] = \bar{Q}(A,W)$ and treatment mechanism, $\mathbb{E}[A|W] = g_0(A,W)$), as in a logistic linear model for both the treatment mechanism and outcome models. The only point of these simulations is to show that TMLE preserves excellent initial estimates and also to show approximately what size sample and true blip variance will lead to skewing of the sampling distribution when the truth is near the lower parameter bound of 0. We refer the reader to the online supplemental materials to see these results (refer) but we can say as a rule of thumb, a sample size of 500 or more is probably needed to get reliable estimates for blip variances in the neighborhood of 0.02 (14% standard deviation).

$\bar{Q}(A,W) = \mathit{expit}(0.14(2A + W1 + aAW_1 - bAW_2 + W_2 - W_3 + W_4))$ for the outcome regression, varying a and b to adjust the size of the blip variance.

$\mathbb{E}[A|W] = g0 = \mathit{expit}(-0.4 * W1 + 0.195 * W2 + 0.04 * W3 - 0.06 * W4 - 0.075)$ was the true treatment mechanism.

Figure 11



truth at black line, true blip variance = 5e-06
tmle bias is 0.010122

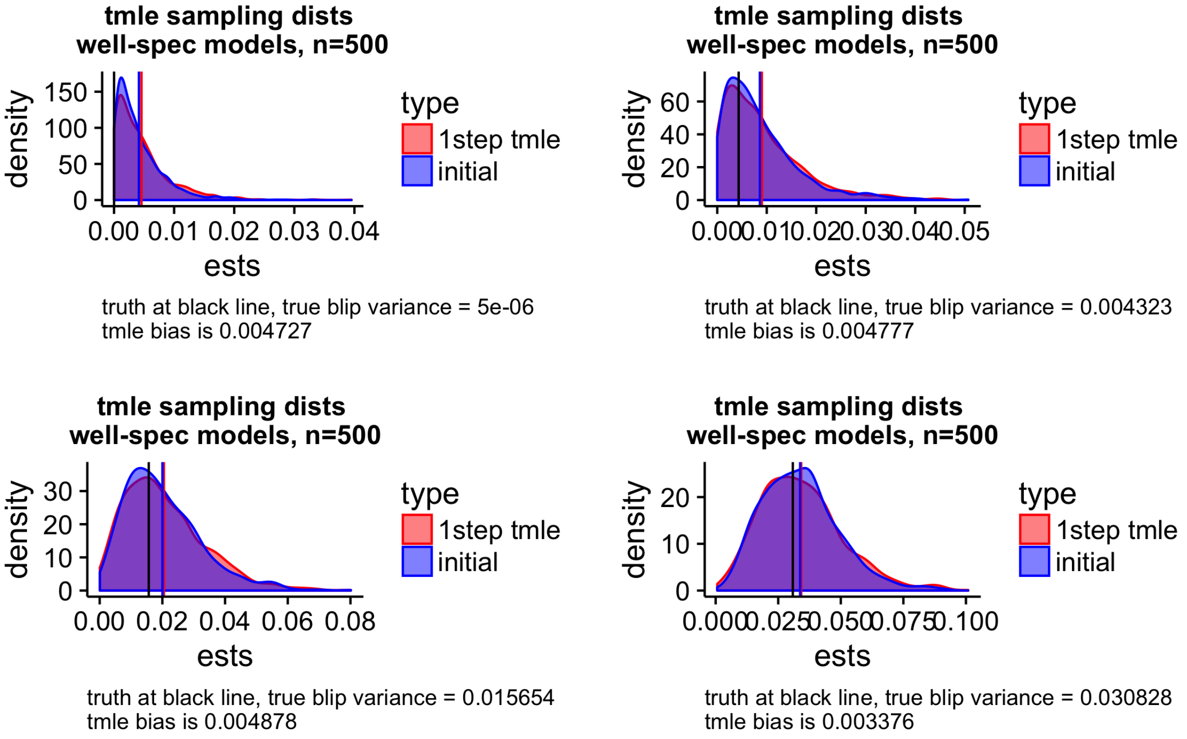
truth at black line, true blip variance = 0.004323
tmle bias is 0.009074

truth at black line, true blip variance = 0.015654
tmle bias is 0.007905

truth at black line, true blip variance = 0.030828
tmle bias is 0.008

Figure 12

page 1 of 1



tmle sampling dists
well-spec models, n=500

truth at black line, true blip variance = 5e-06
tmle bias is 0.004727

tmle sampling dists
well-spec models, n=500

truth at black line, true blip variance = 0.004323
tmle bias is 0.004777

tmle sampling dists
well-spec models, n=500

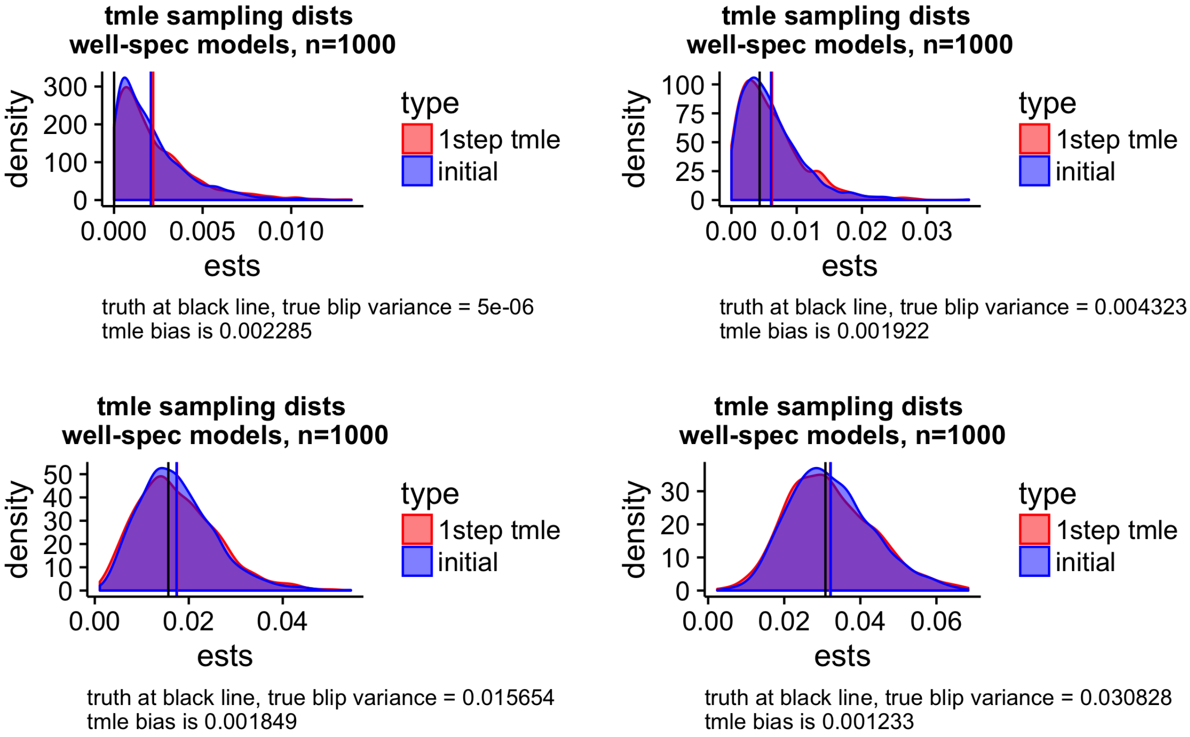
truth at black line, true blip variance = 0.015654
tmle bias is 0.004878

tmle sampling dists
well-spec models, n=500

truth at black line, true blip variance = 0.030828
tmle bias is 0.003376

Figure 13

page 1 of 1



tmle sampling dists
well-spec models, n=1000

truth at black line, true blip variance = 5e-06
tmle bias is 0.002285

tmle sampling dists
well-spec models, n=1000

truth at black line, true blip variance = 0.004323
tmle bias is 0.001922

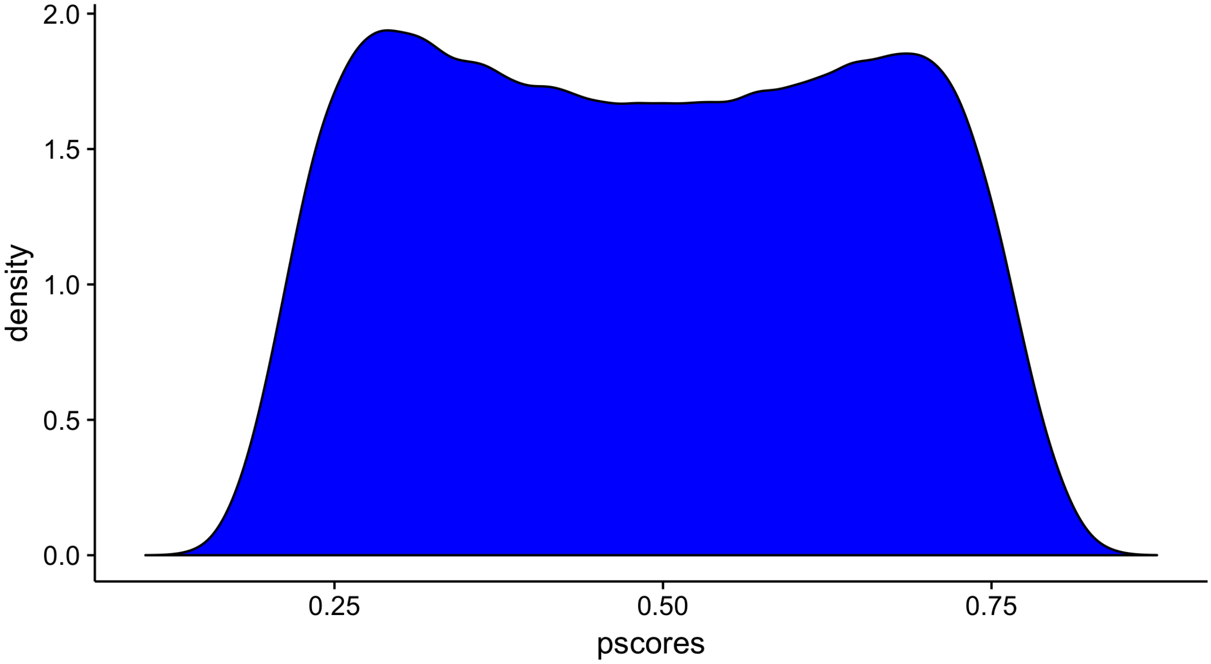
tmle sampling dists
well-spec models, n=1000

truth at black line, true blip variance = 0.015654
tmle bias is 0.001849

tmle sampling dists
well-spec models, n=1000

truth at black line, true blip variance = 0.030828
tmle bias is 0.001233

Figure 14



4.4 Case 2: Complex Transcendental Form for $\bar{Q}$, g_0 is well-specified, Superlearner Employed

The point here is to show TMLE can have good performance even when the outcome model is a complicated transcendental form that is impossible to specify exactly with a polynomial model. With a very limited superlearner library, which we employ throughout without change at sample size of 1000, we start seeing good performance in a normally catastrophic situation for standard methods. TMLE enables the use of the best possible ensemble learning for prediction while allowing for incredibly fast, conservative inference via the efficient influence curve. In the setting of ensemble learning, which is so necessary here, finding inference via expensive bootstrapping is still often theoretically problematic. In addition to using the sample standard deviation of the influence curve approximation to form confidence intervals, we compute the true variance of our estimators based on the 1000 simulations and use this in standard errors to form confidence bounds based on the normal distribution, i.e. $\hat{\theta} \pm 1.96 * \sqrt{\text{true variance}}$. This gives an indication of how well we captured the true variance.

Data Generating Distribution:

case 2a

$\mathbb{E}[Y|A,W] = Q0 = expit(0.28*A+0.28*A*W1+2.8*cos(W1)*A-0.42*W1*sin(2*W2)+0.14*cos(W1)-0.42*W2+0.56*A*(W2^2)+0.42*cos(W4)*A+0.14*A*W1^2-0.28*sin(W2)*W4-0.72*A*W3*W4-3)$

$\mathbb{E}[A|W] = g0 = expit(-0.4*W1+0.195*W2+0.04*W3-0.06*W4-0.075)$

True Causal Risk Difference = 0.263

True Blip Variance = 0.079

Number of simulations: 1000

Sample size of each simulation: n=1000

case 2b

$\mathbb{E}[Y|A,W] = Q0 = expit(0.28*A+2.8*cos(W1)*A+cos(W1)-0.56*A*(W2^2)+0.42*cos(W4)*A+0.14*A*W1^2)$

Same g_0 as case 2a.

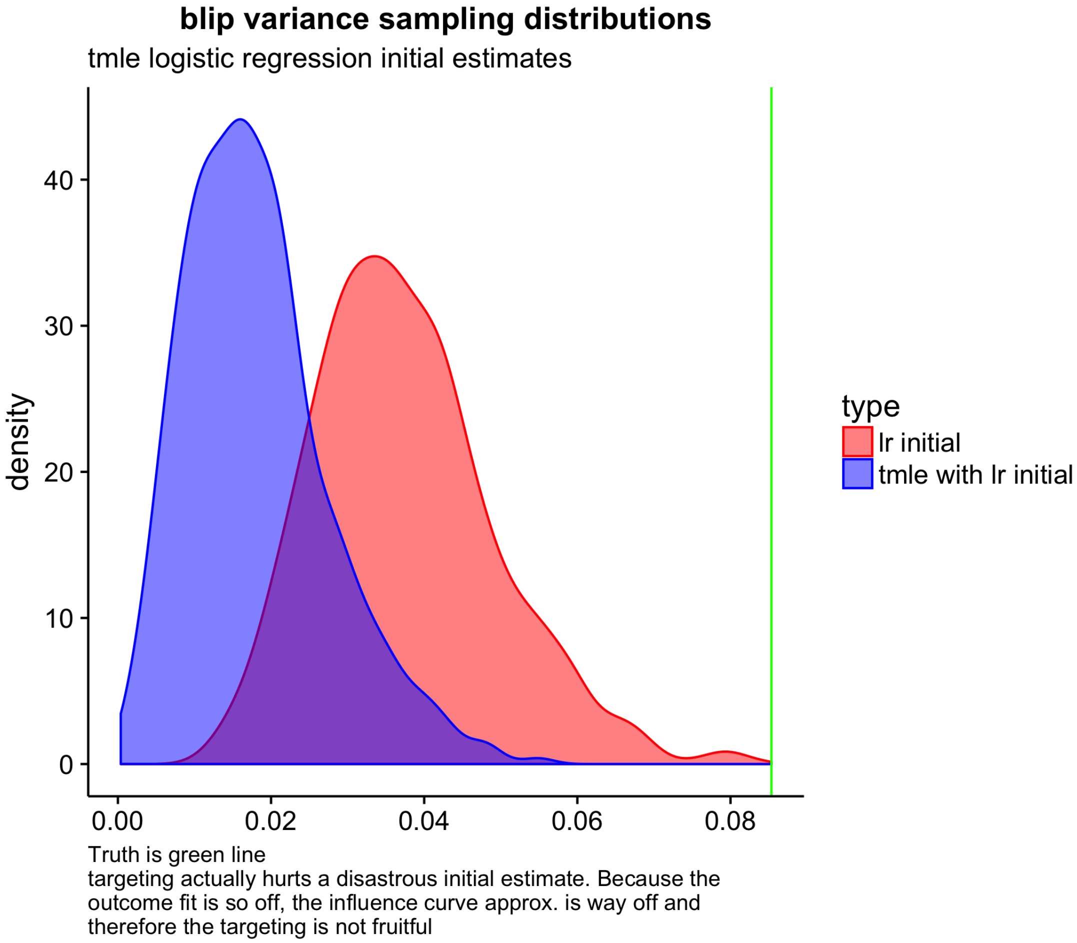
True Causal Risk Difference = 0.078

True Blip Variance = 0.085

4.4.1 Pitfall of a Narrow Model

To illustrate the pitfall of using a too narrow model, we will form initial predictions using a standard glm with all main terms and interactions with treatment. The TMLE for blip variance, assuming a well-specified treatment mechanism, relies on estimating the blip function to eliminate bias to account for the 2nd order remainder term but, also needs to estimate the outcome well to provide the right inference. The targeting that lessens empirical loss while solving the influence curve equation, will not be worthwhile if the influence curve approximation is off.

Figure 15: For case 2a

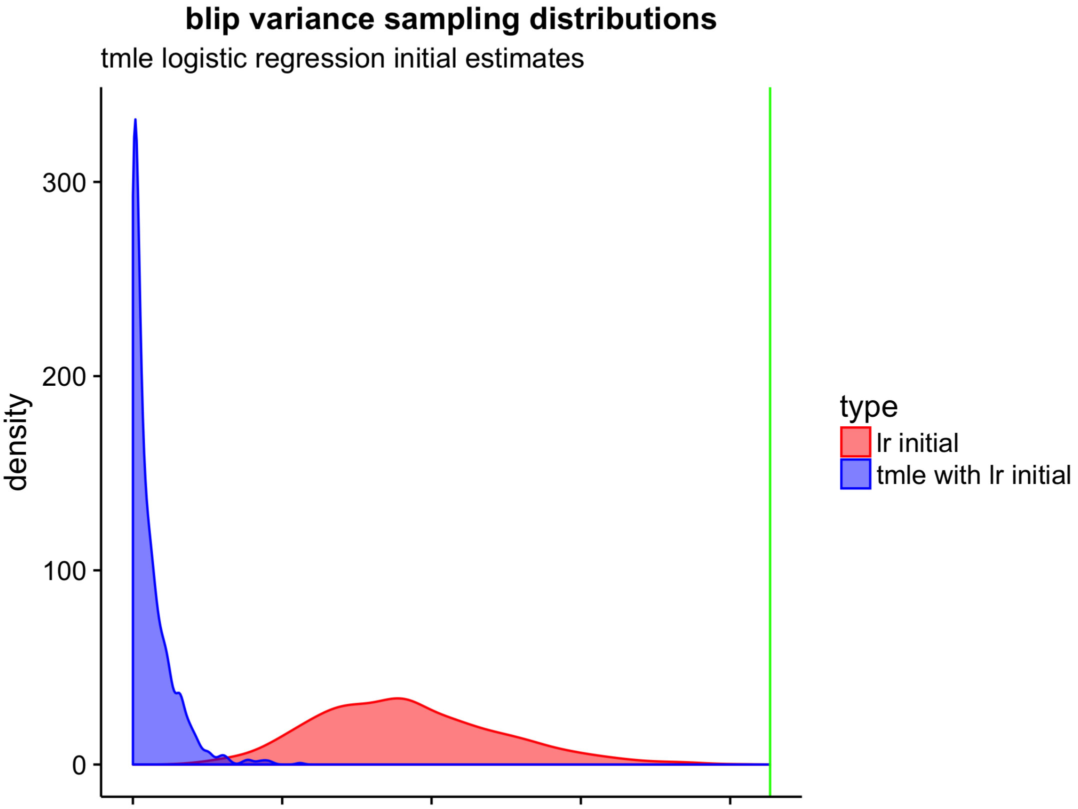


blip variance sampling distributions

tmle logistic regression initial estimates

Truth is green line
targeting actually hurts a disastrous initial estimate. Because the outcome fit is so off, the influence curve approx. is way off and therefore the targeting is not fruitful

Figure 16: for case 2b



Truth is green line
targeting actually hurts a disastrous initial estimate. Because the outcome fit is so off, the influence curve approx. is way off and therefore the targeting is not fruitful

4.4.2 Using Highly Adaptive Lasso (HAL) For Initial Estimates

We can see for this data generating process that using glm with interactions cannot be helped by targeting but using the highly adative lasso (HAL), though greatly biased in the initial estimate of the parameter, allowed for targeting to salvage reasonable coverage. We see TMLE gives up a good amount of slack in the variance to reduce the bias and achieve a much better balance.

We see below that coverage improved with use of the true variance, suggesting that targeting the variance of the influence curve might help coverage here or more simply, just using CV-TMLE.

Table 1: Performance Case 2a

	var	bias	mse
1 step tmle HAL	0.000494	0.014832	0.000714
init est HAL	0.000066	-0.046406	0.002219

Table 2: Coverage Case 2a

1 step TMLE HAL	1 step TMLE HAL TRUE VAR
0.825	0.891

Figure 17: Sampling Dists under case 2a

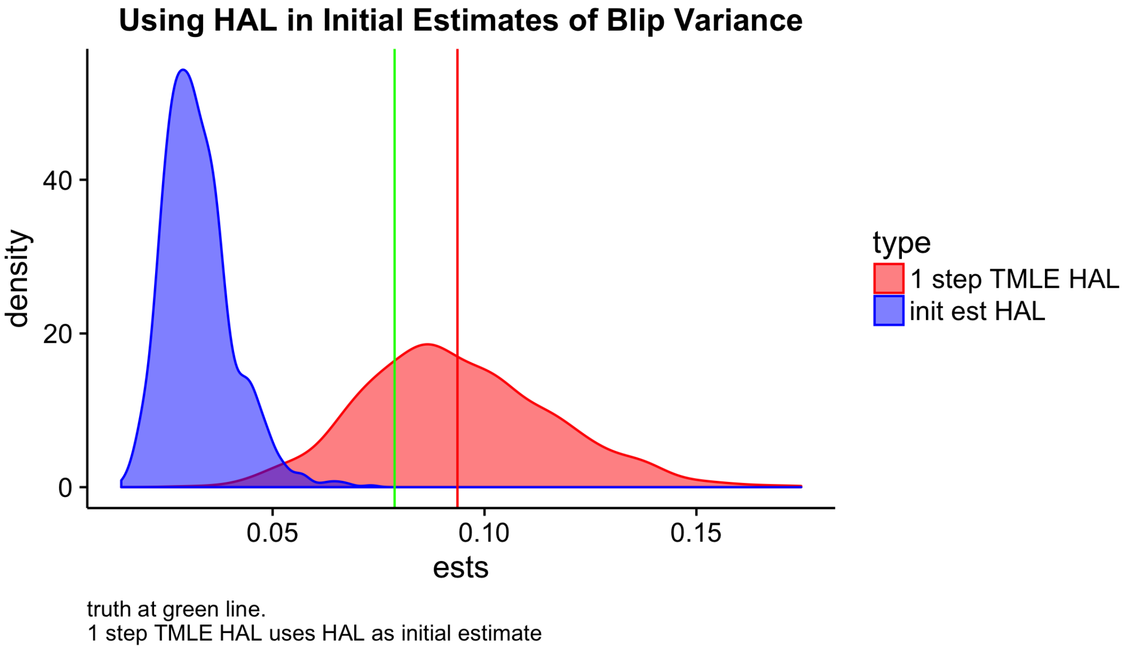


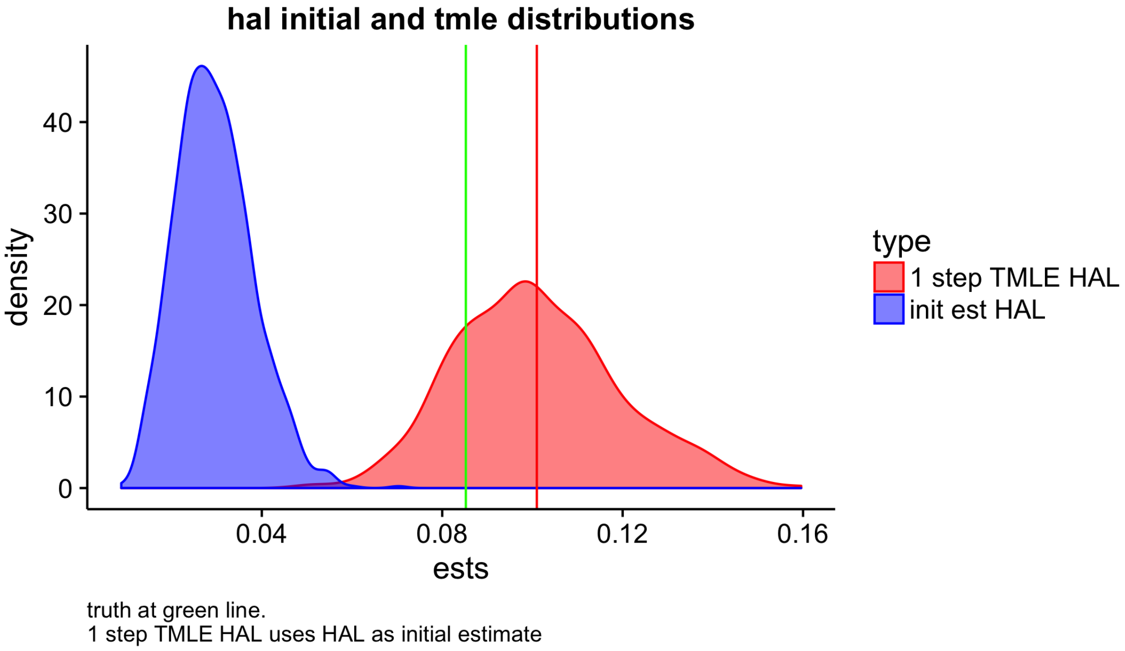
Table 3: Performance case 2b

	var	bias	mse
1 step TMLE HAL	0.000326	0.015737	0.000574
init est HAL	0.000072	-0.055659	0.003170

Table 4: Coverage case 2b

1 step TMLE HAL	1 step TMLE HAL TRUE VAR
0.859	0.857

Figure 18: Sampling Dists under case 2b



truth at green line.
1 step TMLE HAL uses HAL as initial estimate

4.4.3 case 2a, using SuperLearner Library 1

Below we keep the same data generating process but use SuperLearner Library 1, which does not overfit, to obtain the outcome prediction, staying with a correctly specified model for the treatment mechanism. We can see in figure 20 how TMLE gave up some slack in the variance to reduce the bias and also appears quite a bit more normally distributed than the ”flat top” initial SuperLearner estimate.

Table 5: Performance estimating ATE

	var	bias	mse
simultaneous TMLE	0.000972	-0.000630	0.000972
ate TMLE	0.000965	-0.000463	0.000966
init est SL1	0.000889	-0.003329	0.000900

simultaneous TMLE estimated ATE and blip variance

Table 6: Performance estimating Blip Variance

	var	bias	mse
Blip Variance TMLE	0.000301	0.004460	0.000321
Simultaneous TMLE	0.000301	0.004519	0.000322
init ests SL1	0.000210	-0.020031	0.000611

simultaneous TMLE estimated ATE and blip variance

Table 7: Coverage of Estimators

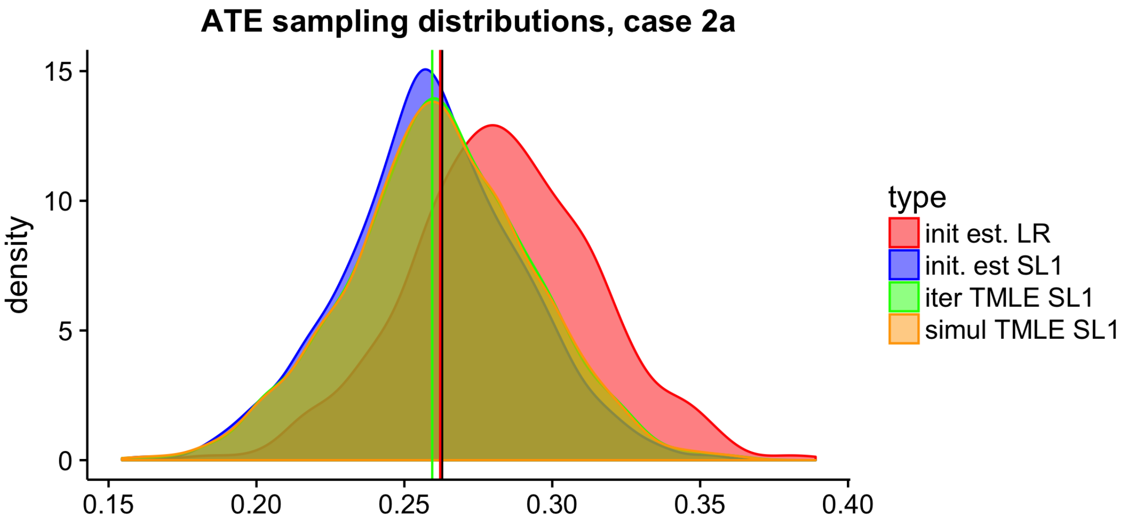
	coverage
TMLE Blip variance	0.946
Simultaneous TMLE	0.933
TMLE ATE	0.927
TMLE Blip var using true var	0.940
ATE using true var	0.949

simultaneous TMLE coverage is for covering true ATE and Blip Variance simultaneously

Table 8: SuperLearner Library 1 Avg Coef, case 2a

	coef
SL.gam3_screen.Main	0.00030
SL.gam3_screen10	0.06662
SL.gam3_screen6	0.02327
SL.glmnet_1_All	0.00507
SL.glmnet_2_All	0.01476
SL.glmnet_3_All	0.01382
rpartPrune	0.00014
nnetMain_screen.Main	0.23125
earthMain_screen.Main	0.35669
SL.glm_screen.Main	0.00007
SL.glm_screen6	0.00253
SL.glm_screen10	0.03033
SL.stepAIC_All	0.14267
SL.hal_screen.Main	0.11138
SL.mean_All	0
glm.mainint_screen.Main	0.00103

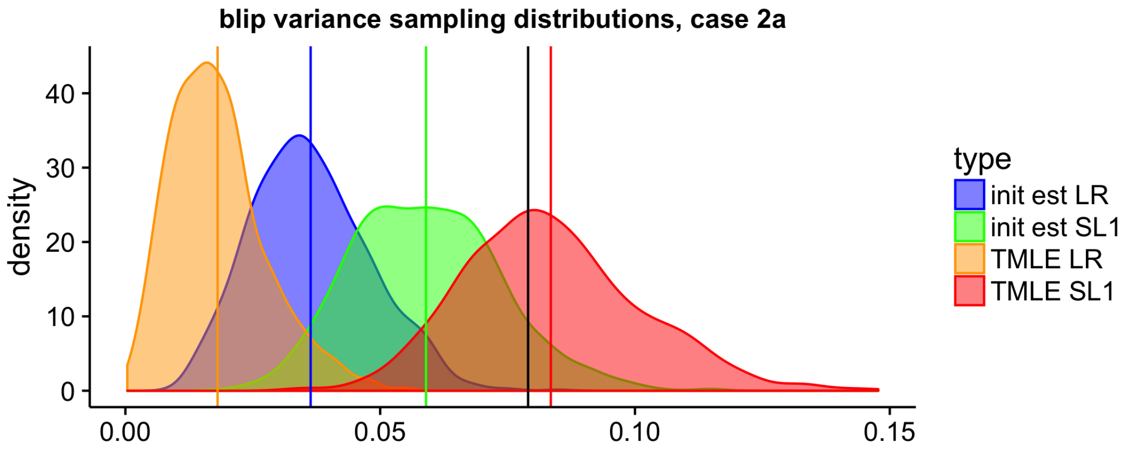
Figure 19



ATE sampling distributions, case 2a

Both TMLE's are virtually the same, initial is excellent
Truth is at black vline. TMLE for ATE using true variance covers at 94.9%
as opposed to 92.7% using IC approximation for variance, suggesting
targeting the variance of IC would be a little fruitful or running cv-TMLE. 1step simul TMLE SL1
estimated blip variance with ATE

Figure 20



truth is black line
tmle SL1, which used Superlearner Library 1 for initial ests
does a great job coverage is 94.6%, debiases initial est SL1
and is more normal-looking. tmle LR used logistic regression
with main terms and interactions init ests is a disaster
even worse than logistic regression plug-in initial est.
TMLE holds no promises for terrible initial estimates.

4.4.4 case 2b: A Case for Using CV-TMLE as a precaution

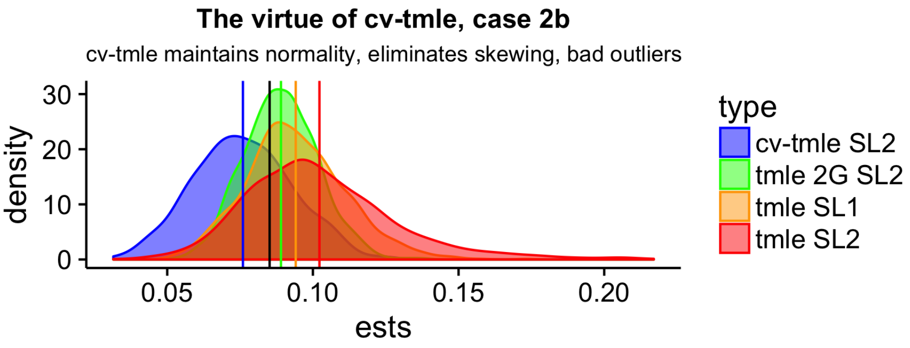
We performed 1000 simulations for each of the four cases below. Sample size was 1000 except for 1 step tmel 2G SL2 which had sample size 2000. We can see that though the slightly overfitting SuperLearner library (Library 2) caused some bias, cv-tmle still covers nearly nominally and has a nice normal-shaped sampling distribution where as the regular one-step tmle using this same library, suffers from some skewing and severe outliers even though the library only had 1 out of the 13 algorithms sometimes overfitting. These overfitting learners got 0 SuperLearner coefficient once sample size was bumped up to 2000 instead of 1000 and hence the excellent performance. The regular 1 step tmle using library 1, which was designed not to overfit, had a normal-looking sampling distribution and covered very near nominally.

Table 9: CV-TMLE saves an overfitting library, SL2

	var	bias	mse	coverage%
TMLE SL1	0.0003	0.009	0.0004	93.8
simul TMLE SL1	0.0003	0.009	0.0004	93.8
init est SL1	0.0002	-0.007	0.0003	NA
TMLE SL2	0.001	0.017	0.001	82
simul TMLE SL2	0.001	0.017	0.001	77.8
init est SL2	0.0003	-0.010	0.0004	NA
CV-TMLE SL2	0.0003	-0.009	0.0004	87.4
simul CV-TMLE SL2	0.0003	-0.009	0.0004	88.6
CV-init est SL2	0.0002	-0.012	0.0003	NA
TMLE 2G SL2	0.0002	0.004	0.0002	93.3
simul TMLE 2G SL2	0.0002	0.004	0.0002	95.4
init ests 2G SL2	0.0001	-0.005	0.0002	NA

2G means samples size was 2000 instead of 1000
SL2 was SuperLearner Library 2, which overfit somewhat.
SL1 did not overfit
simul means estimation of blip variance and ATE was performed simultaneously and coverage for such was for covering both true parameter values

Figure 21



truth at the black line
'SL2' means we used SuperLearner library 2, slightly overfiting
SL1 was a library that did not overfit
2G had samples size 2000 instead of 1000 like the others
For 2G, the overfitters in library 2 got 0 coefficient.
All have nice symmetry except '1 step tmle SL2' which had some bad outliers due to SuperLearner library overfitting. Also MSE and coverage for '1 step tmle SL2' is 82% as opposed to 87.4%, 93.8% and 93.3% for cv-tmle tmle SL1 and tmle 2G, respectively. We can see cv-tmle has big benefits here and a safer bet for an aggressive SuperLearner library.

Table 10: SuperLearner library 1, ave coefficients, case 2b, n=1000

	coef
SL.gam3_screen.Main	0.008
SL.gam3_screen10	0.083
SL.gam3_screen6	0.153
SL.glmnet_1_All	0.006
SL.glmnet_2_All	0.010
SL.glmnet_3_All	0.013
rpartPrune	0.025
nnetMain_screen.Main	0.112
earthMain_screen.Main	0.252
SL.glm_screen.Main	0
SL.glm_screen6	0.025
SL.glm_screen10	0.018
SL.stepAIC_All	0.252
SL.hal_screen.Main	0.041
glm.mainint_screen.Main	0.003
SL.mean_All	0.0002

Table 11: SuperLearner library 2, ave coefficients, case 2b, n=1000

	coef
SL.gam3_screen.Main	0.00691
SL.gam3_screen10	0.11883
SL.gam3_screen6	0.19020
SL.glmnet_1_All	0.00606
SL.glmnet_2_All	0.01267
SL.glmnet_3_All	0.02540
rpartPrune	0.0012
xgbFull_All	0.02310
xgbMain_screen.Main	0.01953
nnetMain_screen.Main	0.04382
earthMain_screen.Main	0.08092
rangerFull_screen.Main	0.03104
SL.glm_screen.Main	0.00014
SL.glm_screen6	0.03132
SL.glm_screen10	0.02691
SL.stepAIC_All	0.35589
SL.hal_screen.Main	0.02415
SL.mean_All	0.00191

NOTE, xgb and ranger, which overfit, got coefficients
This caused some bad outliers, skewing and less efficiency

4.5 Mis-specified Treatment Mechanism and Outcome Model

Number of simulations: 1000, **sample size of each simulation: n=1000**

TMLE with good initial estimates achieve the right bias-variance trade-off, reducing bias and either lowering MSE or keeping it very close to the initial estimates, covering nominally or very nearly so. We also use the same SuperLearner library, Library 1, for different data generating distribution and it does remarkably well. We use Library G for estimating the treatment mechanism.

4.5.1 Case 3, True blip variance: 0.05497 and true ATE = 0.1943

outcome model: $E[Y \mid A, W] = expit(0.14(2A + 5AW_1 + 4AW_3W_4 + W_2W_1 + W_3W_4 + 10Acos(W_4)))$

treatment mechanism: $E[A \mid W_] = expit(0.5(-0.08W_1^2W_2 + 0.5W_1 + 0.49cos(W_2)W_3 + 0.18W_3^2 - 0.12sin(W_4) - 0.15))$

Table 12: Performance for Blip Variance Estimators

	var	bias	mse	coverage%
1 step tmle SL1	0.00019	0.00529	0.00022	95.4
1 step simul tmle SL1	0.00019	0.00535	0.00022	95.2
1 step tmle LR	0.00014	-0.01187	0.00028	78.5
init est SL1	0.00011	-0.01353	0.00029	NA
init est LR	0.00013	-0.01314	0.00030	74.9

coverage for 95% CI's

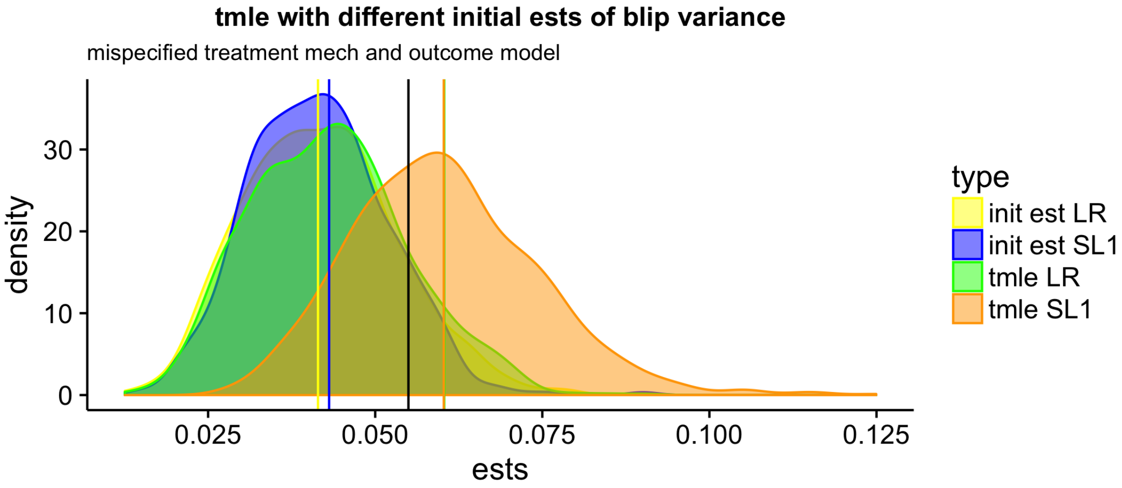
Table 13: SuperLearner library 1, ave coefficients, case 3

	SL.coef
SL.gam3_screen.Main	0.004
SL.gam3_screen10	0.052
SL.gam3_screen6	0.009
SL.glmnet_1_All	0.019
SL.glmnet_2_All	0.024
SL.glmnet_3_All	0.054
rpartPrune	0.001
nnetMain_screen.Main	0.164
earthMain_screen.Main	0.094
SL.glm_screen.Main	0.006
SL.glm_screen6	0.011
SL.glm_screen10	0.014
SL.stepAIC_All	0.426
SL.hal_screen.Main	0.093
SL.mean_All	0
glm.mainint_screen.Main	0.009

Table 14

	SLg.coef
nnetMain_All	0.124
SL.mean_All	0.0005
SL.hal_All	0.098
SL.earth_All	0.074
SL.glm_All	0.445
SL.step.interaction_All	0.193
SL.glm.interaction_All	0.065

Figure 22


tmle with different initial ests of blip variance

mispecified treatment mech and outcome model

truth at black line.
1 step tmle SL1 coverage is 95.4% when using SuperLearner
library 1 for outcome regression and library G for tx mech
1 step tmle LR used glm with main terms and interactions for outcome
and main terms glm for treatment mechanism initial estimates
and coverage was 78.5%.

4.5.2 Case 4, True blip variance = 0.0263 and true ATE = 0.2294

This is an example of fairly small but substantial blip variance one might find in practice. At sample size 1000 we TMLE featuring HAL initial estimates or HAL with glm SuperLearner initial estimates does a remarkable job over TMLE using logistic regression based initial estimates.

outcome model: $E[Y \mid A, W] = expit(0.14(2A + 2AW_1 + 4AW_3W_4 + W_2W_1 + W_3W_4 + 10Acos(W_4)))$

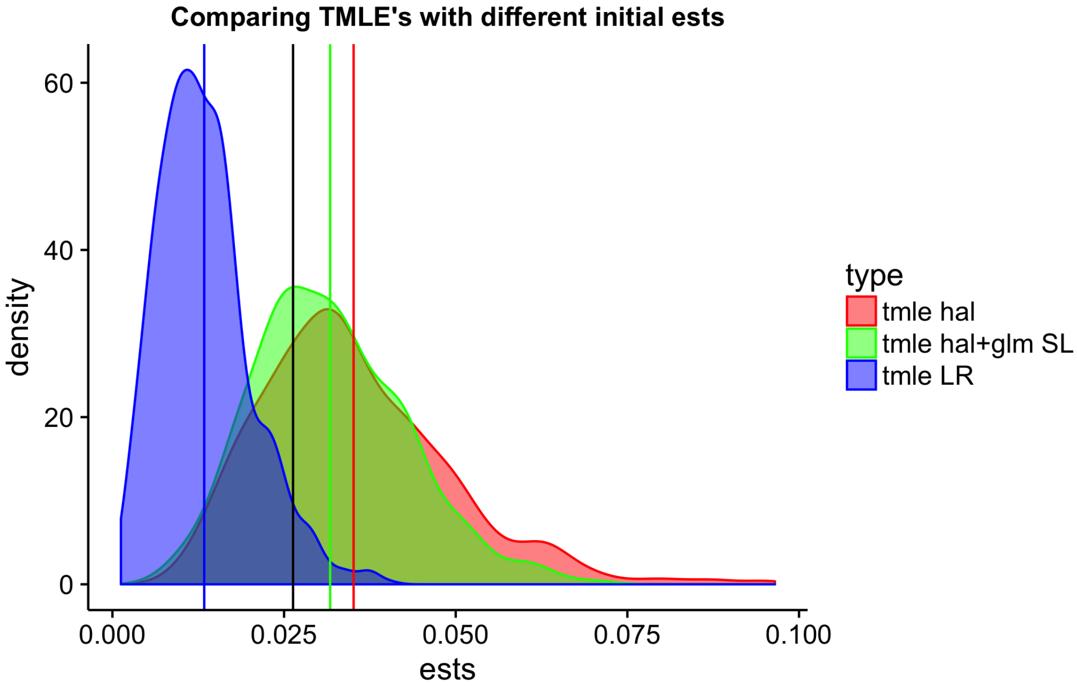
treatment mechanism: $E[A \mid W_{]} = expit(0.5(-0.08W_1^2W_2 + 0.5W_1 + 0.49cos(W_2)W_3 + 0.18W_3^2 - 0.12sin(W_4) - 0.15))$

Table 15

	var	bias	mse	coverage%
glm tmle	0.000045	-0.012936	0.000212	49.5
hal tmle	0.000192	0.008820	0.000270	86
hal+glm SL tmle	0.000124	0.005404	0.000153	93.2
glm init	0.000039	-0.014150	0.000239	43.4
hal init	0.000010	-0.017333	0.000310	NA
hal+glm SL init	0.000009	-0.017440	0.000313	NA

glm tmle still bad bias var trade-off
glm+hal SL tmle clearly wins
Using the true variance, hal tmle covers at 90.6% so variance is slightly underestimated
Using cv-tmle might have gotten the variance for hal tmle closer to the true estimator variance.

Figure 23



truth at black line.
tmle LR uses glm with interactions for outcome model and glm for treatment mechanism initial estimates. tmle LR CI's cover at 50%
hal tmle uses highly adaptive lasso for initial estimates of both outcome and treatment mechanism initials estimates.
tmle hal CI's cover at 86%. tmle hal+glm SL uses a SuperLearner with hal and glm for outcome and treatment mechanism initial estimates
hal+glm tmle CI's cover at 93.2%. All coverage here is using the infuence curve approximation for inference.

5 Demonsration on Real Data

Here we will demonstrate computing a simultaneous one-step TMLE for blip variance, blip standard deviation and average treatment effect on a well-known public data set, Western Collaborative Group Study (WCGS), which aimed to relate behavioral categories to myocardial infarction. The study began in 1960 with a baseline examination of 3154 men, ages 39-59, who were employed in one of ten California companies. Follow-up occurred until 1969 by which time 257 of the men had a diagnosis of coronary heart disease (the outcome of interest). The study was one of the first to investigate personality type and its association on heart disease. The confounders or W = (Age, Height, Weight, Systolic Blood Pressure, Diastolic Blood Pressure, Fasting serum cholesterol, Number of Cigarettes per day) and the treatment, A = Binary behavior type and the binary outcome, Y = indicator of coronary heart disease. As far as determining causal effects it is obvious behavioral type (the treatment in this case) might precede many of the confounders in the time ordering. For the purpose of this demonstration, however, we will overlook this problem and assume the appropriate time-ordering and randomization assumption necessary for identifying the average treatment effect and the blip variance from the data.

5.0.1 Results and Interpretation

The results below indicate clinical effect homogeneity. Using the point estimates for average treatment effect and blip standard deviation, we can say that on average, behavior type 1 has a 5.1% higher chance of myocardial infarction give or take 3.7%. There is very little evidence of effect modification here if we are to believe our predictions achieve the sufficient rate of convergence as required in section 3.1.1. Our confidence interval for blip variance needed to be log scaled to avoid being in the impossible negative range. Our well-specified model simulations (refer to section in supp materials) show that in the case of small blip variance we might bias to the right of 0 as much as the point estimate here.

Table 16: CI's for WCGS data

	est	lower	upper
ate	0.05055	0.02934	0.07175
blip st. dev	0.03741	0.01065	0.06418
blip var log-scaled	0.00140	0.00033	0.00585
blip var	0.00140	-0.00060	0.00340

These are simultaneous confidence intervals which cover
all true parameter values at 95% simultaneously
Here we have a blip standard dev that is smaller than
the average blip so we have relative clinical effect homogeneity

Below are the average coefficients SuperLearner gave to each algorithm over the 10 training folds.

Table 17

	QLearner	Coef	GLearner	Coef.1
1	nnetMain_screen.Main	0.0038	nnetMainG_All	0
2	nnetMain1_screen.Main	0.0308	nnetMainG1_All	0.0192
3	earthFull_screen6	0.0040	SL.earth_All	0.0059
4	earthFull_screen12	0	SL.rpartPrune_All	0.0405
5	earthFull_All	0.0595	SL.gam_All	0.5674
6	SL.earth_screen.Main	0.0037	rpartMain_All	0.0479
7	xgboostFull_screen12	0	SL.step.interaction_All	0
8	xgboostFull_All	0.0103	SL.glm_All	0.217
9	xgboostMain_screen.Main	0.1420	SL.hal_All	0.0256
10	gamFull_screen6	0	SL.mean_All	0
11	gamFull_screen12	0	xgboostG_All	0.0765
12	gamFull_All	0.0127		
13	SL.gam_screen.Main	0.0537		
14	SL.rpartPrune_All	0.0023		
15	rangerMain_screen.Main	0.1155		
16	rpartMain_screen.Main	0.0023		
17	SL.stepAIC_All	0.0042		
18	SL.glm_screen6	0		
19	SL.glm_screen12	0		
20	SL.glm_screen.Main	0.5402		
21	SL.glm_All	0.0128		
22	SL.hal_screen.Main	0		
23	SL.mean_All	0.0023		

6 Concluding Remarks

We should note that there are considerable places to explore as a researcher in after or in conjunction with estimating the blip variance and establishing the existence of a high degree of effect modification. Estimating the blip function itself has been explored quite thoroughly in order to get at strata-specific causal effects. One may reference Polley and van der Laan (Polley and Laan 2009), who estimated the blip non-parametrically and applied a doubly robust technique for handling right censoring. Continuing this work in the context of estimating the optimal dynamic rule without knowledge about the propensity score, Alex Luedtke and Mark van der Laan (Luedtke and Laan 2016), sequentially fit the blip in the longitudinal setting. Now, this lineage of work has born fruits with an R package, opttx (Coyle 2017), enabling us to compute an optimal rule for assigning treatment, based in part on blip function estimation.

One might also strive to detect subgroups that stand out in response to treatment. Lu et al (Tian et al. 2014) offered a simple clever way to isolate interactions of treatment with confounders in a randomized trial by transforming the predictors of a parametric model. The main idea is to form a variable, $z = 2A - 1$, where A is the usual treatment indicator, and then put the interaction of this variable with the predictors in the outcome regression. This enables direct estimation of the blip function. One could also employ recursive partitioning in a number of ways to divide the data into homogeneous subgroups as far as treatment effects (Athey and Imbens 2016) as well as random forests (Athey and Imbens 2015).

We can also note that estimating blip variance demands more data than estimating the average causal effect, lacks double robustness and can more easily suffer from complications of bias and skewed sampling distributions caused by the true blip variance being quite close to 0. However, blip variance is fundamental in determining what an individual can expect from a static intervention and a measure of how much can be gained with a more precise approach to assigning treatment.

A Appendix

Set Up

$O \sim P \in \mathcal{M},$ non-parametric and our observed data, O, is of the form, O = (W,A,Y), where W is a set of confounders, A is a binary treatment indicator and Y is the outcome, continuous or binary. It is important that we can factorize the density for P is as $p(o) = p_Y(y|a,w)g(a|w)p_W(w)$.

A.0.1 Tangent Space for Nonparametric Model

We consider the one dimensional set of submodels that pass through p at $epsilon = 0$. (van der Vaart, 2007) $\{p_\epsilon = (1+\epsilon s)p| \int sdP = 0, \int s^2 dP < \infty\}$
The tangent space is the closure in L^2 norm of the set of scores, s, or directions for the paths defined above. We write:

$$\begin{aligned} T &= \overline{\{s(o)|\mathbb{E}s = 0, \mathbb{E}s^2 < \infty\}} \\ &= \overline{\{s(y|a,w)|\mathbb{E}_{P_Y}s = 0, \mathbb{E}s^2 < \infty\}} \oplus \overline{\{s(a|w)|\mathbb{E}_{P_A}s = 0, \mathbb{E}s^2 < \infty\}} \oplus \overline{\{s(w)|\mathbb{E}_{P_W}s = 0, \mathbb{E}s^2 < \infty\}} \\ &= T_Y \oplus T_A \oplus T_W \end{aligned}$$

$L_0^2(P)$ forms a Hilbert space with inner product defined as $\langle f, g\rangle = \mathbb{E}_P fg$. The set of all cauchy sequence limits under the induced norm forms the Hilbert space. Our notion of orthogonality now is $f \perp g$ if and only if $\langle f, g\rangle = 0$ and, therefore, the above direct sum becomes clearly valid. In other words, every score, s, can be written as $\frac{d}{d\epsilon} log(p_\epsilon)|_{\epsilon=0} = s(w,a,y) = s_Y(y|a,w) + s_A(a|w) + s_W(w)$ where, due to the fact $p_\epsilon = (1+\epsilon s)p = p_{Y\epsilon} p_{A\epsilon} p_{W\epsilon}$, it is easy to see $\frac{d}{d\epsilon} log(p_{Y\epsilon})|_{\epsilon=0} = s_Y(y|a,w)$, $\frac{d}{d\epsilon} log(p_{A\epsilon})|_{\epsilon=0} = s_A(a|w)$ and $\frac{d}{d\epsilon} log(p_{W\epsilon})|_{\epsilon=0} = s_W(w)$.

A.0.2 Efficiency Theory in brief

Our parameter of interest is a mapping from the model, $\mathcal{M}$ to the real numbers given by $\Psi(P) = \mathbb{E}(B(W) - \mathbb{E}B)^2$ where $B(W) = \mathbb{E}[Y|A=1,W] - \mathbb{E}[Y|A=0,W]$. We refer to van der Vaart (2007) to define an influence function, otherwise known as a gradient, as a continuous linear map from T to the reals given by

$$\lim_{\epsilon \to 0} \left(\frac{\Psi(P_\epsilon) - \Psi(P)}{\epsilon} \right) \longrightarrow \dot{\Psi}_P(s) \tag{3}$$

We note to the reader, we imply a direction, s, when we write P_e, which has density $p(1+\epsilon s)$, but generally leave it off the notation as understood.

By the riesz representation theorem for Hilbert Spaces, the mapping in (2) can be written in the form of an inner product $\langle D(P), g\rangle$ where $D(P) \in L_0^2(P)$ is not necessarily unique if T is not all of $L_0^2(P)$. On the subspace, T, there exists a unique gradient known as the canonical gradient or efficient influence curve. Thus, in the case of a nonparametric model, the only gradient is the canonical gradient. The efficient influence curve has a variance that is the lower bound for any regular asymptotically linear estimator (van der Vaart, 2007). Since the TMLE, under conditions as discussed in this paper, asymptotically achieves variance equal to that of the efficient influence curve, the estimator is asymptotically efficient.

As a note to the reader: Our parameter mapping does not depend on the treatment mechanism, g, and also $T_A \perp T_Y \oplus T_W$ which, means our efficient influence curve must therefore be in $T_Y \oplus T_W$ for the nonparametric model. Therefore, our efficient influence curve for both blip variance, as given in section 5.3, will have two orthogonal components, $D^\star_{2,1}$ and $D^\star_{2,2}$, in T_Y and T_W respectively. We have no component in T_A, which is why we need not perform a TMLE update of the initial prediction, g_n, of $g_0(A|W)$. The same is true for the causal risk difference.

A.1 Derivation of the efficient influence curve

For the below proof, when we take the derivatives we will indicate that such is in the direction of a given score, $s \in T$. We will assume a dominating measure ν and merely use ν to always denote the dominating measure of all densities involved. We could perhaps just assume continuous densities and use lebesque measure as well and nothing would change below.

Theorem A.1. *Let $\Psi(P) = var_P(B(W))$. The efficient influence curve for Ψ at P is given by:*

$$\mathbf{D^\star(P)(W,A,Y)} = \mathbf{2\left(B(W) - \mathbb{E}B(W)\right)\left(\frac{2A-1}{g(A|W)}\right)\left(Y - \bar{Q}(A,W)\right) + \left(B(W) - \mathbb{E}B\right)^2} - \mathit{\Psi}(\mathbf{P})$$

where $\bar{Q}(A,W) = \mathbb{E}(Y|A,W)$

Proof. By our previous discussion, we are guaranteed a unique representer in $L_0^2(P)$, called $D^\star$ such that

$$\frac{d}{d\epsilon}\Psi(P_\epsilon)(s)\bigg|_{\epsilon=0} = \langle D^\star, s\rangle = \mathbb{E}D^\star s$$

We will now write $\frac{d}{d\epsilon}\Psi(P_\epsilon)(s)\Big|_{\epsilon=0}$ as $\langle D^\star, s\rangle$ and $D^\star$ will be our desired formula:

$$\begin{aligned}
\frac{d}{d\epsilon}\Psi(P_\epsilon)(s)\bigg|_{\epsilon=0} &= \frac{d}{d\epsilon}\mathbb{E}_{P_\epsilon}\left(B_{P_\epsilon}(W) - \mathbb{E}_{P_\epsilon}B_{P_\epsilon}(W)\right)^2\bigg|_{\epsilon=0} \\
&= \frac{d}{d\epsilon}\int\left(B_{P_\epsilon}(w) - \mathbb{E}_{P_\epsilon}B_{P_\epsilon}(W)\right)^2 p_\epsilon(o)\nu(do)\bigg|_{\epsilon=0} \\
&= \int 2\left(B_{P_\epsilon}(w) - \mathbb{E}_{P_\epsilon}B_{P_\epsilon}(W)\right)\frac{d}{d\epsilon}\left(B_{P_\epsilon}(w) - \mathbb{E}_{P_\epsilon}B_{P_\epsilon}\right)p(o)\nu(do)\bigg|_{\epsilon=0} + \mathbb{E}\left[\left(B(W) - \mathbb{E}B(W)\right)^2 s(O)\right] \\
&= \int 2\left(B_P(w) - \mathbb{E}_P B_P(W)\right)\frac{d}{d\epsilon}B_{P_\epsilon}(w)p(o)\nu(do)\bigg|_{\epsilon=0} + \mathbb{E}\left[\left(\left(B(W) - \mathbb{E}B(W)\right)^2 - \Psi(P)\right)s(O)\right] \quad (4)
\end{aligned}$$

Now we can compute the first term in (4)

$$\begin{aligned}
& 2\int\left(B_{P_\epsilon}(w) - \mathbb{E}_{P_\epsilon}B_{P_\epsilon}(W)\right)\frac{d}{d\epsilon}\left[\int\left(yp_{Y\epsilon}(y|a=1,w) - yp_{Y\epsilon}(y|A=0,w)\right)\nu(dy)\right]p_W(w)\nu(dw)\bigg|_{\epsilon=0} \\
= \; & 2\int\left(B_{P_\epsilon}(w) - \mathbb{E}_{P_\epsilon}B_{P_\epsilon}(W)\right)\frac{d}{d\epsilon}\left[\int\frac{2a-1}{g(a|w)}p_{Y\epsilon}(y|a,w)g(a|w)\nu(d(y\times a))\right]p_W(w)\nu(dw)\bigg|_{\epsilon=0} \\
& 2\int\left(B_{P_\epsilon}(w) - \mathbb{E}_{P_\epsilon}B_{P_\epsilon}(W)\right)\left[\int\frac{2a-1}{g(a|w)}\frac{d}{d\epsilon}\frac{p_\epsilon(w,a,y)}{p_{A\epsilon}(a|w)p_{W\epsilon}(w)}\bigg|_{\epsilon=0}g(a|w)\nu(d(y\times a))\right]p_W(w)\nu(dw) \quad (5) \\
= \; & 2\int\left(B_{P_\epsilon}(w) - \mathbb{E}_{P_\epsilon}B_{P_\epsilon}(W)\right)\frac{y(2a-1)}{g(a|w)}\left(s(w,a,y) - \left(s_W(w) + s_A(a|w)\right)p(w,a,y)\nu(do)\right) \\
= \; & 2\int\left(B_{P_\epsilon}(w) - \mathbb{E}_{P_\epsilon}B_{P_\epsilon}(W)\right)\int\frac{(2a-1)}{g(a|w)}(y - \bar{Q}(a,w))s(w,a,y)p(w,a,y)\nu(d(o)) \\
= \; & 2\int\left(B_{P_\epsilon}(w) - \mathbb{E}_{P_\epsilon}B_{P_\epsilon}(W)\right)\int\frac{(2a-1)}{g(a|w)}(y - \bar{Q}(a,w))s(o)p(o)\nu(d(o)) \quad (6)
\end{aligned}$$

$$2\left(B(W) - \mathbb{E}B(W)\right)\left(\frac{2A-1}{g(A|W)}\right)(Y - \bar{Q}(A,W)) + \left(B(W) - \mathbb{E}B\right)^2 - \Psi(P)$$

is the aforementioned representer, completing the proof. □

A.2 Remainder terms and asymptotic linearity:

The reader may recall the three conditions assuring asymptotic efficiency of the TMLE estimator. Here we will focus on the 2^{nd} condition regarding the remainder term.

Theorem A.2. *If P_0 is the true distribution, it is necessary to estimate the true blip function B_0 at a rate of $\frac{1}{n^{0.25}}$ in the $L^2(P)$ norm in order for TMLE to be a consistent asymptotically efficient estimator under a known treatment mechanism, g_0. If g_0 is unknown, we also need the product of the $L^2(P_0)$ rates for estimating g_0 and $\bar{Q}_0$ to be $\frac{1}{n^{0.5}}$.*

Proof. For this discussion we will drop the subscript, n, and superscript, $\star$ in $P_n^\star$ and merely consider, P, as an estimate of the truth, P_0. We will use $B(W)$ to denote the blip function where the conditional expectation is with respect to distribution, P, ie the estimated blip, and $B_0(W)$ to be the true blip function. Likewise, $\mathbb{E}_0$ is the expectation with respect to the true observed data distribution, P_0, and leaving the subscript, 0, off the expectation sign means the expectation is with respect to P.

$$\begin{aligned}
R_2(P,P_0) &= \Psi(P)-\Psi(P_0)+P_0\left(D^\star(P)\right)\\
&= \mathbb{E}\left(B(W)-\mathbb{E}B(W)\right)^2-\mathbb{E}_0\left(B_0(W)-\mathbb{E}_0B_0(W)\right)^2+\\
&\quad \mathbb{E}_0\left[2\left(B(W)-\mathbb{E}B(W)\right)\frac{2A-1}{g(A|W)}\big(Y-\bar{Q}(A,W\big)+\left(B(W)-\mathbb{E}B(W)\right)^2-\Psi(P)\right]\\
&= -\mathbb{E}_0\left(B_0(W)-\mathbb{E}_0B_0(W)\right)^2+\\
&\quad \mathbb{E}_0\left[2\left(B(W)-\mathbb{E}B(W)\right)\frac{2A-1}{g(A|W)}\big(Y-\bar{Q}(A,W)\big)+\left(B(W)-\mathbb{E}B(W)\right)^2\right]\\
&= \mathbb{E}_0\left[\left(B(W)-\mathbb{E}B(W)\right)^2-\left(B_0(W)-\mathbb{E}_0B_0(W)\right)^2\right]+\\
&\quad \mathbb{E}_0\mathbb{E}_0\left[2\left(B(W)-\mathbb{E}B(W)\right)\frac{2A-1}{g(A|W)}\left(\bar{Q}_0(A,W)-\bar{Q}(A,W)\right)|W\right]\\
&= \mathbb{E}_0\left[\left(B(W)-\mathbb{E}B(W)\right)^2-\left(B_0(W)-\mathbb{E}_0B_0(W)\right)^2\right]+\\
&\quad +\mathbb{E}_{P_W}\left[2\left(B(W)-\mathbb{E}B(W)\right)\left(\frac{g_0(1|W)}{g(1|W)}\left(\bar{Q}_0(1,W)-\bar{Q}(1,W)\right)-\frac{g_0(0|W)}{g(0|W)}\left(\bar{Q}_0(0,W)-\bar{Q}(0,W)\right)\right)\right]\\
&= \mathbb{E}_0\left[\left(B(W)-\mathbb{E}B(W)\right)^2-\left(B_0(W)-\mathbb{E}_0B_0(W)\right)^2+2\left(B_0(W)-B(W)\right)\left(B(W)-\mathbb{E}B(W)\right)\right]\\
&\quad +\mathbb{E}_0\left[2\left(B(W)-\mathbb{E}B(W)\right)\left(\frac{g_0(1|W)-g(1|W)}{g(1|W)}\left(\bar{Q}_0(1,W)-\bar{Q}(1,W)\right)-\frac{g_0(0|W)-g(0|W)}{g(0|W)}\left(\bar{Q}_0(0,W)-\bar{Q}(0,W)\right)\right)\right]\\
&= \left(\mathbb{E}_0B_0(W)-\mathbb{E}B(W)\right)^2-\mathbb{E}_0\left(B_0(W)-B(W)\right)^2\\
&\quad +\mathbb{E}_0\left[2\left(B(W)-\mathbb{E}B(W)\right)\left(\frac{g_0(1|W)-g(1|W)}{g(1|W)}\left(\bar{Q}_0(1,W)-\bar{Q}(1,W)\right)-\frac{g_0(0|W)-g(0|W)}{g(0|W)}\left(\bar{Q}_0(0,W)-\bar{Q}(0,W)\right)\right)\right]
\end{aligned}$$

We can regard the $\left(\mathbb{E}_0B_0(W)-\mathbb{E}B(W)\right)^2$ term and notice that for an unknown, g_0, it is well-known that the double robustness of TMLE in estimating the causal risk difference, $\mathbb{E}_0B_0(W)$, implies that if we estimate both g_0 and $\bar{Q}_0$ so that the product of the respective L_2 rates of convergence is $o(n^{-0.5})$, then we obtain $\sqrt{n}\left(\mathbb{E}_0B_0(W)-\mathbb{E}B(W)\right) \overset{D}{\Longrightarrow} N\left[0, var_0(D_1^\star(P_0))\right]$ where $D_1^\star(P_0)$ is the efficient influence curve for the causal risk difference. We therefore know $\mathbb{E}_0 B_0(W) - \mathbb{E}B(W) \xrightarrow{p} 0$ and by slutsky's theorem, $\sqrt{n}\left(\mathbb{E}_0 B_0(W) - \mathbb{E}B(W)\right)^2 \overset{D}{\Longrightarrow} 0$. Therefore this term poses no additional problem to the rest of the terms.

Now we can address the standard "double robust" term:

$$\begin{aligned}
&\mathbb{E}_0\left[2\left(B(W)-\mathbb{E}B(W)\right)\left(\frac{g_0(1|W)-g(1|W)}{g(1|W)}\left(\bar{Q}_0(1,W)-\bar{Q}(1,W)\right)-\frac{g_0(0|W)-g(0|W)}{g(0|W)}\left(\bar{Q}_0(0,W)-\bar{Q}(0,W)\right)\right)\right]\\
\leq &K\mathbb{E}_0\left[\left|\frac{g_0(1|W)-g(1|W)}{g(1|W)}\left(\bar{Q}_0(1,W)-\bar{Q}(1,W)\right)\right|+\left|\frac{g_0(0|W)-g(0|W)}{g(0|W)}\left(\bar{Q}_0(0,W)-\bar{Q}(0,W)\right)\right|\right]\\
\leq &K\mathbb{E}_0\left|\frac{g_0(A|W)-g(A|W)}{g(A|W)g_0(A|W)}\left(\bar{Q}_0(A,W)-\bar{Q}(A,W)\right)\right|\\
\leq &K\|g_0(A|W)-g(A|W)\|_{L^2(P_0)}\|\bar{Q}_0(A,W)-\bar{Q}(A,W)\|_{L^2(P_0)}
\end{aligned}$$

where the last inequality follows from cauchy-schwarz and the strict positivity assumption on g_0. Therefore we have proven the theorem. □

B Logistic Regression Plug-in Estimator Inference

Here we will just employ basic statistics to obtain the influence curve for both the causal risk difference and variance of the blip functions using the influence curve for the coefficients in a logistic linear model.
Now if we take n iid draws of O we get that the likelihood of drawing $\{O_i\}_{i=1}^n$ is

$$\prod_{i=1}^n expit\left(m(A_i,W_i|\beta\right)^{Y_i}\left(1-expit\left(m(A_i,W_i|\beta)\right)\right)^{1-Y_i} p_A(A_i|W_i)p_W(W_i)$$

The product follows from independence of compound events, of course, ie, Pr(A and B) = Pr(A)Pr(B) if A is indep of B.
We can also note that in maximizing the likelihood, the other parts of the likelihood will disappear. Of course, if the derivative wrt β of the likelihood is 0 then the derivative wrt β of the log-likelihood is also 0. Taking the log then taking the derivative we can write it as follows:

$$\begin{aligned}
\nabla_\beta log\prod_{i=1}^n expit\left(m(A_i,W_i|\beta\right)^{Y_i}\left(1-expit\left(m(A_i,W_i|\beta)\right)\right)^{1-Y_i} p_A(A_i|W_i)p_W(W_i) &= \\
\nabla_\beta\sum_{i=1}^n\left[Y_i log\left(expit\left(m(A_i,W_i|\beta)\right)+(1-Y_i)log\left(1-expit\left(m(A_i,W_i|\beta)\right)\right)\right]\right. &= \quad (7)\\
\sum_{i=1}^n\left[(-1+Y_i)\nabla_\beta m(A_i,W_i|\beta)+log\left(expit\left(m(A_i,W_i|\beta)\right)\right)\right] &= \quad (8)
\end{aligned}$$

Here, $\bigtriangledown_\beta$ is just shorthand for a multivariable calculus $\left(\frac{\partial}{\partial\beta_0}, \frac{\partial}{\partial\beta_1}, ..., \frac{\partial}{\partial\beta_d}\right)^T$ which just makes a vector with a partial derivative for each component of β as a gradient in multivariable calculus class. We thus get d simultaneous equations to solve here much as we do for regular linear regression, except here we cannot solve explicitly but must use Newton's iterative method.
We arrive at the following simplification from (1):

$$\sum_{i=1}^{n} \left(\frac{\partial m}{\partial\beta_{0,n}}, \frac{\partial m}{\partial\beta_{1,n}}, ..., \frac{\partial m}{\partial\beta_{d,n}}\right)^T (Y_i - expit\,(m(A_i, W_i|\beta_n)) = 0$$

We can go one step further from here and derive the multidimensional influence function via the use of a taylor series about $\mathbb{E}_{P\beta} f_\beta(O)$ where

$$S_\beta(O) = \left(\frac{\partial m}{\partial\beta_0}, \frac{\partial m}{\partial\beta_1}, ..., \frac{\partial m}{\partial\beta_d}\right)^T (Y - expit\,(m(A, W|\beta))$$

a multivariate function mapping to $\mathbb{R}^d$. Also note that the derivative of the log-likelihood, $S_\beta(O)$ has mean 0 always. $(P_\beta S_\beta(O) = \mathbb{E}_{P_\beta} S_\beta(O) = 0)$.
Thus we get:

$$\begin{aligned}
P_n S_{\beta_n}(O) - P_\beta S_\beta(O) &= 0 \\
P_n S_{\beta_n}(O) - P_\beta S_\beta(O) + P_\beta S_{\beta_n}(O) - P_\beta S_{\beta_n}(O) &= 0 \\
(P_n - P_\beta) S_{\beta_n}(O) &= P_\beta\left(S_\beta(O) - S_{\beta_n}(O)\right) \\
\sqrt{n}(P_n - P) S_{\beta_n}(O) &= -\sqrt{n} P_\beta\left(\nabla_\beta S_\beta(O)\right)(\beta_n - \beta) + \sqrt{n} O_p \|\beta_n - \beta\|^2 \qquad (9) \\
\implies \sqrt{n}(\beta_n - \beta) &\overset{D}{\implies} \sqrt{n}(P_n - P_\beta)\left(-P_\beta\left(\nabla_\beta S_\beta(O)\right)^{-1} S_\beta(O)\right) \qquad (10)
\end{aligned}$$

since we can consider the $\|\beta_n - \beta\|^2$ term as second order. Therefore $\left(-P_\beta\left(\nabla_\beta S_\beta(O)\right)^{-1} S_\beta(O)\right)$ is the influence curve for the maximum likelihood estimator. In the case of logistic regression we have $m(A, W|\beta) = X^T\beta$ where $X^T = (1, A, W)^T$ and we consider β as a column vector of coefficients, including the intercept, β_0.

$$S_\beta(O) = X\,(Y - expit(\beta X))$$

$$\nabla_\beta S_\beta(O) = \begin{bmatrix} \nabla_\beta^T S_{\beta,0} \\ . \\ \nabla_\beta^T S_{\beta,d} \end{bmatrix} = expit(\beta X)(1 - expit(\beta X))XX^T$$

What is the influence curve for the estimator of $\bar{Q}(A, W)$ for a fixed (A, W) where the estimator is given by $\bar{Q}_{\beta_n}$?

The parameter $\Psi_{A_0 W_0}(P) = \bar{Q}_\beta(A_0, W_0)$ for fixed (A_0, W_0) is a continuously differentiable function of β, which means we can apply the ordinary delta method in the sense that

$$\begin{aligned} \bar{Q}_{\beta_n}(A_0, W_0) - \bar{Q}_\beta(A_0, W_0) &= \nabla_\beta^T \bar{Q}_\beta(A_0, W_0) IC_{\beta_n}(O) + R_2(P_\beta, P_{\beta_n}) \\ &= expit(\beta^T X_0)(1 - expit(\beta^T X_0))X_0^T P_\beta \left(\left(XX^T \right) expit(\beta^T X)(1 - expit(\beta^T X)) \right)^{-1} X \left(Y - expit(\beta^T X) \right) \\ &= expit(\beta^T X_0)(1 - expit(\beta^T X_0))X_0^T IC_{\beta_n} \end{aligned}$$

where due to the parametric model assumption, we can say $R_2 = o_p(n^{-.5})$.

Let $\tilde{Q}(W) = \bar{Q}(1, W) - \bar{Q}(0, W)$ for convenience. $IC_{\tilde{Q}(W_0)}$ is then just

$\left(expit(\beta^T(1, W_0))(1 - expit(\beta^T(1, W_0))(1, W_0)^T - expit(\beta^T(0, W_0))(1 - expit(\beta^T(0, W_0))(0, W_0)^T \right) IC_{\beta_n} = f_\beta(W_0) IC_{\beta_n}$

We can compute the influence curve of $\tilde{Q}(W_0)^2$ for a fixed W_0 as merely,

$$2\tilde{Q}(W_0) IC_{\tilde{Q}(W_0)}$$

by the ordinary delta method

Then it is easy to verify that the 2 dimensional influence curve for the maximum likelihood plug-in estimator of $\Psi(P) = (\Psi_1(P), \Psi_2(P)) = (ATE(P), blipVar(P))$ is just

$$\mathbb{E} \left(f_\beta, 2(\tilde{Q}(W) - \Psi_1(P)) f_\beta \right) IC_{\beta_n}$$

instructions Biometrics

Papers should be prepared with one-inch margins, in 12-point size letters and no more than 25 lines per page, double-spaced throughout. A one-paragraph summary should be included, followed by a list of key words, in alphabetical order. The summary should not exceed 225 words. The author's name should be followed by a full postal address and email address. Authors should use the [( )] convention in delimiting equations. To save space, display equations only if necessary. References should be typed in Biometrics style, and should be double-spaced throughout. Figures and tables should be separated from the main text, and placed at the end of the manuscript. Detailed algebraic derivations should be placed in an appendix. No footnotes should be used.

Biometrics has a limited number of journal pages. Normally, newly-submitted Biometric Methodology or Biometric Practice papers exceeding 25 pages and Reader Reaction papers exceeding 12 pages in the style described above will be returned to the authors without review. (These page counts include acknowledgements, references, and brief appendices, but not tables and figures. The page counts do not include the title page and abstract.) During the review process, it is common for Editors to request that papers be shortened, and authors should be aware that the typical accepted Biometrics paper is usually considerably shorter than 25 pages. It is also common for Editors to ask that most appendices be moved to Supplementary Web Materials. Authors are encouraged to move appendices and other appropriate content to Supplementary Web Materials at the time of submission in order to achieve a shorter main paper (the page count applies to the main paper only and not to Supplementary Materials).

Papers appearing in the journal rarely have more than six (6) tables or figures combined; about three-fourths have 4 or less. When papers contain numerous tables and figures, editors will always ask that the number be reduced or that some tables and figures be moved to Supplementary Materials. Authors are strongly encouraged to be judicious in the use of tabular and graphical displays and should not combine what ought to be several tables or figures into very large single ones. Authors should also consider moving some tables and figures to Supplementary Material at the time of submission. Papers with an extreme number of tables and/or figures may be returned by the co-editor without review. It is recognized that graphical depictions or images are essential for conveying the message in some substantive areas. In such circumstances, more figures than in the typical submission may be appropriate, and authors should note this explicitly in a covering letter.

References

Athey, Susan and Guido Imbens (2015). “Machine Learning Methods for Estimating Heterogeneous Causal Effects”. In: *arxiv.org*.

— (2016). “Recursive Partitioning for Heterogeneous Effects”. In: *Proceedings of the National Academy of Sciences of the USA* 113(27), pp. 7353–7360.

Benkeser, David and Mark van der Laan (2016). “The Highly Adaptive Lasso Estimator”. In: *Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics*, pp. 689–696.

Bitler, Marianne, Jonah B. Gelbach, and Hilary Williamson Hoynes (2014). “Can Variation is Subgroup’s Average Treatment Effects Explain Treatment Effect Heterogeneity?” In: *NBER Working Paper* w20142. URL: `https://ssrn.com/abstract=2438563`.

Chen, Tianqi et al. (2017). *xgboost: Extreme Gradient Boosting*. R package version 0.6-4. URL: `https://CRAN.R-project.org/package=xgboost`.

Coyle, Jeremy (2017). *opttx*. URL: `https://github.com/jeremyrcoyle/opttx`.

Coyle, Jeremy and Jonathan Levy (2017). *gentmle*. Berkeley, CA: University of California. URL: `https://github.com/jeremyrcoyle/gentmle2`.

Ding, Peng, Avi Feller, and Luke Miratrix (2016). “Randomization inference for treatment effect variation.” In: *J. R. Stat. Soc. B* 78, pp. 655–671.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2010). “Regularization Paths for Generalized Linear Models via Coordinate Descent”. In: *Journal of Statistical Software* 33(1), pp. 1–22. URL: `http://www.jstatsoft.org/v33/i01/`.

Hastie, Trevor (2017). *gam: Generalized Additive Models*. R package version 1.14-4. URL: `https://CRAN.R-project.org/package=gam`.

Hlavac, Mark (2015). *stargazer: Well-Formatted Regression and Summary Statistics Tables*. R package version 5.2. URL: `http://CRAN.R-project.org/package=stargazer`.

Laan, Mark van der (2016). “A Generally Efficient Targeted Minimum Loss Based Estimator”. In: *U.C. Berkeley Division of Biostatistics Working Paper Series* 343. URL: `http://biostats.bepress.com/ucbbiostat/paper343`.

Laan, Mark van der and Sandrine Dudoit (2003). "Unified Cross-Validation Methodology For Selection Among Estimators and a General Cross-Validated Adaptive Epsilon-Net Estimator: Finite Sample Oracle Inequalities and Examples". In: *U.C. Berkeley Division of Biostatistics Working Paper Series* 130.

Laan, Mark van der and Susan Gruber (2016). "One-Step Targeted Minimum Loss-based Estimation Based on Universal Least Favorable One-Dimensional Submodels". In: *The International Journal of Biostatistics* 12(1), pp. 351–378.

Laan, Mark van der, Eric C. Polley, and Alan Hubbard (2007). "Super Learner". In: *U.C. Berkeley Division of Biostatistics Working Paper Series* 6(1).222.

Laan, Mark van der and Sherri Rose (2011). *Targeted Learning*. New York: Springer.

Laan, Mark van der and Daniel Rubin (2006). "Targeted Maximum Likelihood Learning". In: *U.C. Berkeley Division of Biostatistics Working Paper Series* 213. URL: http://biostats.bepress.com/ucbbiostat/paper213.

LeDell, Erin (2017). *h2oEnsemble: H2O Ensemble Learning*. R package version 0.2.1. URL: https://github.com/h2oai/h2o-3/tree/master/h2o-r/ensemble/h2oEnsemble-package.

Luedtke, Alex and Mark van der Laan (2016). "Super-Learning of an Optimal Dynamic Treatment Rule". In: *International Journal of Biostatistics* 12(1).305-332.

Milborrow, Stephen (2017). *earth: Multivariate Adaptive Regression Splines*. Derived from mda:mars by Trevor Hastie and Rob Tibshirani. Uses Alan Miller's Fortran utilities with Thomas Lumley's leaps wrapper. R package version 4.5.0. URL: https://CRAN.R-project.org/package=earth.

Muñoz ID, Mark van der Laan (2012). "Population Intervention Causal Effects Based on Stochastic Interventions". In: *Biometrics* 68(2).541-549.

Pearl, Judea (2000). *Causality*. Cambridge University Press, p. 484.

Polley, Eric C. and Mark van der Laan (2009). "Selecting Optimal Treatments Based on Predictive Factors". In: *U.C. Berkeley Division of Biostatistics Working Paper Series* 244.

Polley, Eric C., Erin LeDell, et al. (2017). *SuperLearner: Super Learner Prediction*. R package version 2.0-23-9000. URL: https://github.com/ecpolley/SuperLearner.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. URL: https://www.R-project.org/.

Therneau, Terry, Beth Atkinson, and Brian Ripley (2017). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-11. URL: https://CRAN.R-project.org/package=rpart.

Tian, Lu et al. (2014). “A Simple Method for Estimating Interactions between a Treatment and a Large Number of Covariates”. In: *Journal of the American Statistical Association* 109(508), pp. 1517–1532.

Vaart, Aad van der (2000). *Asymptotic Statistics*. Vol. Chapter 25. Cambridge, UK: Cambridge University Press.

Vaart, Aad van der and Jon A. Wellner (1996). *Weak Convergence and Empirical Processes*. New York: Springer-Verlag.

Venbles, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S*. Fourth. ISBN 0-387-95457-0. New York: Springer. URL: `http://www.stats.ox.ac.uk/pub/MASS4`.

Wickham, Hadley (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Wright, Marvin N. and Andreas Ziegler (2017). *ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R*.

Zheng, Wenjing and Mark van der Laan (2010). “Asymptotic Theory for Cross-validated Targeted Maximum Likelihood Estimation”. In: *U.C. Berkeley Division of Biostatistics Working Paper Series* 273.