Computational Considerations for Targeted Learning

by

Jeremy Robert Coyle

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Biostatistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alan Hubbard, Chair
Professor Mark van der Laan
Professor Barbara Abrams

Spring 2017

Computational Considerations for Targeted Learning

Abstract

Computational Considerations for Targeted Learning

by

Jeremy Robert Coyle

Doctor of Philosophy in Biostatistics

University of California, Berkeley

Professor Alan Hubbard, Chair

Targeted Learning represents a principled methodology that has the potential to leverage the availability of big datasets and large scale computing facilities. However, many of the methods are computationally demanding, and therefore require careful consideration as to their implementation. This thesis comprises three cases studies at the intersection between Targeted Learning and computation. Chapter 1 describes the Targeted Bootstrap, a novel bootstrap technique that samples from a TMLE distribution and therefore has asymptotic performance guarantees, while avoiding issues related to cross-validation on bootstrap samples. Chapter 2 considers the problem of estimating both a target parameter and nuisance parameter on which it depends, when ideally both would be estimated with cross-validation. By carefully considering what parts of the sample are used for what estimation tasks, nested cross-validation can be avoided at great computational savings. This is achieved using the novel SplitSequential cross-validation approach. Chapter 3 describes the `opttx` package for learning optimal treatment rules. This package contains an implementation of SplitSequential Super Learner, and also contains a novel approach to learning an optimal rule for a categorical treatment variable. Further, performance-based variable importance measures are used to evaluate which of the covariates are most useful for making treatment decisions.

Contents

List of Figures

List of Tables

Acknowledgments

The work represented by this dissertation was by enabled many generous organizations and individuals, without whose support it would not have been possible. My advisors Professor Alan Hubbard, Mark van der Laan, Barbara Abrams, and Nick Jewell have of course made countless substantiative suggestions and contributions to my research. Perhaps more significantly, throughout my graduate career, they have been incredibly encouraging, supportive, and patient. I would also like to thank the following organizations for funding my PhD training: National Institutes of Health, under grant R01MD006014, and PCORI ME-1306-02735.

I am grateful for the community of the Biostatistics Division. My fellow graduate students have been a been a constant source of support and encouragement. I would like to specifically thank Alex Luedtke, Lucia Petito, Sara Moore, Anna Decker, Oleg Sofrygin, Kathy Evans, Molly Davies, Jonathan Levy, and Jade Benjamin-Chung for helpful discussions and collaboration. I would also like to thank Department Administrator Sharon Norris, who is an amazing asset to the department with her positivity, knowledge, and ability to cut through red tape. In addition to collaborations within biostatistics, have been fortunate to have had many fruitful collaborations outside of biostatistics. I would especially like to thank my colleagues and coauthors: Danny Wilson, Jason Trager, Jonathon Harper, Marc Pare, Angeli Kirk, and Ajay Pillarisetti.

Writing my dissertation has been an intense challenge. Throughout this process, the support of my broader community of family and friends has been invaluable. I would like to thank the bay area cycling community for providing lasting friendships and a much needed outlet for academic stress. Also thanks to the Bowen family, who have taken me in as one of their own. Finally, thanks to Jessica who has stood by me with caring, understanding, and grace throughout this process.

Chapter 1

Targeted Bootstrap for the Sampling Distribution of an Asymptotically Linear Estimator

1.1 Introduction

The bootstrap is used to obtain statistical inference (confidence intervals, hypothesis tests) in a wide variety of settings [Efron and Tibshirani, 1993, Davison and Hinkley, 1997]. Bootstrap-based confidence intervals have been shown in some settings to have higher-order accuracy compared to Wald-style intervals based on the normal approximation [Hall, 1988, DiCiccio and Romano, 1988, Hall, 1992]. For this reason it has been widely adopted as a method for generating inference in a range of contexts, not all of which have theoretical support. One setting in which it fails to work in the manner it is typically applied is in the framework of Targeted Learning. We describe the reasons for this failure in detail and present a solution in the form of a Targeted Bootstrap, designed to be consistent for the first two moments of the sampling distribution.

Suppose we want to estimate a particular pathwise differentiable parameter using a Targeted Learning approach [van der Laan and Rose, 2011]. The typical workflow is to obtain initial estimates for relevant factors of the likelihood using Super Learner [van der Laan et al., 2007a], and then generate a targeted estimate using TMLE. By using Super Learner and TMLE, we can generate correct inference for our parameter of interest without assuming that the likelihood can be modeled by simple parametric models. Relying on the fact that TMLE is an asymptotically linear estimator, we can use the normal approximation to generate Wald style confidence intervals where the standard error is based on the influence curve. These confidence intervals are first-order accurate. It is tempting to instead obtain higher-order correct confidence intervals by applying the non-parametric bootstrap. However, in the case of TMLE with initial estimates obtained via the Super Learner algorithm, naïve application of the nonparametric bootstrap is not justified and which we will shown to have poor performance, because Super Learner and therefore TMLE behaves differently on nonparametric bootstrap samples than it does on samples from the true data generating distribution. It is therefore important to develop a bootstrap method that works in the context of Targeted Learning.

We illustrate the reason for this difference in Super Learner's behavior, and present a solution in the form of the Targeted Bootstrap, a novel model based bootstrap that samples from a distribution targeted to a parameter of interest and the asymptotic variance of estimators of that parameter. In the process, we outline a TMLE that targets both a parameter of interest and its asymptotic variance. This TMLE can be used to generate another Wald style confidence interval, by directly using the targeted estimate of the asymptotic variance. Additionally, it can be used to generate a confidence of interval for the asymptotic variance itself. We demonstrate the practical performance of the targeted bootstrap confidence intervals relative to the Wald-type confidence intervals as well as confidence intervals generated by other bootstrap approaches.

The remainder of this paper is organized as follows: First, we state the problem and describe the example of estimating the Treatment Specific Mean (TSM). This example will be used in the remainder of the exposition, with a more general presentation given in the Appendix. Section 1.2 gives a review of TMLE and presents the TMLE for the Treatment Specific Mean as well as TMLEs targeting asymptotic variance in general and specifically for the TSM parameter. Section 1.3 reviews loss based estimation, including Super Learner. Section 1.4 reviews bootstrap theory and section 1.4 presents the novel Targeted Bootstrap approach. Section 1.5 presents simulation results demonstrating the performance of Targeted Bootstrap.

Problem Statement

Suppose that we observe n independent and identically distributed copies of O with probability distribution P_0 known to be an element of the statistical model $\mathcal{M}$. In addition, assume we are concerned with statistical inference of the target parameter value $\psi_0 = \Psi(P_0)$ for a given parameter mapping $\Psi : \mathcal{M} \to I\!R$. Consider a given estimator $\hat{\Psi} : \mathcal{M}_{np} \to I\!R$ that maps an empirical distribution P_n of $O_1, \ldots, O_n$ into an estimate of ψ_0, and assume that this estimator $\psi_n = \hat{\Psi}(P_n)$ is asymptotically linear at P_0 with influence curve $O \to D(P_0)(O)$ at P_0, so that we can write:

$$\psi_n - \psi_0 \;=\; (P_n - P_0)D(P_0) + o_P(1/\sqrt{n})$$

In that case, we have that $\sqrt{n}(\psi_n - \psi_0)$ converges in distribution to a normal distribution $N(0, \Sigma^2(P_0))$, where $\Sigma^2 : \mathcal{M} \to I\!R$ is defined by $\Sigma^2(P) = PD(P)^2$ as the variance of the influence curve $D(P)$ under P.

We wish to estimate a confidence interval for ψ_n. A one sided confidence interval is defined by a quantity $\psi_{n,[\alpha]}$ such that $P_0(\psi_0 < \psi_{n,[\alpha]}) = \alpha$. Two sided confidence intervals are typically equal-tailed intervals, having the same error in each tail: $P_0(\psi_0 < \psi_{n,[\alpha/2]}) = P_0(\psi_0 > \psi_{n,[1-\alpha/2]}) = \alpha$. These can be constructed using a pair of one-sided intervals.

A one-sided Wald confidence interval can be generated using the asymptotic normality discussed above: defining variance estimator $\hat{\sigma}_n^2 = \Sigma^2(P_n)$, the endpoint is $\psi_{n,[\alpha],\text{Wald}} = \psi_n - n^{-1/2}\hat{\sigma}_n \phi^{-1}(1-\alpha)$, where $\phi^{-1}(1-\alpha)$ is the $1-\alpha$th quantile of the standard normal distribution. This approach ignores the remainder term $o_P(1/\sqrt{n})$ and is therefore said to be first order correct.

Example: Treatment Specific Mean

To provide a concrete motivating example, suppose we observe n i.i.d. observations of $O = (W, A, Y) \sim P_0$, for baseline covariates W, treatment $A \in \{0, 1\}$, and outcome $Y \in \{0, 1\}$, and suppose that $\mathcal{M}$ is the nonparametric model, making no assumptions about the distribution from which O is sampled. The target parameter $\Psi : \mathcal{M} \to \mathbb{R}$, is a Treatment Specific Mean (TSM) defined as $\Psi(P) = E_P E_P(Y \mid A = 1, W)$. Let $\bar{Q}(P)(W) = E_P(Y \mid A = 1, W)$ and $\bar{g}(P)(W) = E_P(A \mid W)$.

1.2 TMLE

Targeted minimum loss-based estimation (TMLE) is an estimation framework that produces asymptotically linear substitution estimators of target parameters [van der Laan and Rubin, 2006, van der Laan and Rose, 2011]. TMLE fluctuates an initial estimate of the target parameter, resulting in an estimate which makes the correct bias-variance trade-off. TMLE estimators are asymptotically linear with a known influence curve, even when the components of the likelihood are estimated using data-adaptive methods (like these).

TMLE Definition

A TMLE is defined by four components. First, the parameter, $\Psi(P)$ which maps $\mathcal{M} \to \mathbb{R}$. For example, the TSM parameter described above: $\Psi(P_0) = E_{0,W}[E_{0,Y}[Y|A = 1, W]]$. Second, an efficient influence curve, $D^*(P_0)(O)$, which defines the estimator with the minimal asymptotic variance. Third, a submodel for relevant factor(s) of likelihood, $P^*_{n,\epsilon}(O)$. Finally, a loss function, $L(O_i, P^j_{n,\epsilon})$, with which to evaluate the submodel. Minimizing the loss in the submodel is equivalent to setting $P_n D(P^*_n) = 0$, which solves the efficient influence curve.

TMLE for Treatment Specific Mean

The efficient influence curve of Ψ at P is given by:

$$D^*(P)(O) = \frac{A}{\bar{g}(W)}(Y - \bar{Q}(W)) + \bar{Q}(W) - \Psi(P).$$

[van der Laan and Robins, 2003]. Note that $\Psi(P)$ only depends on P through $\bar{Q}(P)$ and the probability distribution $Q_W(P)$ of W. Let $Q(P) = (Q_W(P), \bar{Q}(P))$ and let $Q(\mathcal{M}) = \{Q(P) : P \in \mathcal{M}\}$ be its model space. We will also denote the target parameter as $\Psi : Q(\mathcal{M}) \to I\!R$ as a mapping that maps a Q in the parameter space $Q(\mathcal{M})$ into a numeric value, abusing notation by using the same notation Ψ for this mapping. Similarly, we will also denote $D^*(P)$ with $D^*(Q, G)$. The efficient influence curve $D^*(P)$ satisfies the expansion $\Psi(P) - \Psi(P_0) = -P_0 D^*(P) + R_\psi(P, P_0)$, where

$$R_\psi(P, P_0) = P_0 \frac{\bar{g} - \bar{g}_0}{\bar{g}}(\bar{Q} - \bar{Q}_0).$$

Let $\psi_n^* = \Psi(Q_n^*)$ be a TMLE of ψ_0 so that it is asymptotically linear at P_0 with influence curve $D^*(P_0)$. This TMLE can be defined by letting $\bar{Q}_n^0$ being an initial estimator of $\bar{Q}_0$, $\bar{g}_n$ an estimator of g_0, $L(\bar{Q})(O) = -I(A = 1)(Y \log \bar{Q}(W) + (1 - Y) \log(1 - \bar{Q}(W))$ being the log-likelihood loss function for $\bar{Q}_0$, the submodel $\text{Logit}\bar{Q}_n^0(\epsilon)) = \text{Logit}\bar{Q}_n^0 + \epsilon H(\bar{g}_n)$ with $H(\bar{g}_n) = A/\bar{g}_n(W)$, $\bar{Q}_n^1 = \bar{Q}_n^0(\epsilon_n^0)$ with $\epsilon_n^0 = \arg\min_\epsilon P_n L(\bar{Q}_n^0(\epsilon))$, and $\psi_n^* = \Psi(Q_n^1)$, where $Q_n^1 = (\bar{Q}_n^1, Q_{W,n})$ and $Q_{W,n}$ is the empirical distribution of $W_1, \ldots, W_n$. Let P_n^* be a probability distribution compatible with Q_n^*.

TMLE for the Asymptotic Variance of a Target Parameter

Let $O \sim P_0 \in \mathcal{M}$, and we have two target parameter $\Psi : \mathcal{M} \to I\!R$ and $\Sigma^2 : \mathcal{M} \to I\!R$. We are given an estimator ψ_n^* that is asymptotically linear at P_0 with influence curve $D(P_0)$. For simplicity, we will consider the case that $\psi_n^* = \Psi(P_n^*)$ is an efficient targeted maximum likelihood estimator so that $D(P) = D^*(P)$ and $D^*(P)$ is the efficient influence curve of Ψ at P.

In this case, $\Sigma^2(P) = P\{D^*(P)\}^2$. Let $D^*_\Sigma(P)$ be the efficient influence curve of Σ^2 at P. Suppose that $\Sigma^2(P) = \Sigma_1^2(Q_\Sigma(P))$ for some parameter $Q_\Sigma(P)$ that can be defined by minimizing the risk of a loss function $L_\Sigma(Q_\Sigma)$ so that $Q_\Sigma(P) = \arg\min_{Q_\Sigma} PL_\Sigma(Q)$. In addition, we assume that $D^*_\Sigma(P)$ only depends on P through $Q_{\Sigma(P)}$ and some other parameter $g_\Sigma(P)$. For notational convenience, we will denote these to alternative representations of the asymptotic variance parameter and its efficient influence curve with $\Sigma^2(Q_\Sigma)$ and $D^*_\Sigma(Q_\Sigma(P), g_\Sigma(P))$ respectively. We now develop a TMLE of $\Sigma^2(P_0)$ as follows. First, let $Q^0_{\Sigma,n}$ be an initial estimator of $Q_\Sigma(P_0)$, which could be based on the super-learner ensemble algorithm using the loss function $L_\Sigma()$. Similarly, let $g_{\Sigma,n}$ be an estimator of $g_{\Sigma,0}$. Set $k = 0$. Consider now a submodel $\{Q^k_{\Sigma,n}(\epsilon \mid g_{\Sigma,n}) : \epsilon\} \subset Q_\Sigma(\mathcal{M})$ so that the linear span of the components of the generalized score $\frac{d}{d\epsilon} L_\Sigma(Q^k_{\Sigma,n}(\epsilon \mid g_{\Sigma,n}))$ at $\epsilon = 0$ spans $D^*_\Sigma(Q^k_{\Sigma,n}, g_{\Sigma,n})$. Define $\epsilon_n^k = \arg\min_\epsilon P_n L_\Sigma(Q^k_{\Sigma,n}(\epsilon \mid g_{\Sigma,n}))$ as the MLE and define the update $Q^{k+1}_{\Sigma,n} = Q^k_{\Sigma,n}(\epsilon_n^k \mid g_{\Sigma,n})$. We iterate this updating process till convergence at which step K we have $\epsilon_n^K \approx 0$. We denote this final update with $Q^*_{\Sigma,n}$ and we call that the TMLE of $Q_\Sigma(P_0)$, while $\Sigma^2(Q^*_{\Sigma,n})$ is the TMLE of the asymptotic variance $\Sigma^2(Q_0)$ of the TMLE ψ_n^* of ψ_0. Let $\tilde{P}_n^*$ be a probability distribution compatible with $Q^*_{\Sigma,n}$.

Application to the Treatment Specific Mean

The asymptotic variance of $\sqrt{n}(\psi_n^* - \psi_0)$ is given by:

$$\begin{aligned}\Sigma^2(P_0) &= E_{P_0}\{D^*(P_0)\}^2 \\ &= Q_{0,W}\left(\frac{\bar{Q}_0(1-\bar{Q}_0)}{\bar{g}_0} + (\bar{Q}_0 - Q_{0,W}\bar{Q}_0)^2\right)\end{aligned}$$

The following lemma presents its efficient influence curve $D^*_\Sigma(P)$.

Lemma 1. *The efficient influence curve $D^*_\Sigma(P)$ of Σ^2 at P is given by:*

$$D^*_\Sigma(P)(W,A,Y) = D_{\Sigma^2,Q_W}(P)(W) + D_{\Sigma^2,\bar{Q}}(P)(O) + D_{\Sigma^2,\bar{g}}(P)(O),$$

where

$$\begin{aligned}D_{\Sigma^2,Q_W}(P)(W) &= \frac{\bar{Q}(1-\bar{Q})}{\bar{g}}(W) - Q_W\frac{\bar{Q}(1-\bar{Q})}{\bar{g}} \\ &\quad +(\bar{Q}(W) - \Psi(Q))^2 - Q_W(\bar{Q} - \Psi(Q))^2 \\ D_{\Sigma^2,\bar{Q}}(P)(O) &= \frac{I(A=1)}{\bar{g}(W)}\left(\frac{1-2\bar{Q}(W)}{\bar{g}(W)} + 2(\bar{Q}(W) - \Psi(Q))\right)(Y - \bar{Q}(W)) \\ D_{\Sigma^2,\bar{g}}(P)(O) &= -\frac{\bar{Q}(1-\bar{Q})(W)}{\bar{g}^2(W)}(A - \bar{g}(W)).\end{aligned}$$

This allows us to develop a TMLE $\Sigma^2(Q_{W,n}, \bar{Q}_n^*, \bar{g}_n^*)$ of $\Sigma^2(Q_{W,0}, \bar{Q}_0, \bar{g}_0)$. Define the clever covariates:

$$\begin{aligned}C_Y(\bar{g}, Q)(A,W) &\equiv \frac{I(A=1)}{\bar{g}(W)}\left(\frac{1-2\bar{Q}(W)}{\bar{g}(W)} + 2(\bar{Q}(W) - \Psi(Q))\right) \\ C_A(\bar{g}, \bar{Q})(W) &\equiv \frac{\bar{Q}(1-\bar{Q})(W)}{\bar{g}^2(W)}.\end{aligned}$$

Let $Q_n^0 = (Q_{W,n}, \bar{Q}_n^0)$ for an initial estimator $\bar{Q}_n^0$ of $\bar{Q}_0$, where $Q_{W,n}$ is the empirical distribution which will not be changed by the TMLE algorithm. Let $k = 0$. Consider the submodels

$$\begin{aligned}\text{Logit}\bar{Q}_n^k(\epsilon_1) &= \text{Logit}\bar{Q}_n^k + \epsilon_1 C_Y(\bar{g}_n^k, Q_n^k) \\ \text{Logit}\bar{g}_n^k(\epsilon_2) &= \text{Logit}\bar{g}_n^k + \epsilon_1 C_A(\bar{g}_n^k, \bar{Q}_n^k).\end{aligned}$$

In addition, consider the log-likelihood loss functions

$$\begin{aligned}L_1(\bar{Q}) &= -I(A=1)\left\{Y\log\bar{Q}(W) + (1-Y)\log(1-\bar{Q}(W))\right\} \\ L_2(\bar{g}) &= -\left\{A\log\bar{g}(W) + (1-A)\log(1-\bar{g}(W))\right\}\end{aligned}$$

Define the MLEs $\epsilon^k_{1n} = \arg\min_\epsilon P_n L_1(\bar{Q}^k_n(\epsilon))$ and $\epsilon^k_{2n} = \arg\min_\epsilon P_n L_2(\bar{g}^k_n(\epsilon))$. This defines now the first step update $\bar{Q}^{k+1}_n = \bar{Q}^k_n(\epsilon^k_{1n})$ and $\bar{g}^{k+1}_n = \bar{g}^k_n(\epsilon^k_{2n})$. Now set $k = k+1$ and iterate this process till convergence defined by $(\epsilon^*_{1n}, \epsilon^*_{2n})$ being close enough to $(0,0)$. Let $\bar{g}^*_n, \bar{Q}^*_n$ denote these limits of this TMLE procedure, and let $Q^*_n = (Q_{W,n}, \bar{Q}^*_n)$. The TMLE of $\Sigma^2(P_0)$ is given by $\Sigma^2(\tilde{P}^*_n)$ where $\tilde{P}^*_n$ is defined by $(Q_{W,n}, \bar{Q}^*_n, \bar{g}^*_n)$. We note that at $(\epsilon^*_{1n}, \epsilon^*_{2n} = (0,0)$, we have

$$0 = P_n D_{\Sigma^2}(\tilde{P}^*_n) = 0,$$

and if the algorithm stops earlier at step K, and one defines $\tilde{P}^*_n = P^K_n$, one just needs to make sure that

$$P_n D_{\Sigma^2}(\tilde{P}^*_n) = o_P(1/\sqrt{n}).$$

Joint TMLE of both the Target Parameter and its Asymptotic Variance.

We could also define a TMLE targeting both parameters Ψ and Σ^2. This is defined exactly as above, but now using a submodel $\{P^k_n(\epsilon) : \epsilon\} \subset \mathcal{M}$ that has a score $\frac{d}{d\epsilon} L(P^k_n(\epsilon))$ at $\epsilon = 0$ whose components span both efficient influence curves $(D^*_\psi(P), D^*_\Sigma(P))$. In this manner, one obtains a TMLE $\tilde{P}^*_n$ that solves $P_n D^*_\psi(\tilde{P}^*_n) = P_n D^*_\Sigma(\tilde{P}^*_n) = 0$ and, under regularity conditions, yields an asymptotically efficient estimator of both ψ_0 and σ^2_0. In this case our TMLE of ψ_0 could just be $\Psi(\tilde{P}^*_n)$: so in this special case, we have $P^*_n = \tilde{P}^*_n$.

In particular, we could estimate both ψ_0 and σ^2_0 with a bivariate TMLE $(\Psi(\tilde{P}^*_n), \Sigma^2(\tilde{P}^*_n))$ where $\tilde{P}^*_n$ is a TMLE that targets both ψ_0 and $\sigma^2_0 = \Sigma^2(P_0)$. In this case, $P^*_n = \tilde{P}^*_n$ and thus $\psi^*_n = \Psi(\tilde{P}^*_n)$, $\sigma^*_n = \Sigma^2(\tilde{P}^*_n)$. This TMLE can be defined as the above iterative TMLE of $\Sigma^2(P_0)$ but now using the augmented submodel:

$$\text{Logit}\bar{Q}^k_n(\epsilon_1) \;\; = \;\; \text{Logit}\bar{Q}^k_n + \epsilon_0 H(\bar{g}^k_n) + \epsilon_1 C_Y(\bar{g}^k_n, Q^k_n),$$

where $H(\bar{g})(A,W) = I(A=1)/\bar{g}(W)$. Conditions under which $(\Psi(\tilde{P}^*_n), \Sigma^2(\tilde{P}^*_n))$ is an asymptotically efficient estimator of $(\Psi(P_0), \Sigma^2(P_0))$ are given in theorem 6.

1.3 Super Learner

TMLE requires initial estimates of factors of the likelihood. For the treatment specific mean example discussed in section 1.1 and section 1.2, we need estimates of $\bar{Q}(A,W)$ and $\bar{g}(W)$. In the targeted learning framework, these factors are typically estimated with Super Learner [van der Laan et al., 2007a]. Super Learner is a model selection/model stacking algorithm based on minimizing cross-validation based estimates of risk amongst a set of candidate learners [van der Laan et al., 2007a, Polley and van der Laan, 2010]. For the purposes of this paper, we applied Discrete Super Learner, which selects the risk-minimizing individual algorithm. A more flexible Super Learner, which selects a risk-minimizing convex combination of learners is described in van der Laan et al. [2007a]. We will outline the underlying loss based estimation framework and cross-validation approach that provides Super Learner with its theoretical justification. This framework is covered in much greater detail in Dudoit and van der Laan [2005], van der Laan and Dudoit [2003b], Vaart et al. [2006].

Loss Based Estimation

Loss based estimation is a framework that allows us to objectively evaluate the quality of estimates and select amongst competing estimators based on this evaluation. Super Learner is a particular implementation of this framework, and understanding this framework is important to understanding the behavior of Super Learner on nonparametric bootstrap samples. In the context of Super Learner, we will refer to estimators of the factors of the likelihood as "learners". This exposition will focus on the example of learning an estimate of $\bar{Q}_0(P_0) = E_{P_0}[Y|A,W]$, but it applies equally well to other factors of the likelihood. Here, $\bar{Q}(P)$ indicates an estimate of $\bar{Q}$ based on P. Consider the problem of selecting an estimate $\bar{Q}$ from a class of possible distributions $\bar{\mathbf{Q}}$. In the context of discrete Super Learner, this becomes selecting from a set of candidate learners: $\bar{Q}_k : k = 1, ..., K$.

The key ingredient in this framework is a loss function, $L(\bar{Q}, O)$, that describes the severity of the difference between a value predicted by a learner and the true observed value. For example, the squared error loss: $L(\bar{Q}, O) = (\bar{Q} - Y)^2$. This leads to the risk, which is the expected value of the loss with respect to distribution P: $\theta(\bar{Q}, P) = PL(\bar{Q}, O) = E_{P_0}[L(\bar{Q}, O)]$. Evaluated at the truth, P_0, we get the true risk $\theta_0(\bar{Q}) = \theta(\bar{Q}, P_0)$, which provides a criteria by which to select a learner: $\bar{Q}_0 = \arg\min_{\bar{Q} \in \bar{\mathbf{Q}}} \theta_0(\bar{Q}) = \bar{Q}(P_0)$, the learner we want is the one that minimizes the true risk. Crucially, this should be equal to the parameter we're trying to estimate, here $\bar{Q}_0(A, W)$. The value of this risk at the minimizer is called the optimal risk, or the irreducible error: $\theta_0(\bar{Q}_0) = \min_{\bar{Q} \in \bar{\mathbf{Q}}} \theta_0(\bar{Q})$. Then, in the context of discrete Super Learner, the we can define oracle selector as $\tilde{k}_n = \arg\min_{k \in \{1,...,K_n\}} \theta_0(\bar{Q}_k)$ selects the learner that minimizes true risk amongst set of candidates. This is the learner we would select if we knew P_0

Empirical Risk Estimate

The empirical or resubstitution risk estimate, $\hat{\theta}_{P_n}(\bar{Q}(P_n)) = \theta(\bar{Q}(P_n), P_n)$, estimates the risk on the same dataset used to train the learner. This is known to be optimistic (biased downwards) in most circumstances, with the optimism increasing as a function of model complexity. Therefore, the resubstitution selector $\arg\min_{k \in \{1,...,K_n\}} \hat{\theta}_{P_n}(\bar{Q}_k(P_n))$ selects learners which "overfit" the data, selecting learners which are unnecessarily complex, and therefore have a higher risk than models which make the correct bias-variance trade-off. Hastie et al. [2003] discusses this phenomenon in more detail.

Cross-Validation

Cross-validation allows more accurate risk estimates that are not biased towards more complex models. Our formulation relies on a split vector $B_n \in \{0,1\}^n$, which divides the data into two sets, a training set $(O_i : B_n(i) = 0)$, with the empirical distribution P^0_{n,B_n}, and a validation set $(O_i : B_n(i) = 1)$, with the empirical distribution P^1_{n,B_n}. Averaging over the distribution of B_n, yields a cross-validated risk estimate: $\hat{\theta}_{\mathrm{CV}}(\bar{Q}) = E_{B_n}\theta(\bar{Q}(P^0_{n,B_n}), P^1_{n,B_n})$. This yields a cross-validated selector $\hat{k}_n = \arg\min_{k\in\{1,\ldots,K_n\}} \theta_{\mathrm{CV}}(\bar{Q})$, which selects the learner that minimizes the cross-validated risk estimate. Because cross-validation uses separate data for training and risk estimation for each split vector B_n, it has an important oracle property. Under appropriate conditions the CV selector will do asymptotically as well as the oracle selector in terms of a risk difference ratio:

$$\frac{\theta_0(Q_{\hat{k}}) - \theta_0(\bar{Q}_0)}{\theta_0(Q_{\tilde{k}}) - \theta_0(\bar{Q}_0)} \xrightarrow{P} 1 \tag{1.1}$$

$$\tag{1.2}$$

That is, the ratio of the risk difference between the cross-validated selector and the optimal risk and the risk difference between the oracle selector and the optimal risk approaches 1 in probability. Conditions and proofs for this result are given in Dudoit and van der Laan [2005], van der Laan and Dudoit [2003b], Vaart et al. [2006]. It is through this property that Discrete Super Learner does asymptotically as well as the best of its candidate learners. We will soon describe how this property fails for nonparametric bootstrap samples.

1.4 Bootstrap

Before presenting our generalization of the bootstrap, we briefly review the bootstrap theoretical framework. The key idea of the bootstrap is as follows: we wish to estimate the sampling distribution of an estimator $G(x) = P(\psi_n \leq x)$, or more typically its existing moments or quantiles. It is difficult to estimate this distribution directly because because we only observe one sample from P_0, and therefore only one realization of ψ_n. However, we can draw B repeated samples of size n from some estimate of P_0 and apply our estimator to those samples. Denoting such a sample $O^\#_1, \ldots, O^\#_n$ and the empirical distribution corresponding to that sample $P^\#_n$ and estimate $\psi^\#_n = \Psi(P^\#_n)$, we can obtain an estimate of the desired sampling distribution:

$$\hat{G}(x) \;=\; \frac{1}{B}\sum_{i=1}^{B} I(\psi^\#_{n,i} \leq x)$$

Nonparametric Bootstrap

The nonparametric bootstrap applies this approach by sampling from the empirical distribution, P_n. This approach has been demonstrated to be an effective tool for estimating the sampling distribution in a wide range of settings.

However, the nonparametric bootstrap is not universally appropriate for sampling distribution estimation. Because P_n is a discrete distribution, repeated sampling from it will create "copied" observations – bootstrap samples will have more than one identical observation in a sample. Bickel et al. [1997] notes that the bootstrap can fail if the estimator is sensitive to ties in the dataset. One example of a class of estimators that may be sensitive to ties are those that use cross-validation to make a bias-variance trade-off. If cross-validation is applied to a non-parametric bootstrap sample, duplicate observations have the potential to appear in both the training and testing portions of a given sample split. This creates an issue for estimators that rely on cross-validation. Hall [1992] specifically notes issue of ties for cross-validation based model selection.

The severity of this problem is determined by how many copied observations we can expect. For a bootstrap sample of size n, and validation proportion p_{B_n}, the probability of a validation observation having a copy in the training sample is given by:

$$p(\text{copy}) \quad = \quad 1 - \left(1 - \frac{1}{n}\right)^{(1-p_{B_n})n} \approx 1 - e^{-(1-p_{B_n})}$$

For 10-fold cross-validation $p_{B_n} = 0.1$, so we can expect ≈59% of validation observations to also be in the training sample. An average of 59% of a CV-risk estimate on a bootstrap sample is therefore something like a resubstitution risk estimate, therefore having the same suboptimal properties described in section 1.3.

One ad-hoc solution to the problem of duplicate observations for cross-validation is to do "clustered" cross-validation where a cluster is defined as a set of identical bootstrap observations, and then split the clusters between training and validation. This way, no observation will appear in both the training and testing sets. Although we lack rigorous justification for this approach, it was evaluated in the simulation study below. It appears in the results as "NP Bootstrap + CVID".

Model Based Bootstrap

In contrast, the parametric bootstrap draws samples from an estimate of P_0 based on an assumed parametric model: $P_{n,\beta}$. The parametric bootstrap can be generalized to a "model-based" bootstrap that using semi- or non-parametric estimates of factors of the likelihood. In the context of our treatment specific mean example, this means using estimates of $P(Y|A,W)$ and $\bar{g}$. If Y is binary, as is the case in our simulation, $P(Y=1|A,W) = E(Y|A,W) = \bar{Q}(A,W)$. Otherwise, we need an estimate of the distribution of $\epsilon(A,W)$ such that $Y = E(Y|A,W) + \epsilon(A,W)$. To be explicit, observations are drawn from an estimate $\tilde{P}_n = P_n(Y|A,W)\bar{g}_n Q_W(P_n)$ according to the following algorithm:

Algorithm 1: Model-Based Bootstrap

```
Sample $W^{\#}$ from the empirical distribution of W: $Q_W(P_n)$
Using $W^{\#}$, sample $A^{\#}$ from $\bar{g}(W^{\#})$
Using $A^{\#}$ and $W^{\#}$, sample $Y^{\#}$ from $P_n(A^{\#}, W^{\#})$
```

The Targeted Bootstrap, described below, is a particular model-based bootstrap using estimates of $\bar{Q}_n^*$ $\bar{g}_n^*$ targeted to ensure correct asymptotic performance.

Targeted Bootstrap

The idea of Targeted Bootstrap is to construct a targeted maximum likelihood estimator $\tilde{P}_n^*$ so that $\Sigma^2(\tilde{P}_n^*)$ is a TMLE of $\sigma_0^2 = \Sigma^2(P_0)$. Then we know that under regularity conditions, $\Sigma^2(\tilde{P}_n^*)$ is an asymptotically linear and efficient estimator of σ_0^2 at P_0 so that we can construct a confidence interval for σ_0^2. In addition, since it is a substitution estimator of σ_0^2 it is often more reliable in finite samples resulting in potential finite sample improvements in coverage of the confidence interval. In addition, we will show that under appropriate regularity conditions, due to the consistency of $\Sigma^2(\tilde{P}_n^*)$, the bootstrap distribution of $\sqrt{n}(\hat{\Psi}(P_{n,\#}) - \Psi(\tilde{P}_n^*))$ based on sampling $O_1^{\#}, \ldots, O_n^{\#} \sim_{iid} \tilde{P}_n^*$, given almost every $(P_n : n)$, $\sqrt{n}(\hat{\Psi}(P_n) - \psi_0)$ converges to the desired limit distribution $N(0, \sigma_0^2)$, even when $\tilde{P}_n^*$ is misspecified. Thus, we can show that the $\tilde{P}_n^*$-bootstrap is consistent almost everywhere for the purpose of estimating the limit distribution of $\sqrt{n}(\psi_n - \psi_0)$, but the bootstrap has the advantage of also obtaining an estimate of the finite sample sampling distribution of the estimator under this bootstrap distribution. Normally, the consistency of a model based bootstrap that samples from an estimate $\tilde{P}_n^*$ of P_0 relies on consistency of the density of $\tilde{P}_n^*$ as an estimator of the density of P_0. In this case, however, the consistency of the bootstrap only relies on the conditions under which the TMLE $\Sigma^2(\tilde{P}_n^*)$ is a consistent estimator of $\Sigma^2(P_0)$. This in turn can allow that parts of P_0 are inconsistently estimated within $\tilde{(}P)_n^*$. Therefore, we refer to this bootstrap as the *targeted bootstrap*.

The TMLE of $\Sigma^2(P_0)$ is typically represented as $\Sigma^2(Q_{\Sigma,n}^*)$ for a smaller parameter $P \rightarrow Q_\Sigma(P)$ utilizing a possible nuisance parameter estimator $g_{\Sigma,n}^*$ of a $g_\Sigma(P_0)$. As a consequence, $\tilde{P}_n^*$ can be defined as any distribution for which $Q_\Sigma(\tilde{P}_n^*) = Q_n^*$, without affecting the consistency of the targeted bootstrap. This demonstrates that the targeted bootstrap is robust to certain types of model misspecification. For the best finite sample performance in estimating the actual sample sampling distribution of $\hat{\Psi}(P_n)$, it might still be helpful that also the remaining parts of P_0, beyond Q_0, are well approximated; however that contribution will be asymptotically negligible.

Bootstrap Confidence Intervals

Once a bootstrap sampling distribution is obtained, a number of methods have been proposed to generate confidence interval endpoints from them. Hall [1988] presents a framework by which to evaluate bootstrap confidence intervals. We follow that approach here. We are interested in studying the *accuracy* and *accuracy* of various confidence interval. For a given confidence interval endpoint, $\psi_{n,[\alpha]}$, we say that it's jth order accurate if we can write $P_0(\psi_0 < \psi_{n,[\alpha]}) = \alpha + O(n^{-j/2})$. Coverage probability of a one-sided interval is closely related to its accuracy. We also discuss the coverage error of a two sided confidence interval.

The most general theoretical results for bootstrap confidence interval accuracy for the nonparametric bootstrap come from the smooth functions setting of Hall [1988]. This setting is for parameters that can be written as $f(P_0 Y)$ where Y is a vector generated from a set of transformations of O, (i.e. $h^j(O)$), and where f is a smooth function. This setting accommodates many common parameters such as means and other moments but also leaves out other common parameters like quantiles. Notably, it does not include the TSM or other kinds of Targeted Learning parameters. Below we present some bootstrap confidence interval methods and state the relevant theoretical results in this setting.

Bootstrap Wald Interval

A Wald interval using the bootstrap estimate of variance:

$$\hat{\sigma}^2_{n,\text{boot}} \quad = \quad \frac{1}{B}\sum_{i=1}^{B}\left(\Psi(P_{n,\#,i}) - \bar{\Psi}(P_{n,\#,i})\right)^2$$

with $\bar{\Psi}(P_{n,\#,i}) = \frac{1}{B}\sum_{i=1}^{B}\Psi(P_{n,\#,i})$. As before:

$$\psi_{n,[\alpha],\text{Wald}} \quad = \quad \psi_n - n^{-1/2}\hat{\sigma}_{n,boot}\phi^{-1}(1-\alpha)$$

The Wald interval method is first order accurate in the smooth functions setting [Hall, 1988].

Percentile Interval

Efron's percentile interval directly using the α quantile of $\hat{G}(x)$:

$$\psi_{n,[\alpha],\text{Percentile}} \quad = \quad \hat{G}^{-1}(\alpha)$$

The percentile interval is also first order accurate in the smooth functions setting [Hall, 1988].

bootstrap-t interval

The bootstrap-t interval can be thought of as an improvement to the Wald style interval. It relies on the following "studentized" distribution function.

$$K(x) = P\left(\frac{n^{1/2}(\hat{\psi}_n - \psi_0)}{\hat{\sigma}_n} < x\right)$$

The bootstrap estimate of this distribution is as follows:

$$\hat{K}(x) = \frac{1}{B}\sum_{i=1}^{B} I\left(\frac{n^{1/2}(\hat{\psi}_n^{\#} - \hat{\psi}_n)}{\hat{\sigma}_n} < x\right)$$

Defining $\hat{y}_\alpha = \hat{K}^{-1}(\alpha)$ as the estimate of the α quantile of this distribution, we modify the Wald interval as follows:

$$\psi_{n,[\alpha],\text{bootstrap-}t} \quad = \quad \psi_n + n^{-1/2}\hat{\sigma}_n\hat{y}_\alpha$$

A commonly cited drawback of this method is it requires a reliable estimate of σ [Hall, 1988]. However, in our setting we have access to estimates of σ both from influence curves and targeted estimates of variance. In our simulation study (below), we used the influence curve variance estimate except in the case of the targeted and joint targeted bootstraps, where we used the targeted estimate. The bootstrap-t interval is second order accurate in the smooth functions setting [Hall, 1988].

BC_a interval

The BC_a (bias-corrected, accelerated) interval, first presented in Efron [1987], accounts for bias and skew in a sampling distribution when forming a confidence interval. Its development was motivated by the practice of employing monotone transformations to normalize the sampling distribution of an estimator. It depends on two additional parameters. The bias constant z_0 captures the bias in the sampling distribution, while the acceleration constant a captures the skewness of the sampling distribution.

Given both of these quantities, the BC_a defines a new quantile to look up:

$$\begin{aligned}\beta_{z_0,a,\alpha} &= \Phi\left(z_0 + \frac{z_0 + z_\alpha}{1 - a(z_0 + z_\alpha)}\right)\\ \psi_{n,[\alpha],BC_a} &= \hat{G}^{-1}(\beta_{z_0,a,\alpha})\end{aligned}$$

Where $\Phi(x)$ is the standard normal distribution and $z_\alpha \equiv \Phi^{-1}(\alpha)$ is its α quantile. To generate this interval in practice, we require estimates of z_0 and a. We estimate z_0 as the normal quantile for the proportion of the bootstrap estimates that fall below the original sample estimate:

$$\hat{z_0} = \Phi^{-1}\left[\hat{G}(\hat{\psi_n})\right]$$

We use our knowledge of the influence function to estimate the acceleration constant a from the original sample:

$$\hat{a} = \frac{\sum_{i=1}^{n} D(O_i)^3}{6\left(\sum_{i=1}^{n} D(O_i)^2\right)^{3/2}}$$

The BC_a interval is also second order accurate in the smooth functions setting [Hall, 1988].

1.5 Simulation

To evaluate the practical performance of the Targeted Bootstrap, we simulate data from the following P_0:

$$\begin{aligned}
W_1 &\sim U(-1,1)\\
W_2 &\sim U(-1,1)\\
W^* &= W_2 - W_1\\
A|W &\sim \text{Bernoulli}\left(\text{inv.logit}(-0.5W^*)\right)\\
Y|A,W &\sim \text{Bernoulli}\left(\text{inv.logit}(A(1-0.5W^* + sin(W^*)))\right)
\end{aligned}$$

The identifiability assumptions needed to identify the treatment specific mean $E_P[Y_1]$ with $E_P E_P(Y \mid A=1, W)$ are met in this simulation. Randomization ($Y_a \perp\!\!\!\perp A|W$) can be seen to be met from the specification of P_0. Positivity ($\bar{g}(P_0)(W) = P_0(A=1|W) > 0$) is also met: $0.26 < \bar{g}(P_0)(W) < 0.74$. Samples of size $n = 1000$ were generated for each of $B = 1000$ Monte Carlo simulations.

Estimation

In our simulation, we estimated $\bar{Q}(P_0)(W) = E[Y|A=1,W]$ using kernel regression with bandwidth selected by 10-fold cross-validation (i.e. Discrete SuperLearner). Bandwidth was chosen from a grid of 30 bandwidths chosen to contain an optimal bandwidth, as well as bandwidths that would both over- and under-fit the truth for the sample sizes evaluated. We estimate $\bar{g}(P_0)(W)$ using a correctly specified logistic regression.

For each simulation iteration, we estimated Q, and fit three TMLEs: a TMLE for the treatment specific mean, a TMLE for its asymptotic variance, and a joint TMLE for both the treatment specific mean and its asymptotic variance. After fitting the TMLEs, we generated 1000 repeated bootstrap samples from five different methods: the Nonparametric Bootstrap, the Clustered Nonparametric bootstrap, model-based bootstrap based on the initial Super Learner fit, the targeted bootstrap sampling from the TMLE distribution targeting the asymptotic variance, and the joint targeted bootstrap sampling from the joint targeted TMLE distribution.

The three TMLEs fit to the simulated dataset generated different confidence interval estimates: one Wald style interval based on the influence curve from the first TMLE, and direct estimates of the variance for the remaining two TMLEs. For the five bootstrap methods, we estimated intervals for all four methods described in section 1.4. We evaluated coverage and interval lengths for all estimated confidence intervals. We also compared the performance of the Super Learner on full samples and samples from all the bootstrap approaches. To evaluate the performance of Super Learner on bootstrap samples, we compared which bandwidths were selected on the various sample types, as well as the risk difference ratios for those selections.

Results

As described above, Super Learner behaves differently on nonparametric bootstrap samples than on full samples, behaving more like a resubstitution estimator. Figure 1.1 illustrates this. While on full samples, the Super Learner (cross-validation) often selects bandwidths close to those selected by the oracle selector (minimizing the true risk), on nonparametric bootstrap samples, Super Learner most often selects the lowest available bandwidth, overfitting the data. On other kinds of bootstrap samples, including targeted bootstrap, Super Learner behaves more like it does on full samples, suggesting that these bootstrap methods don't have the same problem. This difference in the selection behavior impacts the performance of the resulting Super Learner in terms of the risk difference ratio (eq. (1.1)). This can be seen in fig. 1.2. Again, other bootstrap samples behave more like full samples in terms of the risk difference ratio.

Figures 1.3 to 1.5 show how Super Learner performance impacts confidence interval performance for the resulting TMLE estimate. In general, the over-fit Super Learner being used in TMLE on nonparametric bootstrap samples is more variable than the well-fit Super Learner being used in full samples. Therefore, nonparametric bootstrap confidence intervals are unnecessarily long and over-cover. The effect of this over-coverage on length is modest at $n = 1000$, with the Wald intervals estimated from the nonparametric bootstraps are on average just 4% longer than the standard influence-curve based confidence intervals. At smaller sample sizes, the effect is more severe: nonparametric bootstrap intervals are 21% longer than influence curve intervals at $n = 250$, and 39% longer at $n = 100$. This substantial increase in length will negatively impact the power of nonparametric bootstrap confidence intervals. In our simulation, the set of bandwidths from which Super Learner could select was fixed with respect to sample size. We expect that, if smaller bandwidths had been available, Super Learner on nonparametric bootstrap samples would have chosen them, increasing the impact of over-fitting on larger sample sizes.

These figures also show the importance of a bootstrap that jointly targets both the parameter of interest and its asymptotic variance. For interval types other than Wald, the (variance-only) targeted bootstrap intervals have very poor coverage. This is because these intervals are not centered on the TSM estimate from the full dataset, and are instead centered on the average estimate from the bootstrap intervals. In the case of targeted bootstrap samples, these estimates are biased, because the targeted bootstrap is targeting only the variance, and not the actual parameter of interest.

Figure 1.5 shows that at small sample sizes, the asymptotic Wald intervals have lower than nominal coverage, with all methods under-covering by at least 2.5%. Small sample sizes such as this are where the bootstrap has the most potential to improve upon asymptotic confidence intervals. At larger sample sizes, the second order terms become relatively unimportant. However, even at this small sample size, asymptotic intervals are only modestly anti-conservative in this simulation. Figure 1.7 might offer some explanation. Even at $n = 100$, our simulated sampling distribution is already very close to normal.

Focusing only on the Joint Targeted Bootstrap, we can compare the performance of different bootstrap confidence interval types. Figure 1.6 shows this comparison. At modest sample sizes, Bootstrap-t intervals over-cover and are longer than other interval types. The other bootstrap methods generate intervals of similar length. Of the three, BCa has the closest to nominal coverage over the range of sample sizes tested. Therefore, it is recommended that this interval type be used with the Joint Targeted Bootstrap going forward.

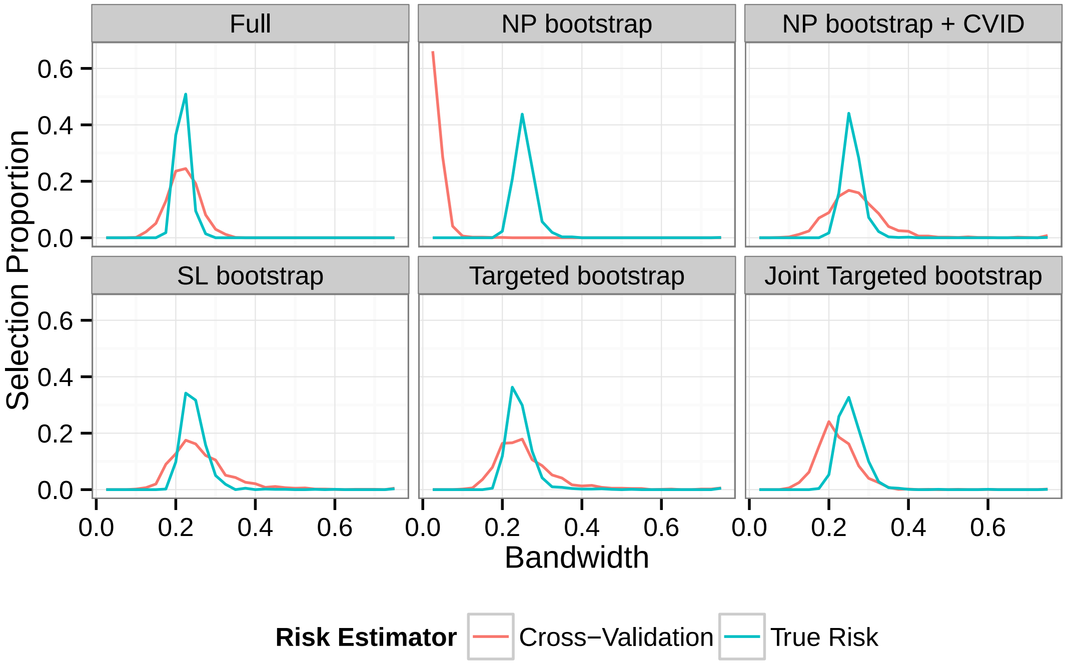


Figure 1.1: Selection Proportion of different bandwidths on different sample types

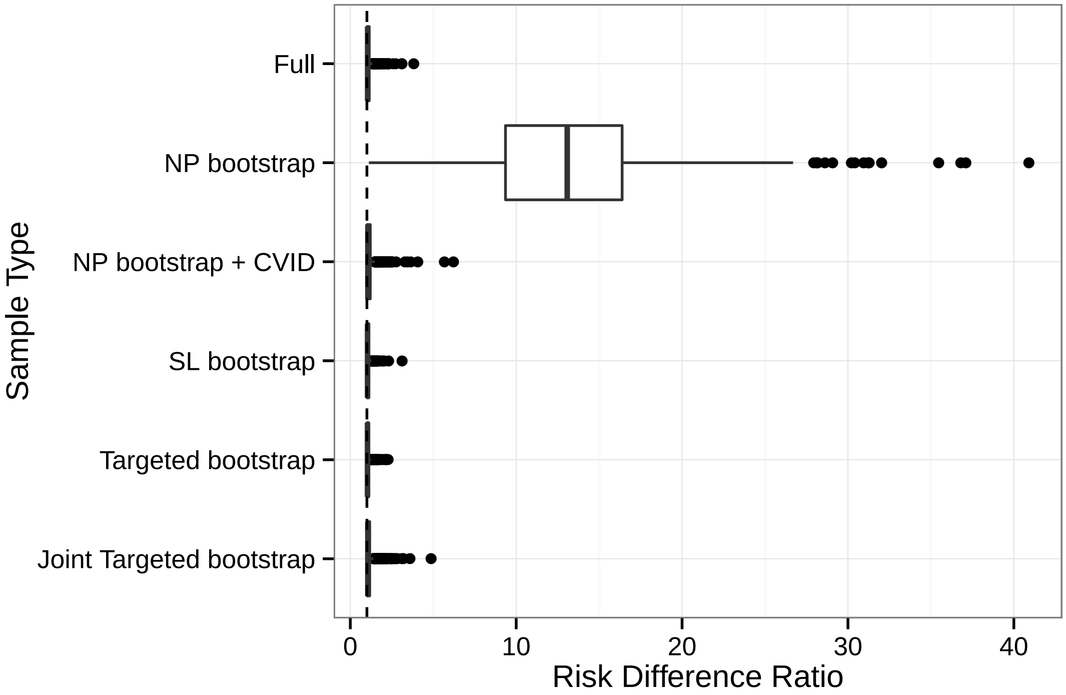


Figure 1.2: Risk Difference Ratio (defined in eq. (1.1)) of Super Learner on different sample types

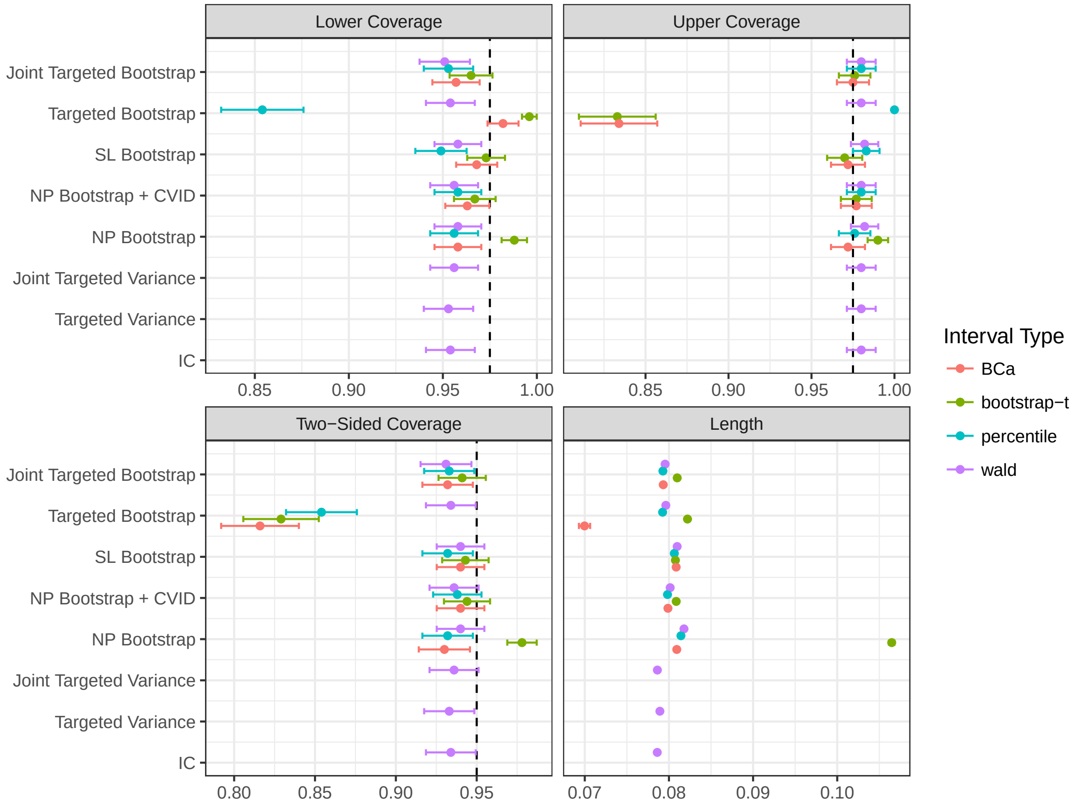


Figure 1.3: Confidence Interval Coverage and Length for n=1000

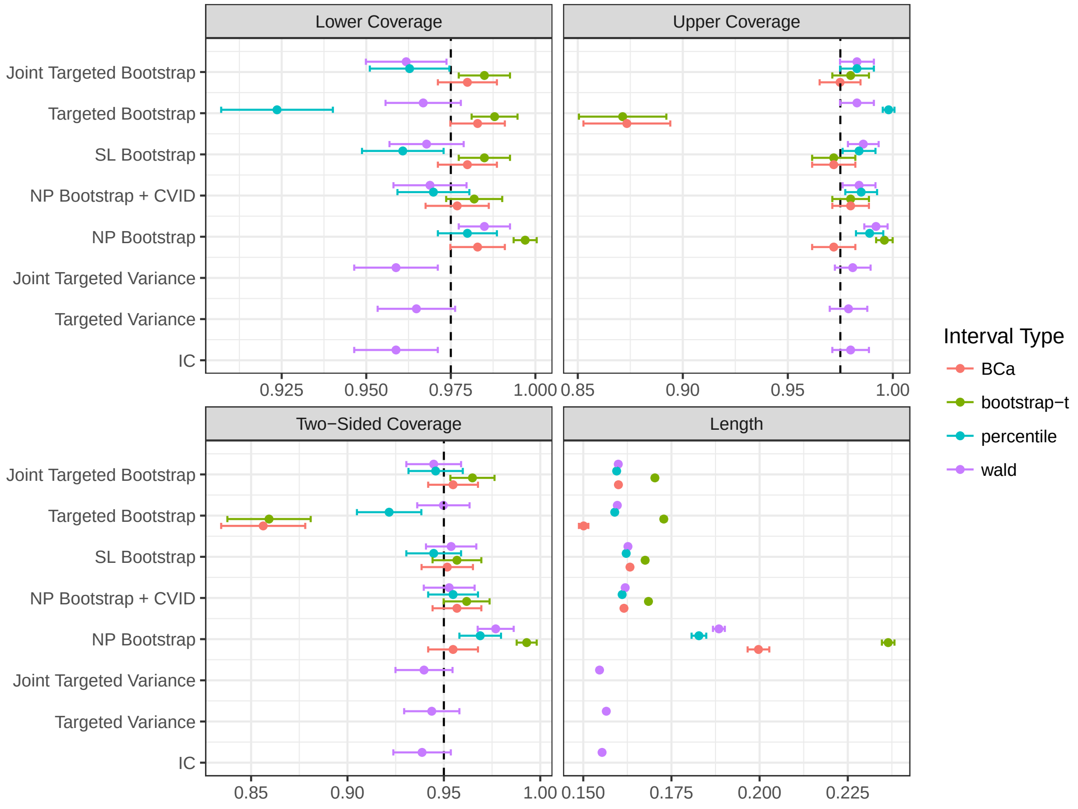


Figure 1.4: Confidence Interval Coverage and Length for n=250

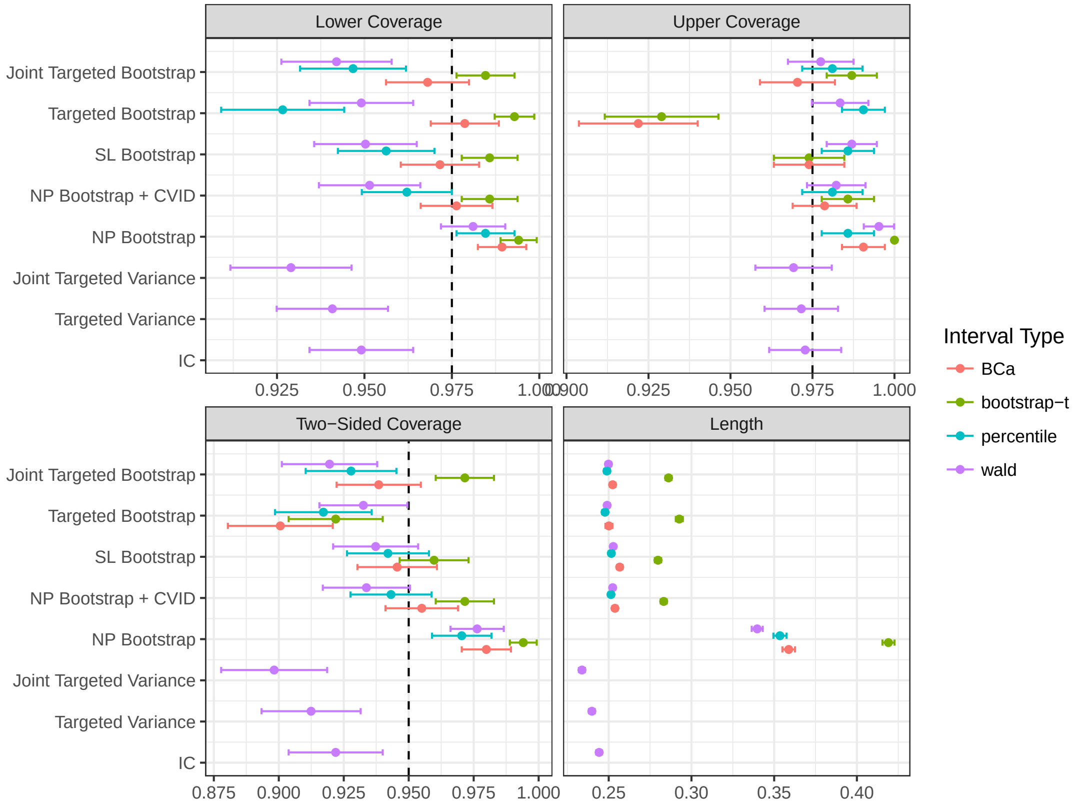


Figure 1.5: Confidence Interval Coverage and Length for n=100

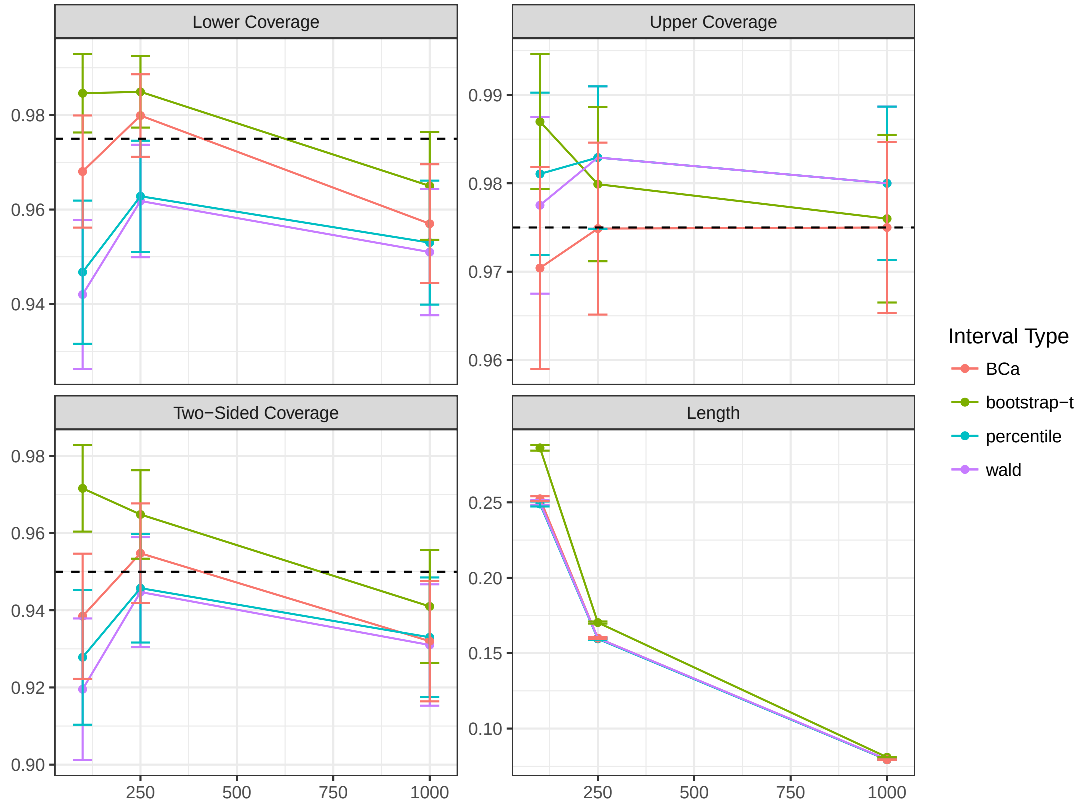


Figure 1.6: Joint Targeted Bootstrap Interval Performance Comparison

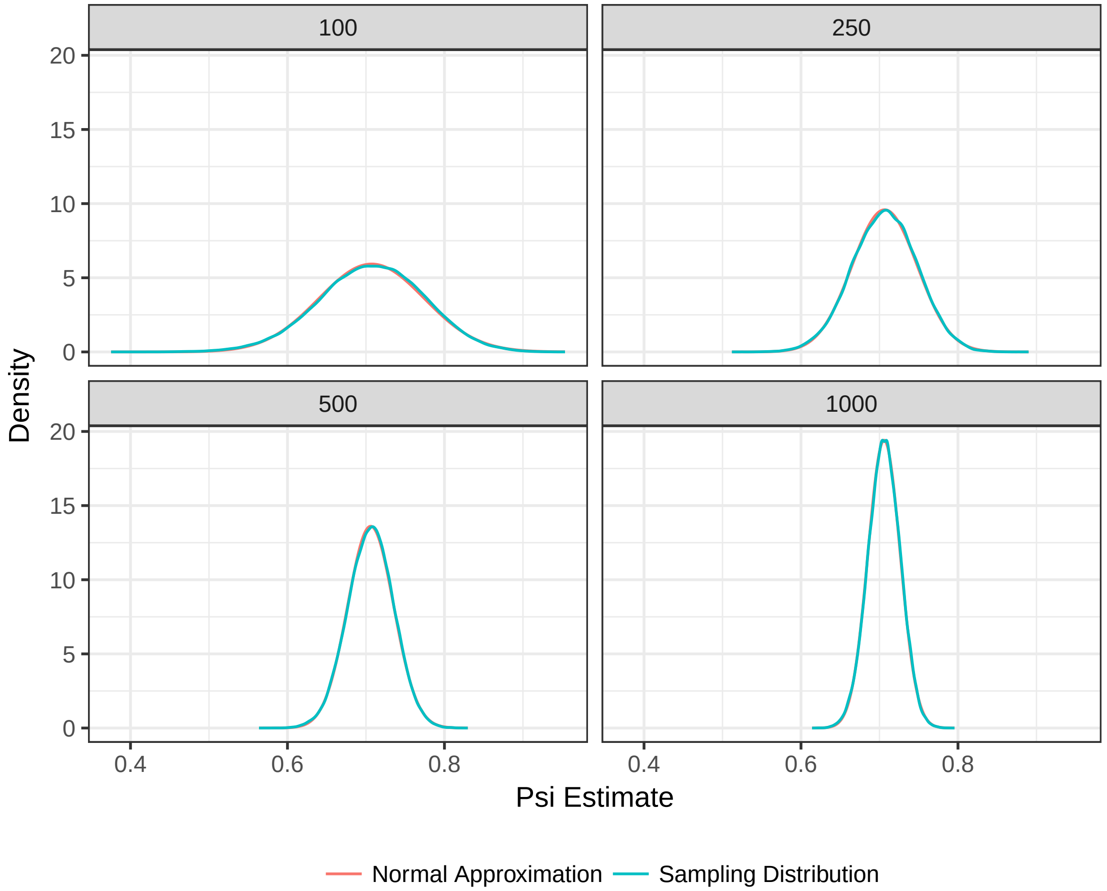


Figure 1.7: Sampling Distribution and Normal Approximation

1.6 Conclusion

We have shown the effectiveness of the targeted bootstrap for estimating properties of the sampling distribution theoretically and through simulation results. Our simulation illustrates the problems of applying the nonparametric bootstrap to a TMLE estimate with initial estimates based on Super Learner. Specifically, ties in nonparametric bootstrap samples sabotage the sample splitting that occurs in cross-validation, causing cross-validating risk estimates to behave more like resubstitution estimates. This leads Super Learner to select over-fit models on nonparametric bootstrap samples. By sampling from a continuous distribution estimate, and one that is targeted to the parameters of interest, Targeted Bootstrap does not create the ties that break cross-validation, and therefore generates confidence intervals with acceptable performance. We have demonstrated the superiority of the targeted bootstrap over the nonparametric bootstrap in the context of Targeted Learning.

Additional work is necessary to further explore the issue of bootstrap confidence intervals for Targeted Learning. A simulation study with a continuous outcome variable (Y), especially one with a skewed error distribution, would be interesting in several ways. First, it would validate the Targeted Bootstrap approach for continuous outcomes, which theory tells us should be consistent even when the error distribution is misspecified. Secondly, it would allow us to investigate the magnitude of second order terms in a setting with a sampling distribution that might be more skewed at smaller sample sizes. Another extension would be to investigate additional bootstrap confidence interval types, especially the tilted and automatic percentile interval types [DiCiccio and Romano, 1990].

Chapter 2

Computationally Efficient Cross-Validation in the Presence of Nuisance Parameters

2.1 Introduction

Targeted learning is a principled framework for the estimation of target parameters of interest from data while minimizing unjustifiable assumptions [Van Der Laan and Rose, 2011]. This framework makes frequent use of cross-validation, both for Super Learner based model selection [van der Vaart et al., 2006a, van der Laan et al., 2007b], and CV-TMLE based parameter estimation [Zheng and Van Der Laan, 2010]. The application of cross-validation in these contexts has powerful theoretical justifications: oracle results for Super Learner model selection, and the elimination of difficult-to-assess empirical process conditions from CV-TMLE step.

In many cases, estimation of the parameters of interest, or target parameters, depends on estimates of relevant nuisance parameters. Theory tells us that these nuisance parameters are best estimated with Super Learner [van der Laan et al., 2007b]. This creates a situation where both the target parameter and the nuisance parameters on which it depends are estimated using cross-validation. In this setting, it becomes important to consider how best to estimate the nuisance parameters while respecting the sample splits of the cross-validation for the target parameter. Considerations of how much cross-validation is needed and sufficient conditions for inference in similar contexts was discussed previously in Hubbard et al. [2016].

A commonly suggested approach is to fully nest the nuisance parameter cross-validation inside each training sample of the cross-validation for the target parameter [van der Laan and Dudoit, 2003b]. However, Super Learner involves the repeated application of computationally demanding machine learning algorithms or “learners” to potentially large datasets, which means that using to estimate the same nuisance parameter multiple times on different training sets comes at a large computational cost. In addition, the sample splitting required by cross-validation reduces the effective sample size to which learners can be fit and from which parameters can be estimated. This reduces the efficiency of estimation in the presence of nested cross-validation. These drawbacks suggest the need to balance the amount cross-validation applied to any estimation problem.

A more naïve approach is to fit the Super Learners sequentially. That is, first fit the Super Learner for the nuisance parameter on the full dataset. Then, using the estimates of the nuisance parameter fit on the full dataset, fit the Super Learner for the target parameter on the full dataset. Intuitively, this "violates" the cross-validation for the target parameter, as some information from a validation set is included in nuisance parameter estimates for the corresponding training set. The question becomes how much sample re-use can we "get away with" and still achieve the desirable optimality properties that we get from cross-validation.

In this paper, we investigate a hybrid "Split Sequential" approach that doesn't require nested Super Learner, but also minimizes the amount of information from validation sets used to generate nuisance parameter estimates for training sets. We present the algorithm for this approach and compare it to the Nested and Sequential approaches described above. Some theoretical results are presented, with proofs and details found in the corresponding appendix. We illustrate the relative performance of these three methods via simulation study.

The remainder of this paper is organized as follows: First, we illustrate the use of nested cross-validation in the context of optimal treatment. Section 2.2 describes the problem of nesting Super Learners formally. Section 2.3 outlines algorithmically the three approaches to estimating a target parameter that depends on a nuisance parameter when cross-validation is used to estimate both. Section 2.4 presents the theoretical results of these nesting schemes. Section 2.5 describes a simulation study in the context of optimal treatment. Section 2.6 describes an additional study of dependent Super Learners motivated by the results of the first simulation.

Motivating Example

To illustrate an application where nested cross-validation is of particular relevance, we consider the problem of learning an optimal treatment rule estimating the mean outcome under this rule. Specifically, we consider this for a single binary treatment (as opposed to multinomial or continuous treatments, or treatments at multiple timepoints). Suppose we observe n i.i.d. observations of $O = (W, A, Y) \sim P_0$, for baseline covariates W, treatment $A \in \{0, 1\}$, and outcome $Y \in \{0, 1\}$. We make no assumptions about the distribution of P_0 and so we say that $P_0 \in \mathcal{M}$ where $\mathcal{M}$ is the fully non-parametric model. We can break the data generating distribution P_0 into three parts:

$$\begin{aligned} P_0(O) &= P_0(Y|A, W)P_0(A|W)P_0(W) \\ &\equiv Q_0(Y|A, W)g_0(A|W)Q_{W,0}(W) \end{aligned}$$

where $Q_{W,0}(W)$ is the marginal distribution of the covariates, $g_0(A|W)$ is treatment mechanism, the distribution of treatments conditional on the covariates, and $Q_0(Y|A,W)$ is the distribution of the outcomes conditional on the covariates and treatment. We also define the condition mean of the outcome: $\bar{Q}_{Y,0}(A,W) \equiv E_0\left[Y|A,W\right]$. This allows us to define the notation $E_0[Y_a] \equiv E_{0,W}\left[\bar{Q}_{Y,0}(A=a,W)\right]$

In this context we wish to estimate a dynamic treatment rule, a function $d(V)$ that takes a subset of covariates $V \subseteq W$ and assigns a treatment based on them. We are also interested in the *value* of a dynamic rule, $E_0[Y_{d(V)}] = E_{0,W}\left[E_0\left[Y|A=d(V),W\right]\right]$, which, under causal assumptions, can be interpreted as the mean outcome if, possibly contrary to fact, treatment was assigned according to the rule. The optimal rule is the rule with the maximal value: $d_0 \equiv \arg\max_{d\in\mathcal{D}} E_0\left[Y_{d(V)}\right]$.

A key quantity for optimal treatment is the blip function. For a binary treatment, $A \in \{0,1\}$, we define a blip function as $\bar{Q}_0(V) \equiv E_0[Y_1 - Y_0|V] \equiv E_0[\bar{Q}_{Y,0}(1,W) - \bar{Q}_{Y,0}(0,W)|V]$. This also referred to as a *contrast*. The rule can be derived directly from this quantity: $d_0(V) = I(\bar{Q}_0(V) > 0)$.

Estimation

We will use the estimation approach outlined in Luedtke and van der Laan [2016b] and van der Laan and Luedtke [2015], which makes frequent use of cross-validation, both for Super Learner based model selection [van der Vaart et al., 2006a, van der Laan et al., 2007b], and CV-TMLE based parameter estimation [Zheng and Van Der Laan, 2010]. Luedtke and van der Laan [2016b] presents several methods for learning optimal treatment rules with Super Learner. However, for the sake of simplicity, we will focus on Super Learning of the blip function.

Figure 2.1 illustrates the amount of cross-validation necessary in this context. Our parameter of interest is the mean outcome under the optimal rule: $\Psi(P_0) = E_{P_0}Y_{d_0}$. This is indexed by the optimal rule itself $d_0(W) = \arg\max_A Q_0(A,W)$. In order to obtain an estimate $\hat{d}_n(W)$ of this rule, we first require estimates $\bar{Q}_n(A,W)$ and $\bar{g}_n(A|W)$ of the relevant factors of the likelihood – $\bar{Q}_0(A,W) \equiv E_0(Y|A,W)$ and $\bar{g}_0(A|W) \equiv P_0(A|W)$, respectively. Having estimated these parameters,, we can now estimate the mean performance under the rule $E_0[Y_{d_0}]$.

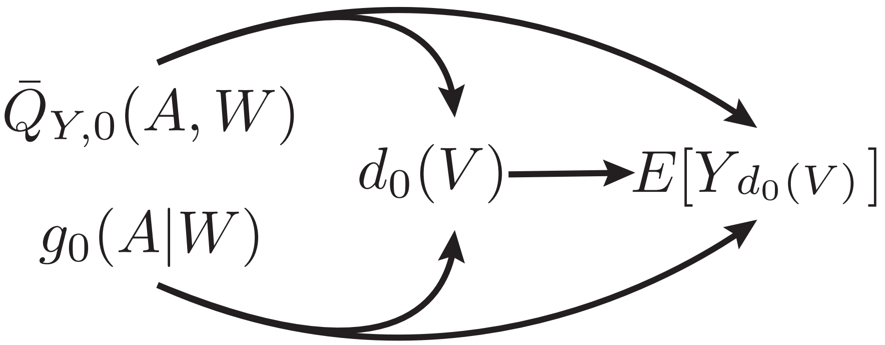


Figure 2.1: Dependency between different parameters for the optimal treatment problem

The above estimation problem involves three applications of cross-validation: first in Super Learners to estimate $\bar{Q}_0(A,W)$ and $\bar{g}_0(A|W)$, next in a Super Learner to estimate $d_0(W)$, and finally in a CV-TMLE to estimate $E_0[Y_{d_0}]$. As discussed in van der Laan and Luedtke [2015], CV-TMLE is necessary as standard TMLE is optimistic (biased upward) for the mean outcome under the rule. The most conservative possible approach would be to nest these cross-validation procedures within one another: in each CV-TMLE fold, estimate $d_0(W)$ using Super Learner applied to the training set, and in each fold of that Super Learner estimate $\bar{Q}_0(A,W)$ and $\bar{g}_0(A|W)$ using Super Learner applied to the training set of the training set. These nesting procedures are described more formally below. For 10-fold cross-validation, this procedure involves running the Super Learners for $\bar{Q}_0(A,W)$ and $\bar{g}_0(A|W)$ 100 times, a huge computational burden. In addition, the Super Learners for $\bar{Q}_0(A,W)$ and $\bar{g}_0(A|W)$ are being estimated with only $0.9^2 \approx 80\%$ of the full sample, reducing statistical efficiency for these estimates. In a longitudinal optimal treatment problem, the amount of nested cross-validation gets even more severe, with each time-point requiring an estimate of the rule at the next time-point. It's therefore important to determine if we can avoid this fully conservative approach.

2.2 Problem Statement

As illustrated by the optimal treatment example, there are some problems where estimating a high dimensional parameter or function requires first estimating or knowing some other high dimensional nuisance parameter or function. In the optimal treatment example we needed to have estimates of $\bar{Q}_0$ and g_0 to estimate the optimal rule. We now present this learning task in generality.

Suppose the nuisance parameter γ_0 is known to belong to some set Γ. For each γ in Γ, we suppose that we have a loss function L^γ that takes as input an observation O and an element of a set Θ and outputs a nonnegative real number. We wish to estimate some θ_0

known to belong to Θ, with the property that

$$\theta_0 \equiv \arg\min_{\theta \in \Theta} E_{P_0}\left[L^{\gamma_0}(O, \theta)\right].$$

In the optimal treatment example, Θ is the set of binary treatment strategies which take as input W, and θ_0 is the optimal such strategy under P_0. The set Γ is the set of pairs of a treatment mechanism (a function mapping from covariates W to the probability of treatment given W) and an outcome regression (a function mapping from (A, W) to the outcome given treatment status A and covariates W).

A naïve approach to estimate θ_0 is to first estimate γ_0 using all of the data and then minimize the empirical mean of the loss, i.e. the empirical risk, using the estimate of γ_0 as the nuisance parameter value for the loss. Empirical risk minimization approaches are well known to over-fit the data unless some sort of regularization is imposed [Hastie et al., 2003]. Rather than consider empirical risk minimization, we minimize the cross-validated rather than empirical risk. Nonetheless, one still needs to estimate the nuisance parameter γ_0 from the data. In the next section we present three sample split schemes which can be used to estimate γ_0, and subsequently to estimate θ_0.

2.3 Algorithms

We model our formal cross-validation exposition on Section 2 of van der Laan and Dudoit [2003a]. Assume a random split vector $B_n \in \{0,1\}^n$ that partitions the sample into training and validation sets. $B_{n,i} = 0$ indicates that observation O_i is in the training sample, and $B_{n,i} = 1$ indicates that observation O_i is in the validation sample. In this context, $P^0_{B_n}$ and $P^1_{B_n}$ indicate the empirical distribution of the training and validation sets respectively.

Here, we will focus on K-fold cross-validation, where there are K realizations $\{b_{n,1}, \ldots, b_{n,K}\}$ of the split vector, each one having roughly $n^1 = \lfloor n/K \rfloor$ observations in the validation sample, and all other observations in the training sample. KfoldSplitVectors describes the generation of the split vectors for K-fold cross-validation algorithmically. The K validation samples are all mutually exclusive and exhaustive. Other cross-validation schemes can also be specified in terms of split vectors [van der Laan and Dudoit, 2003a].

```
Function KfoldSplitVectors($n$,$K$)
    Input: Sample size: $n$
    Number of folds: $K$
    Output: Split vector realizations: $\{b_{n,k}\}_{k=1}^{K}$
    Define the number of observations in each validation sample: $n^1 = \lfloor n/K \rfloor$
    Define a vector that indexes the observations into each validation set:
    $V \equiv \{V_i = \lceil i/n_1 \rceil\}_{i=1}^{n}$
    Randomly permute $V$
    Then the split vectors are as follows: $\{b_{n,k} = \{I(V_i = k\}_{i=1}^{n}\}_{k=1}^{K}$
```

B_n is distributed according to P_{B_n}, which puts mass $\frac{1}{K}$ on each realization $b_{n,k}$ generated by KfoldSplitVectors.

Recursive Super Learner

We define the function RecursiveSL as a general recursive Super Learner procedure using K-fold cross-validation. This function has many similarities to the canonical Super Learner described in van der Laan et al. [2007b].

The canonical Super Learner has a number of components. First, a set of J candidate estimators $\{\hat{\theta}^j(P_n)\}_{j=1}^{J}$ that are trained on set of data (here P_n) that each of which generates predictions for arbitrary new data. Second, a metalearner or combination function $\hat{\theta}^{\hat{\alpha}}(P_n) \equiv m(\{\hat{\theta}^j(P_n)\}_{j=1}^{J}, \alpha)$ indexed by α that combines predictions J candidate estimators into a single prediction. For example, a linear combination is often used: $\hat{\theta}^{\alpha}(P_n) \equiv \sum_{j=1}^{J} \alpha_j \hat{\theta}^j(P_n)$. Finally, a loss function $L(0, \hat{\theta}^{\hat{\alpha}}(P_n))$ to evaluate that combination. For example the squared error loss $L(0, \hat{\theta}^{\hat{\alpha}}(P_n)) = \Big(Y - \hat{\theta}^{\hat{\alpha}}(P_n)(X)\Big)^2$. The loss function and the candidate estimators $\hat{\theta}^J$ are allowed to rely on an estimate of the nuisance parameter ρ_0.

Once the candidate estimators, metalearner, and loss function are specified, the task of Super Learner is to choose $\hat{\alpha}$ to minimize the risk associated with the specified loss function. This is accomplished by selecting $\hat{\alpha}$ to minimize the cross-validated risk estimate. The RecursiveSL algorithm below describes this process in more detail. Specifically, the algorithm relies on is given a set of split vector realizations $\{b_{n,k}\}_{k=1}^{K}$, generated, for example with KfoldSplitVectors. For each realization $b_{n,k}$, the J candidate estimators are fit on the training set (i.e. $\hat{\theta}_k^j = \hat{\theta}^j(P_{b_{n,k}}^0)$). These estimators are then used to estimate predicted values on the corresponding validation sets. For a given α, the predictions are combined using the specified metalearner, and the loss is averaged across predictions for all validation sets, yielding a cross-validated risk estimate.

Because we use RecursiveSL to estimate both the parameter of interest θ_0 and the nuisance parameter γ_0, we use the notation η_0 to represent the target of estimation when using RecursiveSL and ρ_0 to represent the correct value of the nuisance parameter on which the loss may rely. We then show how one can use this function to estimate the nuisance parameters. Finally, we show how one can call this function to first estimate the nuisance parameter and, given these estimates, estimate θ_0.

To illustrate this in the context of the more concrete optimal treatment example, RecursiveSL will be used to estimate three parameters: $\bar{Q}_0(A, W) \equiv E_0(Y|A, W)$, $\bar{g}_0(A|W) \equiv P_0(A|W)$, and the blip function $Q_0(V) \equiv E_0[\bar{Q}_{Y,0}(1, W) - \bar{Q}_{Y,0}(0, W)|V]$. The first two parameters do not depend on any nuisance parameters, and so in those cases $\rho_0 = \varnothing$, and η_0 is $\bar{Q}_0(A, W)$, and $\bar{g}_0(A|W)$, respectively. Both $\bar{Q}_0(A, W)$ and $\bar{g}_0(A|W)$ are nuisance parameters for $Q_0(V)$, so in this case $\rho_0 = \big(\bar{Q}_0(A, W), \bar{g}_0(A|W)\big)$, and $\eta_0 = Q_0(V)$.

```
Function RecursiveSL($\{O_i\}_{i=1}^n$,{$J$ candidate estimators for $\eta_0$},$\rho \mapsto L^\rho$,EstNuis,$\{b_{n,k}\}_{k=1}^K$)
  Input: Observations $\{O_i\}_{i=1}^n$
  {$J$ candidate estimators for $\eta_0$}
  Loss function $L^\rho$ for each $\rho$. For a correctly specified nuisance parameter $\rho_0$, satisfies
  $\eta_0 = \arg\min_\eta E_{P_0}[L^{\rho_0}(O, \eta)]$.
  Nuisance parameter estimation procedure EstNuis
  Split vector realizations $\{b_{n,k}\}_{k=1}^K$
  Output: Convex combination $\hat{\alpha}$
  Split specific SL estimates $\{\hat{\eta}_k^{\hat{\alpha}}\}$
  Full sample SL estimate $\hat{\eta}^{\hat{\alpha}}$
  Estimate nuisance parameter separately for each fold, and for the full dataset:
  $\big(\{\hat{\rho}_k\}_{k=1}^K, \hat{\rho}_n\big)$=EstNuis($\{O_i\}_{i=1}^n$,$\{b_{n,k}\}_{k=1}^K$)
  for $k = 1, \ldots, K$ do
      for $j = 1, ..., J$ do
          Fit candidate estimator $j$ on training sample from split vector $b_{n,k}$, where the
          estimator may depend on $\hat{\rho}_k$: $\hat{\eta}_k^j = \hat{\eta}^j(P^0_{b_{n,k},\hat{\rho}_k})$
  For combinations of the form $\hat{\eta}^{\hat{\alpha}} = m(\{\hat{\eta}^j\}_{j=1}^J, \alpha)$, find the $\alpha$ that minimizes the
  cross-validation risk:
  $\hat{\alpha} = \arg\min_\alpha \frac{1}{K}\sum_{k=1}^K P^1_{b_{n,k}} L^{\hat{\rho}_k}(O_i, \hat{\eta}_k^\alpha)$
  Define the split specific SL estimates: $\hat{\eta}_k^{\hat{\alpha}} = m(\{\hat{\eta}_k^j\}_{j=1}^J, \hat{\alpha})$ for $k \in 1, \ldots, K$
  Refit candidate estimators on full sample (using full sample nuisance parameter
  estimate $\hat{\rho}_n$) to obtain: $\hat{\eta}^j = \hat{\eta}^j(P_n, \hat{\rho}_n)$, $j = 1, ..., J$
  Define the full sample SL estimate: $\hat{\eta}^{\hat{\alpha}} = m(\{\hat{\eta}^j\}_{j=1}^J, \hat{\alpha})$
```

RecursiveSL differs from the canonical approach in two key ways. First, RecursiveSL allows estimates of the nuisance parameter specific to each sample split ($\{\hat{\rho}_k\}_{k=1}^K$). Different algorithms for RecursiveSL vary in how they estimate the nuisance parameter (what function is used for EstNuis). Second, once $\hat{\alpha}$ is selected, RecursiveSL generates two kinds of predictions. As in canonical Super Learner, the candidate learners are refit to the full dataset

($\hat{\theta}^j = \hat{\theta}^j(P^n)$), generating full predictions $\hat{\eta}^{\hat{\alpha}} = m(\{\hat{\eta}^j\}_{j=1}^J, \hat{\alpha})$. In addition, RecursiveSL, generates split-specific estimates $\hat{\eta}_k^{\hat{\alpha}} = m(\{\hat{\eta}_k^j\}_{j=1}^J, \hat{\alpha})$ for $k \in 1, \ldots, K$ using the candidate learners fit to the training sets. Both types of predictions have different uses, described below. Now we describe the different algorithms in terms of how EstNuis, the procedure to estimate the nuisance function, is specified.

First, NestedSL, fully nests the nuisance parameter estimation inside training sets, as described in the introduction. Separate estimates of the nuisance parameter are generated for each fold by running a full Super Learner on that fold's training set. If 10-fold cross-validation is used for both Super Learners, the number of splits, K, is 10, and the size of the training set for the outer Super Learner is $0.9n$. This means that the candidate algorithms for the inner Super Learner are run $K^2 = 100$ times and only fit on $0.9^2 n$ observations.

In contrast, SplitSequentialSL and FullSequentialSL apply Super Learner to estimate the nuisance parameter only once, using the full sample. FullSequentialSL then passes the full SL fit to RecursiveSL, and SplitSequentialSL passes the split specific SL fits to the corresponding splits of RecursiveSL. Both approaches have equivalent computational demands, with substantial savings over NestedSL because they avoid nesting and therefore repeated fitting of the Super Learner for γ_0. The key difference is in the amount of sample reuse. FullSequentialSL uses learner fits generated using the full sample with weights generated using the full sample, but SplitSequentialSL uses split-specific learner fits and so only reuses the full sample for the weights.

We use $\bar{L}^{\rho=0}$ to denote a loss function for γ_0. In the algorithm below, we don't pass arguments for the set of J candidate estimators for γ_0 or the loss function $\bar{L}$, instead assuming they are known. In practice these arguments could be passed to the function. There can be a different number of candidate estimators for γ_0 and θ_0, but we use the same J for ease of notation. The input and output is the same for all functions below so this is only written for the first function. In the algorithms below, $\hat{\beta}$ indicates the dependence of the resulting estimators on learning the α for the nuisance parameter using validation data.

```
Function NestedSL($\{O_i\}_{i=1}^n$,$\{V_k\}_{k=1}^K$)
  Input: Observations $\{O_i\}_{i=1}^n$
  Training samples $\{T_k\}_{k=1}^K$
  Split vector realizations: $\{b_{n,k}\}_{k=1}^K$
  Output: Split specific SL estimate $\{\hat{\gamma}_1^{\hat{\beta}}, \ldots, \hat{\gamma}_K^{\hat{\beta}}\}$
  for $k = 1, \ldots, K$ do
      Calculate the size of the training set: $n^0 = \sum_{i=1}^n I(b_{n,k} = 0)$
      Generate split vectors specific to this training set:
      $\{b_{n^0,k,}^{\text{nested}}\}_{k=1}^K$=KfoldSplitVectors($n^0$,k)
      Run RecursiveSL($\{O_i : b_{n,k,i} = 0\}$,$\{J$ candidate estimators for $\gamma_0\}$,$\bar{L}$,0,$\{b_{n^0,k,}^{\text{nested}}\}_{k=1}^K$)
      and save the full sample SL estimate as $\hat{\gamma}_k$
  Use nested SL estimate for all folds: $\{\hat{\gamma}_k \equiv \hat{\gamma}_k\}_{k=1}^K$
```

```
Function FullSequentialSL($\{O_i\}_{i=1}^n$,$\{T_k\}_{k=1}^K$)
  Run RecursiveSL($O_n$,$\{J$ candidate estimators for $\gamma_0\}$,$\bar{L}$,0,$\{b_{n,k}\}_{k=1}^K$) and save the full sample SL estimate $\hat{\gamma}^{\hat{\beta}}$
  Use full SL estimate for all folds: $\{\hat{\gamma}_k \equiv \hat{\gamma}^{\hat{\beta}}\}_{k=1}^K$
```

```
Function SplitSequentialSL($\{O_i\}_{i=1}^n$,$\{T_k\}_{k=1}^K$)
  Run RecursiveSL($O_n$,$\{J$ candidate estimators for $\gamma_0\}$,$\bar{L}$,0,$\{b_{n,k}\}_{k=1}^K$) and save the split specific estimates $\{\hat{\gamma}_k^{\hat{\beta}}\}_{k=1}^K$
  Use split specific SL estimates: $\{\hat{\gamma}_k \equiv \hat{\gamma}_k^{\hat{\beta}}\}_{k=1}^K$
```

To estimate the parameter of interest θ_0, one could then use the following calls:

Nested SL:
RecursiveSL($\{O_i\}_{i=1}^n$,$\{J$ candidate estimators for $\theta_0\}$,$\gamma \mapsto L^\gamma$,NestedSL,$\{b_{n,k}\}_{k=1}^K$)
Full Sequential SL:
RecursiveSL($\{O_i\}_{i=1}^n$,$\{J$ candidate estimators for $\theta_0\}$,$\gamma \mapsto L^\gamma$,FullSequentialSL,$\{b_{n,k}\}_{k=1}^K$)
Split Sequential SL:
RecursiveSL($\{O_i\}_{i=1}^n$,$\{J$ candidate estimators for $\theta_0\}$,$\gamma \mapsto L^\gamma$,SplitSequentialSL,$\{b_{n,k}\}_{k=1}^K$).

Table 2.1 summarizes types of validation data reuse in generating nuisance parameter estimates for the training samples of the cross-validation for the target parameter. The Split Sequential method is a compromise between the Full Sequential and Nested methods in that it reuses validation samples to learn the coefficients used to combine the learner, but not to fit the learners themselves.

	Learner Fits	Coefficients
	$\hat{\psi}^j$	α_n
Full Sequential	yes	yes
Split Sequential	no	yes
Nested	no	no

Table 2.1: Overview of validation data reuse for various cross-validation schemes. A yes indicates that, for a given algorithm, the validation set was used to estimate the parameter indicated by the column header

General TMLE

Similar to the alternatives for recursively applying Super Learner, there are alternatives for how to apply TMLE when initial estimates are based on Super Learner. Function GeneralTMLE defines a general TMLE procedure that allows the initial estimates to be split specific, as with RecursiveSL. By varying how the initial estimates are constructed, we generate three different TMLE procedures that are directly analogous to the three recursive Super Learner procedures described above. By generating the initial estimates using FullSequentialSL, we implement the canonical TMLE. If we instead generate the initial estimates using NestedSL, we implement the canonical CV-TMLE. If we generate the initial estimates using SplitSequentialSL, we implement a novel “Split Sequential” CV-TMLE, with properties similar to CV-TMLE, but with a dramatically reduced computation cost. In this case, the initial estimates are the split specific Super Learners, so that the CV-TMLE validation sets are only used when finding the combination indexed by α_{O_n}.

```
Function GeneralTMLE($\{O_i\}_{i=1}^n$,$\{b_{n,k}\}_{k=1}^K$,$\hat{Q}_\epsilon^k$,EstNuis)
  Input: Observations $\{O_i\}_{i=1}^n$
  Split vector realizations $\{b_{n,k}\}_{k=1}^K$
  Submodel $\hat{Q}^k(\epsilon)$
  Likelihood estimation procedure EstNuis
  Output: Final Estimate $\hat{\psi}_n^*$
  Influence Curve $IC(\hat{\psi}_{P_n^*}, O_i)$
  Estimate relevant likelihood factors:
  $[\hat{Q}_1, \ldots, \hat{Q}_K]$=EstNuis($O_1, \ldots, O_n\}$,$\{\hat{Q}_1, \ldots, \hat{Q}_J\}$,$\{b_{n,1}, \ldots, b_{n,K}\}$)
  for $i \in 1, ..., n$ do
      Generate initial estimates: $\hat{Q}_i^0 = \hat{Q}_k(O_i)$ where $k : b_{n,k,i} = 1$
  $k = 0$
  do
      Find $\epsilon$ that minimizes the cross-validated risk: $\epsilon_n^{j+1} = \arg\min_\epsilon E_{B_n} P_{n,B_n}^1 L(\hat{Q}^j(\epsilon))$
      Update estimate: $\hat{Q}^{j+1} = \hat{Q}^j(\epsilon^{j+1})$
      $j = j + 1$
  while $|\epsilon| > 0$ or some small threshold
  Final estimate $\hat{\Psi}_n^* = \hat{\Psi}_n(\hat{Q}^*)$
```

2.4 Theoretical Properties

We now establish theoretical properties for the split specific methods, which show that we can expect performance similar to that of the nested methods. Proofs for these results are in appendix B.

Super-Learner

SplitSequentialSL satisfies an oracle inequality of the type presented previously in the literature [van der Laan and Dudoit, 2003a, van der Laan et al., 2006, van der Vaart et al., 2006a, van der Laan et al., 2007b]. Throughout this section we assume that the size of the smallest training sample increases at the same rate as n. This is true, for example, in K-fold cross-validation where K does not increase with sample size. In the appendix we give more general results which allow the size of the training samples to increase at a slower rate than n.

Oracle using estimated nuisance parameter

Our first two results concern the case where the the true value of the nuisance parameter is unknown, but is instead estimated from data. This case is interesting in the optimal treatment example, where one wishes to select the optimal first time point rule given that, as will happen in practice, one has the estimated second time point rule in hand. Suppose L^{γ_0} satisfies the following conditions:

$$\sup_{\theta,\gamma} \sup_{o\in\mathcal{O}} |L^{\gamma}(o,\theta)| < \infty \tag{2.1}$$

$$\sup_{\theta,\gamma} \frac{Var_P\left(L^{\gamma}(O,\theta)\right)}{E_P\left[L^{\gamma}(O,\theta)\right]} < \infty. \tag{2.2}$$

The first condition above holds if the loss is bounded uniformly. The second is a quadratic loss property that is satisfied by many losses of interest, e.g. squared error loss [van der Laan and Dudoit, 2003a] and the negative log loss [Corollary 5.4 in van der Laan et al., 2006].

We have the following theorem.

Theorem 1. *Suppose $\{L^{\gamma} : \gamma \in \Gamma\}$ satisfies (B.2) and (B.3). Fix $\delta > 0$. Then we have the following finite sample oracle ineqality:*

$$E_{P^n} R_n^{\hat{\beta}}(\hat{\alpha}) \leq (1+2\delta) E_{P^n} \min_{\alpha} R_n^{\hat{\beta}}(\alpha) + \frac{CK \log\left(1 + N_{\alpha}(n) + N_{\beta}(n)\right)}{n},$$

where P^n represents the distribution of an i.i.d. sample of size n from P and C is a constant which may rely on P, the loss L, and δ.

The above theorem shows that, when the randomness in the candidate learners is averaged across samples of size n, our method performs similarly (with respect to risk indexed by the estimated nuisance parameter) to the oracle using the estimated nuisance parameter. In practice we might want a probabilistic guarantee about the risk in our sample of size n, rather than a statement about the average across all possible samples of size n. The following result provides such a guarantee.

Theorem 2. *Suppose (B.2) and (B.3) hold. Fix $\delta > 0$ and $k > 0$. Then, for a constant C which may rely on P, the loss L, and δ, the following holds with probability at least $1 - 2/(n-1)$:*

$$R_n^{\hat{\beta}}(\hat{\alpha}) \leq (1+2\delta) \min_{\alpha} R_n^{\hat{\beta}}(\alpha) + \frac{CK \log(1 + N_\alpha + N_\beta) \log n}{n}.$$

If $N_\alpha + N_\beta = O(n^d)$ for $d > 0$ and the number of folds K does not increase with sample size, then the above implies that, with probability at least $1 - 2/(n-1)$, the left-hand side in the theorem above is of the order $O(d \log^2 n/n)$.

Oracle using known nuisance parameter

In many applications one may only care about the oracle using the known the true value of the nuisance parameter, rather than an estimate thereof. In the optimal treatment example, this corresponds to wanting to know the truly optimal first time point rule rather than the optimal first time point rule subject to the implementation of a suboptimal (but known) second time point rule.

The upcoming theorem uses $\hat{\alpha}^* \equiv \arg\min_\alpha R_n(\alpha)$ to denote the oracle choice of α, where the oracle uses the known true value of the nuisance function. Note that $\hat{\alpha}^*$ relies on the sample through the candidate learners.

Theorem 3. *Suppose the conditions of Theorem 2. Further suppose that*

$$E_{P^n}\left[\left(R_n^{\hat{\beta}}(\hat{\alpha}^*) - R_n(\hat{\alpha}^*)\right) - \left(R_n^{\hat{\beta}}(\hat{\alpha}) - R_n(\hat{\alpha})\right)\right] = o\left(E_{P^n}\left[R_n^{\hat{\beta}}(\hat{\alpha}^*) - R_n(\hat{\alpha}^*)\right]\right).$$

Then, for any $\delta > 0$,

$$E_{P^n} R_n(\hat{\alpha}) \leq (1+2\delta) E_{P^n} R_n(\hat{\alpha}^*) + O\left(E_{P^n}\left[R_n^{\hat{\beta}}(\hat{\alpha}^*) - R_n(\hat{\alpha}^*)\right] + \frac{K \log(1 + N_\alpha + N_\beta)}{n}\right).$$

The big-O expression above describes the behavior of the term when $\delta > 0$ is treated as fixed and $n \to \infty$.

In the appendix we give an inequality which bounds the expected risk if δ converges to zero as sample size grows.

In the simple case where the nuisance function is one-dimensional and the risk function is sufficiently differentiable in α and β, the condition in the above theorem is automatic by a second-order Taylor expansion of $(\alpha, \beta) \mapsto R_n^\beta(\alpha)$. More generally, such a statement will be reasonable provided the functional R_n which takes as input the nuisance function and a convex combination α is sufficiently differentiable. We will show that it is reasonable in our example.

CV-TMLE

Suppose one wishes to estimate the value of a univariate parameter Ψ and a distribution P_0. In this section we assume that one has available parameter mapping D^* such that, for any distribution P,

$$\Psi(P) - \Psi(P_0) = -E_{P_0}[D^*(P)(O)] + Rem(P, P_0),$$

where $Rem(P, P_0)$ is a remainder term that is small when P is close to P_0 in the right sense, typically in the sense that certain conditional probabilities or regression functions under P approximate the true values under P_0. Note that D^* takes as input a distribution P and outputs a real-valued function of O. We further assume that such a D^* exists such that $P_0 D^*(P_0) = 0$. In the appendix we explain how such a D^* can be found when Ψ is pathwise differentiable [Pfanzagl, 1990]. To avoid extra regularity conditions we make the additional assumption that $D^*(P)(O)$ is almost surely bounded for all P in the appendix.

When one implements a cross-validated TMLE, one chooses a loss $L(O_i, P^j_{n,\epsilon})$, such that minimizing the cross-validated risk yields the following result:

$$E_{B_n} P^1_{B_n} D^*(\hat{P}_{\hat{\beta}}(P^0_{B_n})) = \underbrace{o_P(n^{-1/2})}_{\text{remainder}}.$$

For simplicity consider the case where the above remainder is exactly zero. We then have that

$$\begin{aligned}\psi_n^{cvtmle} - \Psi(P_0) = &(P_n - P_0)D^*(P_0)\\ &+ \underbrace{P_{B_n}(P^1_{B_n} - P_0)\left[D^*(\hat{P}_{\hat{\beta}}(P^0_{B_n})) - D^*(P_0)\right]}_{(\star)} + P_{B_n} Rem(\hat{P}_{\hat{\beta}}(P^0_{B_n}), P_0).\end{aligned}$$

The equality uses the mutual exclusivity and exhaustiveness of the training samples. Our goal is to multiply both sides above by $\sqrt{n}$ and show that the first term on the right above dominates. We can then apply the central limit theorem to the leading term and develop Wald-type confidence intervals for $\Psi(P_0)$.

The P_{B_n} expectation of the remainder terms $Rem(\hat{P}_{\hat{\beta}}(P^0_{B_n}), P_0)$ will converge to zero in probability faster than $1/\sqrt{n}$ provided we can estimate the needed components of P_0 "well enough". The term $(\star)$ is the focus of the analysis we give in the appendix. If one had used a TMLE rather than a CV-TMLE, then typically one would need to ensure that the estimator of P_0 satisfies some entropy conditions, i.e. is a Donsker class (One way to show that a function class is Donsker is to consider how well the class can be approximated by a much smaller class using the idea of covering numbers and metric entropy).

It is thus not surprising that, under an entropy condition on the class of functions $\beta \mapsto \hat{P}_\beta(P^0_{B_n})$, where the training sample $P^0_{B_n}$ is treated as fixed, we are able to establish control over the first term on the final line above. We leave formal presentation of this condition to the appendix to avoid formal definitions and discussion of the uniform entropy condition in the main text. Nonetheless, this entropy condition will hold in any reasonable parametric family of interest.

Controlling the size of the class $\hat{\beta} \mapsto \hat{P}_{\hat{\beta}}(P^0_{B_n}))$, conditional on the training sample $P^0_{B_n}$, is not enough to ensure that $(\star)$ converges to zero faster than $1/\sqrt{n}$. We also need that $\hat{P}_{\hat{\beta}}(P^0_{B_n}))$ approximates P_0. In particular, we require that the mean-squared error of the estimate $D^*(\hat{P}_{\hat{\beta}}(P^0_{B_n}))$ of $D^*(P_0)$ converges to zero in probability at some rate. The rate plays little role in our result provided the class $\beta \mapsto \hat{P}_{\beta}(P^0_{B_n})$ does not grow too much with sample size – essentially we just need that this mean squared error converges to zero eventually.

2.5 Optimal Treatment Simulation

To compare the practical performance of the three dependent cross-validation procedures described above, we used the Optimal Treatment example described in section 2.1. For each of 1000 simulation iterations we sampled 1000 observations from $P(O)$:

$$\begin{aligned}
O &= (W, A, Y) \\
P(O) &= P(Y = 1|A, W)P(A = a_i|W)f(W) \\
f(W) &= \mathcal{N}(\mathbf{0}_3, \boldsymbol{I}_{3,3}) \\
P(A = 1|W) &= \text{logit}^{-1}(0.8 * W_1) \\
P(Y = 1|A, W) &= 0.5\,\text{logit}^{-1}\left[-5I(A = 1)(W_1 - 0.5)\right. \\
&\qquad\qquad\qquad \left.+5I(A = 0)(W_1 - 0.5)\right] \\
&+ 0.5\,\text{logit}^{-1}\left(W_2 W_3\right)
\end{aligned}$$

In this simulation, the value of the true optimal rule, $E_0[Y_{d_0(V)}]$ was approximately 0.563.

Estimation

The first task was to obtain an estimate of the optimal rule, $d_0(V)$. We did this by estimating the blip function $\bar{Q}_0(V) \equiv E_0[Y_1 - Y_0|V]$ with Super Learner. This identifies the rule: $d_0(V) = I(\bar{Q}_0(V) > 0)$. As described above, this requires estimates of both the treatment mechanism, $g_0(A|W)$, and the outcome distribution, $Q_0(Y|A, W)$, which were also estimated using Super Learners. We accomplished this using all three dependent super learners described above: FullSequentialSL, NestedSL, and SplitSequentialSL. In all cases, once the Super Learner combination indexed by $\alpha(O_n)$ was selected for the blip function, the individual learners comprising the Super Learner for the blip function were refit to the full dataset, using estimates of $Q_0(Y|A, W)$ and $g_0(A|W)$ generated from Super Learners also with learners refit to the full dataset.

With the rule estimated using SplitSequentialSL, TMLE was used to estimate the value of this rule: $E_0[Y_{d(V)}] = E_{0,W}\left[E_0\left[Y|A=d(V),W\right]\right]$. We did this using all three TMLE approaches described above: canonical TMLE (FullSequential), CV-TMLE (Nested), and SplitSequential CV-TMLE. Error-bars in the below graphs are Wald-style 95% confidence intervals, representing uncertainty in the Monte Carlo simulation

Results

Figure 2.2 shows that, unsurprisingly, the nested approach takes much longer to compute than the other two approaches, which both take roughly the same time. What was unexpected, however, is that all three approaches did quite similarly in terms of the performance of the estimated rule, as seen in fig. 2.3, with all three approaches generating rules that come close to the true optimal rule in terms of their values (indicated with a dashed line). It's possible that this estimation problem is simple enough that careful choice of learner combination for the blip function is not essential.

Figure 2.4 shows that the results for the TMLE are more varied. Luedtke and van der Laan [2014] showed that canonical TMLE is biased for the value of an optimal rule, a result that we have reproduced here. Both the Nested and SplitSequential CV-TMLE approaches have substantially less bias, although SplitSequential has slightly more than Nested CV-TMLE. However, SplitSequential CV-TMLE has a smaller variance, and therefore a smaller MSE, likely due to the increased sample sizes available for learning the rule. Figure 2.5 shows that these results translate predictably for inference about the value of the estimated rule. Canonical TMLE coverage is very low due to the substantial bias. In contrast, SplitSpecific CV-TMLE coverage is only modestly impacted relative to the coverage of Nested CV-TMLE.

To summarize, this simulation clearly shows that in the context of optimal treatment, SplitSequential CV-TMLE performs similar to Nested CV-TMLE in terms of learning the value of the estimated rule, and that both methods dramatically outperform canonical TMLE in this context. This means that, by using SplitSequential CV-TMLE we can achieve substantial computational savings with only a minor impact on statistical performance. However, the surprising result that nesting or lack thereof does not impact the performance of rule estimation warrants further investigation. Section 2.6 will investigate the performance of these three Super Learner approaches in a different context.

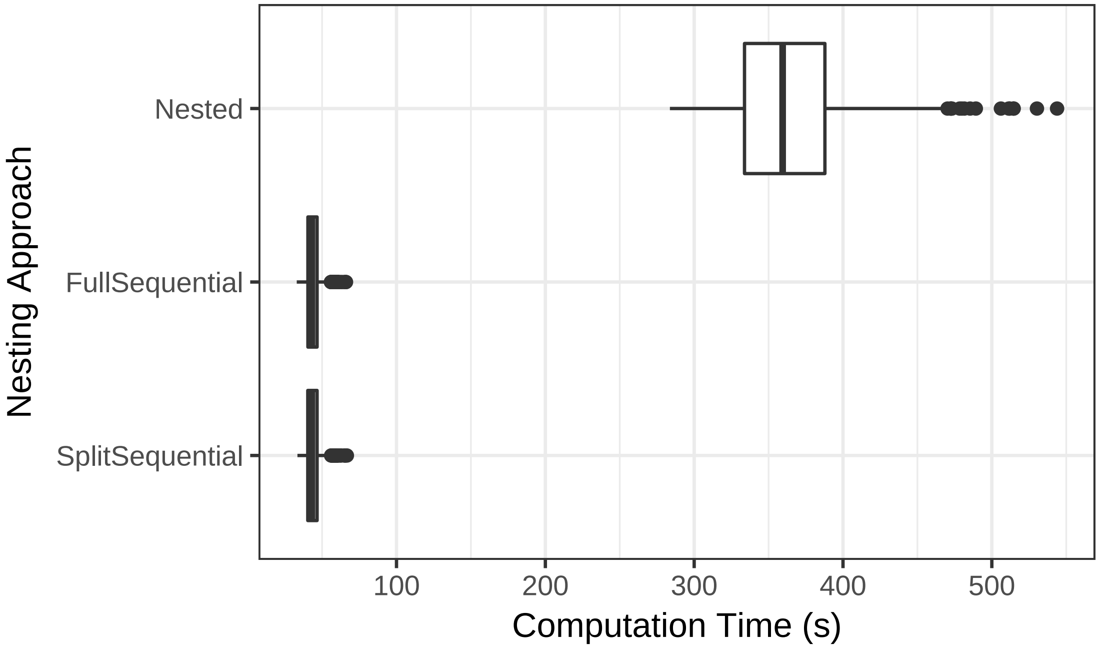


Figure 2.2: Computation Time for the Estimation of the Optimal Rule

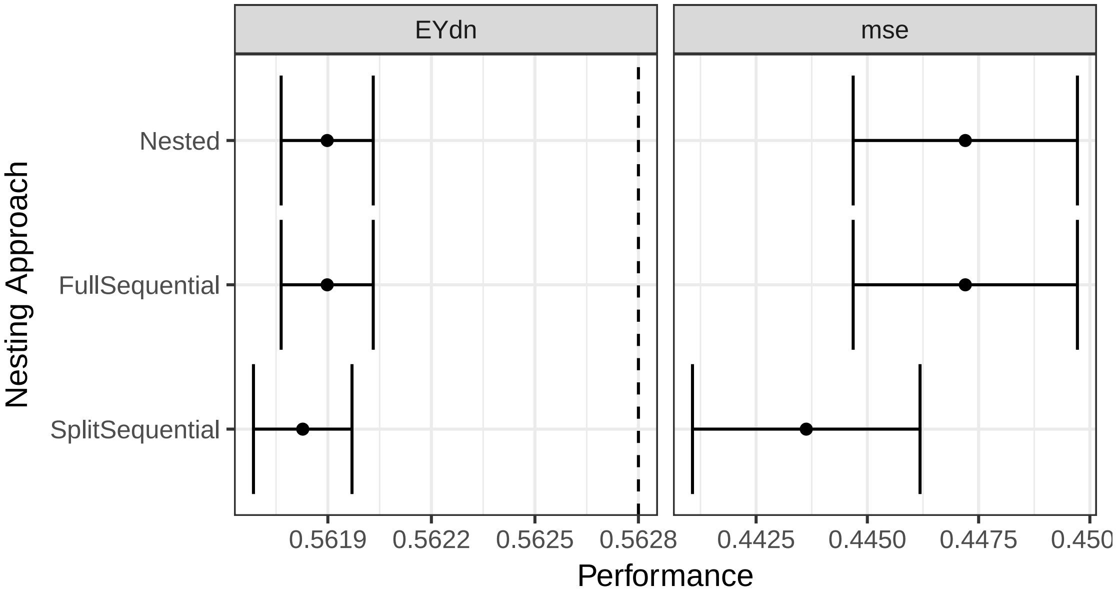


Figure 2.3: Performance of the Estimated Optimal Rule

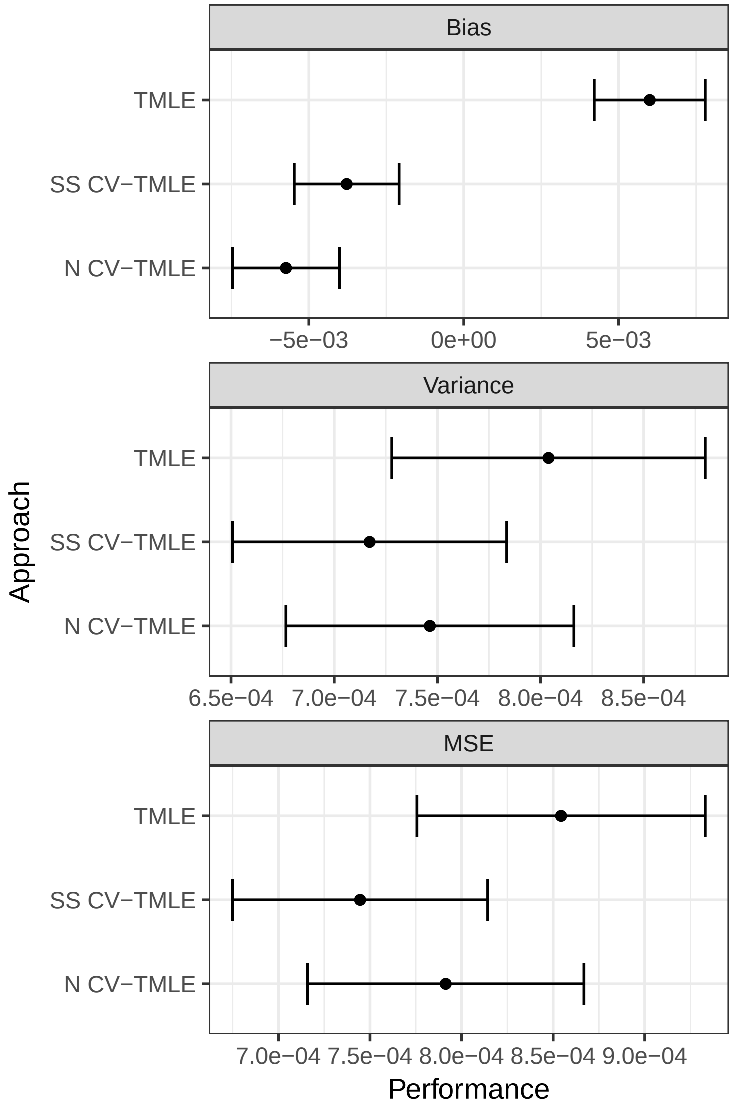


Figure 2.4: Performance of the TMLE for the Value of the Estimated Rule

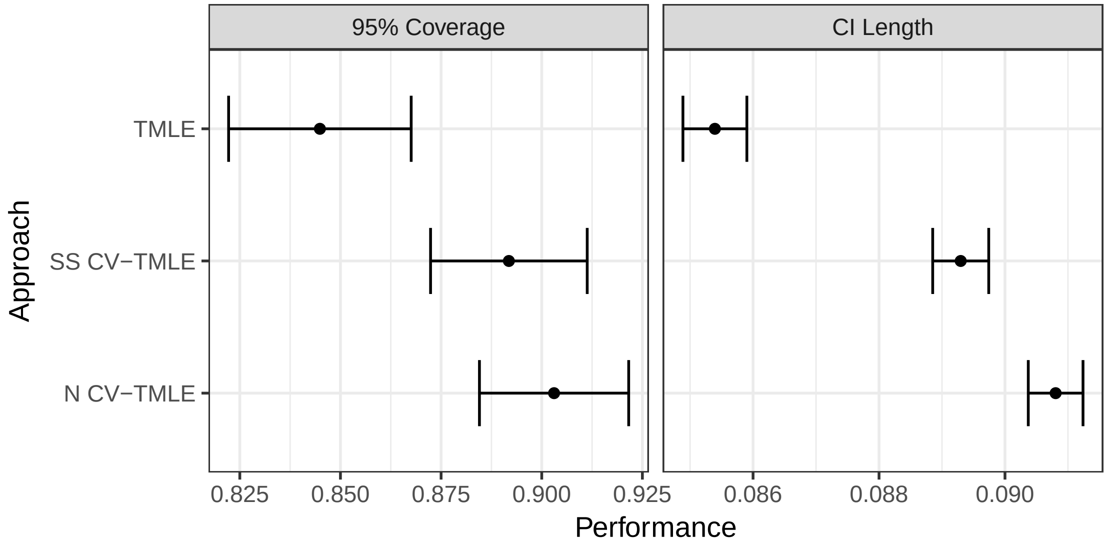


Figure 2.5: Performance of the TMLE Inference for the Value of the Estimated Rule

2.6 Conditional Variance Simulation

Nesting estimates of nuisance parameters inside training sets for the parameters of interest is important for the oracle results justifying the use of Super Learner. Therefore, it was surprising that nesting of Super Learners did not seem to have a practical effect in the optimal treatment example. Intuitively, nesting of Super Learners for nuisance parameters is most important where using full estimates is likely to behave differently than using nested estimates that are more supported by theory. To that end, we investigate the performance of the three Super Learner approaches in the context of estimating a conditional variance. We sought to obtain an estimate of $V[Y|W] = E[(Y - E[Y|W]|W])^2]$ using Super Learner, which depends on an estimate of the nuisance parameter $E[Y|W]$. In this context, we expect that estimating $V[Y|W]$ using an estimate of $E[Y|W]$ where the same dataset was used for both estimates (e.g. FullSequentialSL) will result in a biased estimate of $V[Y|W]$. For each of 1000 simulation iterations we sampled 1000 observations from $P(O)$:

$$\begin{aligned} O &= (W, A, Y) \\ f(O) &= f(Y|W)f(W) \\ f(W) &= \frac{1}{6}I(W \in (-3, 3)) \\ f(Y|W) &= \mathcal{N}(W^2, \text{logit}^{-1}(-W^2)) \end{aligned}$$

Estimation

Both $E[Y|W]$ and $V[Y|W]$, were estimated using kernel regression with bandwidth selected by 10-fold cross-validation (i.e. Discrete SuperLearner). Bandwidth was selected from a range of possible bandwidths by using numerical optimization. We estimated $V[Y|W]$ using all three Super Learner approaches described above. We also investigated the question of whether nuisance parameters should be refit to the full dataset, before the target parameter is fit to the full dataset, as is typically done in Super Learner and as was done for the above optimal treatment simulation. As an alternative, split specific validation set predictions were used for all nuisance parameter estimates. This is analogous to how initial estimates are generated in SplitSequential CV-TMLE. We expected that this would be especially important in this context, because refitting $E[Y|W]$ to the full dataset before fitting $V[Y|W]$ to the full dataset, should result in a biased estimate of $V[Y|W]$, regardless of how the bandwidth was selected.

Results

Figure 2.6 shows the results of this simulation. The results are presented in terms of a risk difference, between estimates of $V[Y|W]$ using the Super Learner selected bandwidth, and estimates of $V[Y|W]$ using the oracle bandwidth (for the oracle using the known true value of $E[Y|W]$. Ideally, these values should approach zero as the Super Learner performs comparably to the oracle. As expected, refitting the kernel smoothing model for $E[Y|W]$ to the full dataset had worse performance than using predictions on validation sets. Focusing on those results where validation set predictions were used, the SplitSequential and Nested approaches perform similarly, and both seem to perform slightly better than the FullSequential approach, although more simulation iterations are necessary to have a conclusive result. However, even in this simulation designed to cause problems for the FullSequential method, the FullSequential method is performing better than expected. Further study is needed to evaluate whether SplitSequential Super Learner is really necessary, or if FullSequential will work for most practical cases.

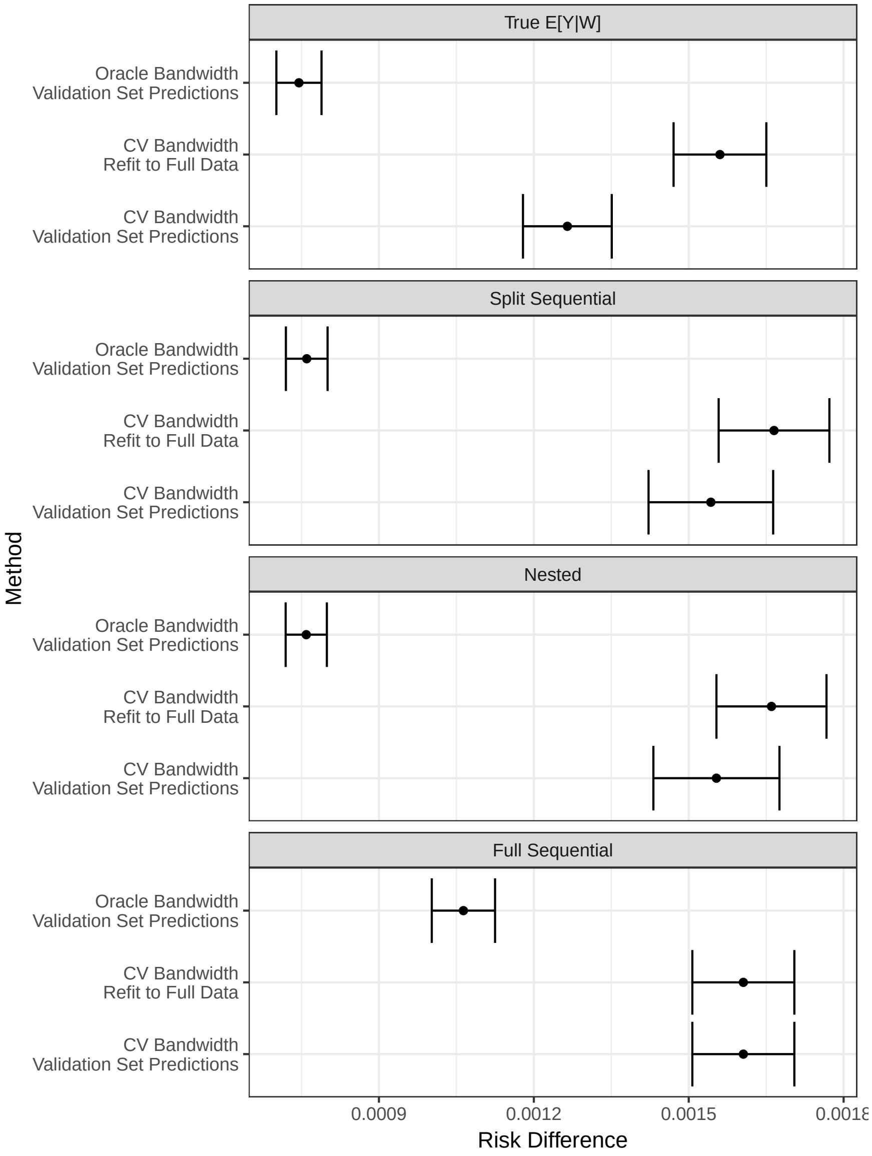


Figure 2.6: Performance of SuperLearner for $V[Y|W]$ under a range of methods for estimating $E[Y|W]$ and selecting the bandwidth

2.7 Conclusion

Cross-validation is a powerful but computationally demanding tool in the Targeted Learning toolbox. SplitSequential cross-validation for both Super Learner and TMLE allows the use of cross-validation for both nuisance and target parameters without the need for the computationally expensive and statistically inefficient method of nested cross-validation. Through theory and simulation we have shown that this approach has statistical performance that's only slightly worse than nested cross-validation, and theory suggests that this difference is asymptotically negligible.

While the simulation results for CV-TMLE are clear, the story for Super Learner is somewhat more muddled. Further work is necessary to determine the situations in which FullSequential Super Learner fails. However, as SplitSequential Super Learner is no more computationally demanding than FullSequential Super Learner, but is better supported by the theory established here, we recommend that it be the default approach in all cases of Super Learner with a nuisance parameter. We therefore plan to implement a general version of SplitSequential Super Learner in software.

Chapter 3

`opttx`, an R package for Optimal Dynamic Rules

3.1 Introduction

Precision medicine, the idea that treatment decisions should be tailored to the individual characteristics of patients, is a rapidly growing field [Collins and Varmus, 2015]. One component of precision medicine is the task of learning an optimal dynamic rule, a decision rule that assigns treatment as a function of a patient's covariates to optimize some outcome. Much work has been done to develop methodologically rigorous approaches to this task [Chakraborty and Moodie, 2013]. In order to make these methods accessible to researchers, we have developed the `opttx` R software package. Specifically, we focus on the Targeted Learning approach to optimal dynamic treatment [Luedtke and van der Laan, 2016a,b, van der Laan and Luedtke, 2015, 2014]. The development of this package was motivated by a research question from the Pragmatic, Randomized Optimal Platelet and Plasma Ratios (PROPPR) trial [Holcomb et al., 2015], a study of blood transfusion in victims of traumatic injury. This research question motivated several methodological additions to the optimal treatment framework, including support for categorical treatment and variable importance measures. Although the optimal treatment methodology and the PROPPR dataset are longitudinal in nature, we focus on a cross-sectional optimal treatment setting for simplicity. Currently, the `opttx` package only supports cross-section optimal treatment. We plan to extend the software to the longitudinal case in the near future.

The remainder of this paper is organized as follows: Section 3.2 describes the PROPPR study and describes how the research question fits into the optimal treatment framework. Section 3.3 formally describes the general statistical estimation problem of learning an optimal rule and estimating the mean outcome under that rule. Section 3.4 details the extension of optimal treatment methodology to categorical treatments. Section 3.5 describes a new Super Learner implementation created to support categorical optimal treatment. Section 3.6 compares the three proposed "pseudo-blip" approaches to categorical optimal treatment via simulation. Section 3.7 describes a variable importance measure for optimal treatment covariates. Finally, Section 3.8 presents the results of the application of the software to the PROPPR example. A separate vignette describing the basic usage of the `opttx` package can be found in appendix C.

3.2 Motivating Example

As an example to motivate discussion of the optimal treatment problem and extensions to it, we will discuss the application of optimal treatment rule estimation to data from the PROPPR trial. This trial sought to compare the effectiveness of different ratios of blood products for recovery of victims of traumatic injury. Blood for transfusion is rarely available as “whole blood”. Instead, it is available as separated blood products: plasma, platelets, and red blood cells (RBCs). Researchers were interested in which of two randomized ratios of plasma:platelets:RBCs – 1:1:1 or 1:1:2 would be more effective. Clinical intuition suggests that a one-size-fits-all treatment approach is inappropriate given the wide variety of demographics and presentations of trauma victims. Therefore, an optimal treatment strategy that assigns blood product ratios based on a patient’s baseline status and characteristics has the potential to increase survival over a uniform treatment strategy. This research question motivated development of the `opttx` software, and informed some of the features that were developed, including support for categorical treatment variables, and variable importance measures.

The PROPPR study had a large number of covariates (W), including vital signs, lab results, and demographic information. We included 49 covariates in total. The assigned treatments were implemented by restricting the order in which doctors could administer the various blood products. However, doctors were free to administer as much or as little blood as was deemed appropriate, which meant that the ratio of blood actually administered could be different than that determined by a patient’s randomization group. Comparing the ratios of blood products actually received by patients to the treatment group assignment, it is clear that patients often received ratios substantially different than those assigned by randomization. For the purposes of this analysis, we chose to focus on the ratio of platelets to RBCs actually administered to patients and defined three treatment (A) groups:

- $A = 0$ No RBCs — ratio is undefined
- $A = 1$ Low platelet ratio
- $A = 2$ High platelet ratio

A high ratio indicates that patients received a platelet:RBC ratio $\geq$0.75. We focused on a joint outcome (Y) of a patient’s status at a given time after treatment where we defined success as $Y = 1$ when the patient was alive and hemostatic (no longer bleeding) and $Y = 0$ when the patient was either deceased or not yet hemostatic.

3.3 Problem Statement

Following the road map outlined in van der Laan and Rose [2011], we formally outline the statistical estimation problem.

Observed Data Structure

Moving from the PROPPR example to a general cross-sectional optimal treatment setting, we observe n i.i.d. copies $O_1, \ldots, O_n$ sampled from $O = (W, A, Y) \sim P_0$: W are the pre-treatment covariates, A is the treatment or exposure of interest, and Y is the outcome. The PROPPR example has a categorical treatment, which we generalize here as $A \in \mathcal{A} \equiv \{a_1, \ldots, a_{n_A}\}$, where $n_A = |\mathcal{A}|$ is the number of categories. We can factor the data generating distribution P_0 into three components:

$$\begin{aligned} P_0(O) &= P_0(Y|A,W)P_0(A|W)P_0(W) \\ &\equiv Q_0(Y|A,W)g_0(A|W)Q_{W,0}(W) \end{aligned}$$

where $Q_{W,0}(W)$ is the marginal distribution of the covariates, $g_0(A|W)$ is treatment mechanism, the distribution of treatments conditional on the covariates, and $Q_0(Y|A,W)$ is the distribution of the outcomes conditional on the covariates and treatment. We also define the condition mean of the outcome: $\bar{Q}_{Y,0}(A,W) \equiv E_0\left[Y|A,W\right]$.

We make no assumptions about the distribution of P_0 and so we say that $P_0 \in \mathcal{M}$ where $\mathcal{M}$ is the fully non-parametric model.

Optimal Rule

A dynamic rule is a function $d(V)$ that takes a subset of covariates $V \subseteq W$ and assigns a treatment based on them. The **value** of a dynamic rule as $E_0\left[Y^{d(V)}\right]$, that is the mean outcome if, possibly contrary to fact, everyone got treatment as assigned by the rule. The optimal rule is the rule with the maximal value: $d^0 \equiv \arg\max_{d \in \mathcal{D}} E_0\left[Y^{d(V)}\right]$. The subset V is assumed to be given by the researcher, and are selected based on subject-matter knowledge of variables that are likely to be both effect modifiers and available when a treatment decision needs to be made. In section 3.7 we describe variable importance measures to evaluate the importance of each covariate in V, as well as a procedure to select a subset of covariates.

Blip Function

A key quantity for optimal treatment is the blip function. For a binary treatment, $A \in \{0, 1\}$, we define a blip function as $\bar{Q}_0(V) \equiv E_0[Y^1 - Y^0|V]$, the average treatment effect within a stratum of V. The rule can be derived directly from this quantity $d^0(V) = I(\bar{Q}_0(V) > 0)$. This approach relies on the treatment variable being binary. In section 3.4 we extend the blip function to categorical treatments.

Statistical Parameter

The value of a rule, and therefore the optimal rule itself are causal parameters based on unobserved counterfactuals and must be identified with statistical parameters if they are to be estimated from the observed data. We need to make two assumptions to identify these causal parameters with statistical ones [van der Laan and Luedtke, 2015]:

- **Positivity:** $P_0(\min_{a\in\mathcal{A}} g_0(a|W) > 0) = 1$
- **Randomization:** $Y_a \perp A|W$ for all $a \in \mathcal{A}$

Under these assumptions, the causal blip function is equal to the following statistical blip function:

$$\bar{Q}_0(V) \equiv E_0[\bar{Q}_{Y,0}(1, W) - \bar{Q}_{Y,0}(0, W)|V]$$

And the value of a rule $E_0\left[Y_d\right]$ is equal to the following statistical parameter:

$$E_0\left[Y_d\right] \equiv E_{0,W}\left[E_0\left[Y|A = d(V), W\right]\right]$$

Therefore, our optimal rule is equal to the following statistical parameter:

$$d_0 \equiv \arg\max_{d\in\mathcal{D}} E_0\left[Y_d\right]$$

Existing Literature

Many methods of learning an optimal rule from data have been developed. See Chakraborty and Moodie [2013] for review. Here we briefly review three key approaches. The first, Q-Learning, is based on generating an estimate of $\bar{Q}_{Y,0}(A, W)$ [Sutton and Barto, 1998]. Second, Structural Nested Mean Models (SNMM), posits a model for the blip $\bar{Q}_0(V)$, and estimates its parameters using either estimating equations or regression [Robins, 2004b, Murphy, 2003]. The third, Weighted Classification, uses classification approaches to predict the optimal treatment assignment directly for each observation [B and van der Laan, 2012, Zhang et al., 2012, Zhao et al., 2014, 2012]. In this approach, the classification loss function is weighted by an estimate of the blip.

Super Learning of the optimal rule

Here we focused on the methods developed in van der Laan and Luedtke [2015] and Luedtke and van der Laan [2016a]. These papers outline a methodology for learning an optimal rule using Super Learner and estimating its value using Targeted Minimum Loss-based Estimation (TMLE). Luedtke and van der Laan [2016a] presents three main approaches for using Super Learner to estimate the optimal rule: Super Learning of the blip function (similar to SNMM approach), Super Learning the weighted classification problem, and a joint Super Learner combining the blip and weighted classification approaches. It also outlines a range of possible loss functions to use: regression type losses (for learning the blip), classification type losses (for learning the weighted classification), and performance of the rule (estimated e.g. using CV-TMLE). Currently, `opttx` only implements the blip learning approach, with extensions to the other two approaches planned.

As outlined in Luedtke and van der Laan [2016a], we estimate the blip $\bar{Q}_0(V) \equiv E_0[\bar{Q}_{Y,0}(1,W) - \bar{Q}_{Y,0}(0,W)|V]$ using Super Learner. We first estimate $\bar{Q}_{Y,0}(A,W) \equiv E_0[Y|A,W]$, and $g_0(A|W) \equiv P_0(A|W)$ using Super Learner. This allows us to apply the doubly robust A-IPW transform to our outcome [van der Laan and Robins, 2003]:

$$D_{\bar{Q}_Y,g,a}(O) \equiv \frac{I(A=a)}{g(A|W)}\left(Y - \bar{Q}_Y(A,W)\right) + \bar{Q}_Y(A=a,W) \tag{3.1}$$

$$E[D_{\bar{Q}_Y,g,a}(O)|V] \overset{*}{=} E[Y^a|V] \tag{3.2}$$

The equality in eq. (3.2) relies on the randomization and positivity assumptions outlined above, and correct estimation of at least one of $\bar{Q}_{Y,0}(A,W)$ and $g_0(A|W)$. Using this transform, we can define a contrast:

$$D^{\text{blip}}_{\bar{Q}_Y,g}(O) = D_{\bar{Q}_Y,g,a=1}(O) - D_{\bar{Q}_Y,g,a=0}(O)$$

This allows us to estimate the blip function, $\bar{Q}_{0,a}(V)$, by regressing $D^{\text{blip}}_{\bar{Q}_Y,g}(O)$ on V using Super Learner (for example with squared error loss $L(O,\bar{Q}_{n,a}) = \left(D^{\text{blip}}_{\bar{Q}_Y,g}(O) - \bar{Q}_{n,a}(V)\right)^2$). Our rule is then $d(V) = \arg\max_{a\in\mathcal{A}} \bar{Q}_{0,a}(V)$.

Inference for the mean outcome under the optimal rule.

Van der Laan and Luedtke [2015] provides CV-TMLE estimators and inference for two performance parameters: performance of truely optimal rule $E_0[Y^{d_0}]$ and performance of the rule actually learned from data $E_0[Y^{d_n}]$. The latter is an example of a data adaptive target parameter. Parameters of this type are discussed in detail in Hubbard et al. [2016]. The `opttx` package implements CV-TMLE for these parameters.

The inference approach presented in van der Laan and Luedtke [2015] relies on the assumption that the optimal treatment is unique in all strata of V. Formulating this for categorical treatment:

$$P_0\left(E_0\left[Y^{d_0(V)}|V\right] > \max_{a\neq d_0(V)} E_0\left[Y^a|V\right]\right) = 1$$

If V is high dimensional, it's reasonable to expect that there exists some strata for which there is no treatment effect. Luedtke and van der Laan [2016a] outlines an on-line estimator for which inference can be derived even for exceptional laws, but it has not yet been implemented in the `opttx` package.

Challenges

Applying the optimal treatment methodology to the PROPPR study [Holcomb et al., 2015] presents two key challenges. First, while the literature focus on optimal rules for binary treatment, we have parameterized the treatment as categorical, requiring some modifications to the methodology. Second, due to the chaotic environment associated with the treatment of traumatic injury, it is unlikely that physicians can apply rules based on a large number of covariates. Therefore, some procedure to identify a small subset of covariates that are relevant for treatment decisions would be a useful addition to an optimal rule based on the full set of covariates. The following sections describe these challenges in further detail and outline solutions.

3.4 Categorical Treatment

Much of the current literature focuses on optimal treatment in the context of binary treatment decisions. As such, learning an optimal rule is cast in terms of learning a blip function, $\bar{Q}_0(V) \equiv E_0[\bar{Q}_{Y,0}(1, W) - \bar{Q}_{Y,0}(0, W)|V]$, which focuses on the difference in average outcomes for the two treatments. As outlined in section 3.2, our PROPPR example has a categorical treatment. Therefore, we need to modify our approach to consider contrasts between more than two treatments. Two types of approaches present themselves. First, categorical treatments can be cast in a longitudinal setting as a series of binary pseudo-treatments with no intervening covariates. Luedtke and van der Laan [2016b] proposes to learning categorical optimal treatment rules by learning the optimal decision for each of these binary pseudo-treatments. The second type of approach is to directly generalize the blip function to more than two treatments.

Categorical Treatment as a Series of Binary Treatments

Categorical Treatments can always be rewritten as a series of binary treatments. For example, if the treatment variable has three levels, $A \in \mathcal{A} \equiv \{a_1, a_2, a_3\}$, it can be reformulated as a joint treatment $A' = (A'_0, A'_1) \in \{(0,0),\ (0,1),\ (1,0)\}$ where $A = \{a_1, a_2, a_3\} \mapsto A' = \{(0,0),\ (0,1),\ (1,0)\}$

The likelihood then becomes:

$$P_0(O) \equiv Q_0(Y|A'_1, A'_0, W) g_{0,A'_1}(A'_1|A'_0, W) g_{0,A'_0}(A'_0|W) Q_{W,0}(W)$$

where $g_{0,A'_1}(A'_1 = 1|A'_0 = 1, W) = 0$ by definition.

Rules for the two "treatments" A'_1 and A'_0 can then be learned from two blip functions as described in Luedtke and van der Laan [2016b]. This approach is attractive because it accommodates categorical treatments within the existing binary treatment methodology. However each additional pseudo-treatment requires an additional Super Learner for the treatment mechanism conditional on all past treatment decisions, so that in the example above, on would need a Super Learner for both $g_{0,A'_0}(A'_0|W)$ and $g_{0,A'_1}(A'_1|A'_0,W)$. Additionally, one needs a Super Learner for the blip function for each pseudo-treatment. This increases computational complexity as a function of the number of treatment categories, limiting the feasibility of this approach, especially in the case of large datasets. Although this additional cost is somewhat reduced by avoiding nested cross-validation as described in chapter 2, it still requires additional computation compared to a truly categorical approach.

Generalizing the Blip Function

We instead chose to focus on extensions to the blip function approach that allow for categorical treatment. In the binary setting, the blip function approach enjoys some advantages over Q-learning in that it is targeted to the part of the likelihood that is important for learning the optimal rule. We can factorize $E[Y|A,W] = f(A,W) + h(W)$, and therefore we can write the blip as $\bar{Q}_0(V) = E_0\left[f(1,W) - f(0,W)|\, V\right]$. This means that we can estimate the blip without having to estimate $h(W)$. Any generalization of the blip function to categorical treatment should maintain this property.

For categorical treatment, we define extensions to the blip function called "psuedo-blips". These blips are vector valued in that the output for a given V is a vector whose length is equal to the number of categories of A:

$$\bar{Q}_0^{\text{pblip}}(V) = \left\{\bar{Q}_{0,a}^{\text{pblip}}(V) : a \in \mathcal{A}\right\}$$

We consider three possible psuedo-blip formulations. First, one can choose a reference category (e.g. $A = 0$) and define the blip for all other categories relative to the reference: $\bar{Q}_{0,a}^{\text{pblip-ref}}(V) \equiv E_0[Y_a - Y_0|V]$ (as in Robins [2004a]). This has the advantage of reducing to the normal blip in the case of binary treatments. Barring some subject matter justification for choosing a particular reference category, it makes sense to choose the most frequently observed category.

Another possible is to define the blip relative to the average of all categories $\bar{Q}_{0,a}^{\text{pblip-avg}}(V) \equiv E_0[Y_a - \frac{1}{n_{\mathcal{A}}}\sum_{a'\in\mathcal{A}} Y'_a|V]$, which eliminates the need to choose a reference category.

A slight adjustment to the average reference approach is to take a weighted average, weighting by the probability of observing each treatment in strata of V: $\bar{Q}_{0,a}^{\text{pblip-wavg}}(V) \equiv E_0[Y_a - \sum_{a'\in\mathcal{A}} p(A = a'|V)Y'_a|V]$.

These three psuedo-blip approaches all have the property of identifying the optimal rule as $d(V) = \arg\max_{a\in\mathcal{A}} \bar{Q}_{0,a}(V)$ without requiring an estimate of $h(W)$. In section 3.6, we compare these three blip approaches to using no reference and simply learning $\bar{Q}_{0,a}^4(V) \equiv E_0[Y_a|V]$ as is done in Q-learning, which does require estimation of $h(W)$. Like the blip function for binary treatments, these "psuedo-blips" can be estimated by regressing contrasts composed using the A-IPW transform described above in section 3.3 on V. For example, in the case of the average approach, we can estimate $\bar{Q}_{0,a}^{\text{pblip-avg}}(V)$ by regressing the following (vector-valued) contrast on V:

$$D^{\text{pblip-avg}}_{\bar{Q}_{Y_n},g_n}(O) = \left\{ D_{\bar{Q}_{Y_n},g_n,a=a}(O) - \frac{1}{n_A}\sum_{a'\in\mathcal{A}} D_{\bar{Q}_{Y_n},g_n,a=a'}(O) : a \in \mathcal{A} \right\}$$

For the purposes of blip estimation, $\bar{Q}_{Y_n}$ and g_n are estimated using Split Specific Super Learner, as described in 2.

The `opttx` package supports estimation of all three psuedo-blips with Super Learner, using a new Super Learner implementation that supports features not supported in the `SuperLearner` package. These features are described in the next section.

3.5 Super Learner

Super Learner is an ensemble machine learning approach using cross-validation and a loss-based estimation framework. A Super Learner is specified by three components. First, a library of learners, $\hat{\psi}^j_{P_n}(X)$ for $j \in 1, \ldots, J$, a set of machine learning algorithms that generate predictions based on a training set. Next, a metalearner $m(Z;\alpha)$ where $Z^j = \hat{\psi}^j_{P_n}(X)$, e.g. $m(Z^j;\alpha) = \sum_{j=1}^J \alpha_j Z^j$, a learner that generates predictions that in some way combine predictions from the library of learners. Finally, a loss function $L(O, \hat{\psi}_{P_n})$, e.g. $L(O, \hat{\psi}_{P_n}) = \left(Y - \hat{\psi}_{P_n}(X)\right)^2$ that can be used to evaluate the performance of the learners and metalearner. Using these three components the Super Learner is defined as $\hat{\psi}_{\alpha_n,P_n} = m(Z;\alpha_n)$, where α_n is chosen via cross-validation to minimize the risk (expected value of the loss function).

While the Super Leaner algorithm is very general, the `SuperLearner` package [Polley and van der Laan, 2011] has some limitations that preclude its use in this setting. Specifically, `SuperLearner` lacks support for categorical outcomes, which is required to directly estimate $g_0(A|W)$ for categorical A. Additionally, `SuperLearner` lacks support for estimating multivariate outcomes, which will allow us to estimate $\bar{Q}^0_{0,a}(V)$ for all $a \in \mathcal{A}$ simultaneously. Finally, the dependencies between nuisance parameters in the optimal treatment setting require support for nesting cross-validation. This is discussed further in chapter 2. We've implemented a Super Learner that supports these features in the `origami` package, available on GitHub [Coyle, 2014]. Below we describe our implementations of multinomial and multivariate Super Learners.

Multinomial Super Learner

We need to estimate $g_0(A|W) = P(A|W)$ for categorical A. That is, we want to estimate $\psi_{0,a}(W) = P(A = a|W)$ for all $a \in \mathcal{A}$.

Many of the learners supported by the SuperLearner package can be applied directly to multi-class classification problems. Those learners that only support binary outcomes can be extended to support categorical outcomes using the same series of binary treatment approach described above. To full specify a multinomial Super Learner, we need a combination method (metalearner) for multinomial outcomes. Here, we used a logistic combination:

$$m(Z_a;\alpha) = \frac{\text{logit}^{-1}\left(\alpha\,\text{logit}(Z_a)\right)}{\sum_{a\in\mathcal{A}} \text{logit}^{-1}\left(\alpha\,\text{logit}(Z_a)\right)}$$

For a loss function, we used negative multinomial log likelihood:

$$L(O,\psi_\alpha(X)) = -\sum_{a\in\mathcal{A}} I(A=a)\log(\psi_{\alpha,a}(X))$$

For binary treatments, this reduces to the binomial log likelihood.

Multivariate Super Learner

We also need to estimate $\bar{Q}_{0,a}(V)$ for all $a \in \mathcal{A}$. We can think of this as a vector-valued or multivariate outcome, the vector of $\bar{Q}^0_{0,a}(V)$ across $\mathcal{A}$. Of course, we can simply apply Super Learner for univariate outcomes separately for each a, but in cases where the blip function is similar for all values of a, we could gain efficiency by estimating a combination that applies to all a's. We accomplished this by using a loss that averages across the a's:

$$L(O,\psi_\alpha(X)) = \frac{1}{n_A}\sum_{a\in\mathcal{A}} L(0,\psi_\alpha(X))$$

Another nice feature of the multivariate approach is that it allows us to use a loss function that directly targets a rule's value:

$$\begin{aligned} L_{\bar{Q}_Y,g}(O,\psi_\alpha(X)) &= -D_{\bar{Q}_Y,g,a=d(V)}(O) \\ \text{where } d(V) &= \arg\max_{a\in\mathcal{A}} \bar{Q}^0_{0,a}(V) \end{aligned}$$

3.6 Pseudo-blip Comparison

Using a Monte Carlo simulation, we compared the three different pseudo-blip functions in terms of how well the learned rule performed. We drew 1000 samples of size $n = 1000$ from the following distribution:

$$\begin{aligned}
O &= (W, A, Y) \\
P(O) &= P(Y = 1|A, W)P(A = a_i|W)f(W) \\
f(W) &= \mathcal{N}(\mathbf{0}_5, \boldsymbol{I}_{5,5}) \\
P(A = a_i|W) &= \frac{\operatorname{logit}^{-1}(W_i)}{\sum_{i=1}^{3} \operatorname{logit}^{-1}(W_i)} \\
P(Y = 1|A, W) &= 0.5 \operatorname{logit}^{-1} [-5I(A = a_2)(W_1 - 0.5) \\
&\qquad\qquad\qquad +5I(A = a_3)(W_1 - 0.5)] \\
&+ 0.5 \operatorname{logit}^{-1} (W_2 W_3)
\end{aligned}$$

Results

Figure 3.1 shows the performance in terms of the mean outcome of the learned rule for the three different blip approaches. All the results are fairly similar, but in general squared error loss outperformed rule value loss. Additionally, the reference category approach (pblip-ref above), was appreciably worse than any of the other approaches including the Q-learning approach (no reference). In this simulation, the best performing approach was the “average reference” (pblip-avg above) approach with squared error loss, and this was used for the remaining simulations and PROPPR data analysis.

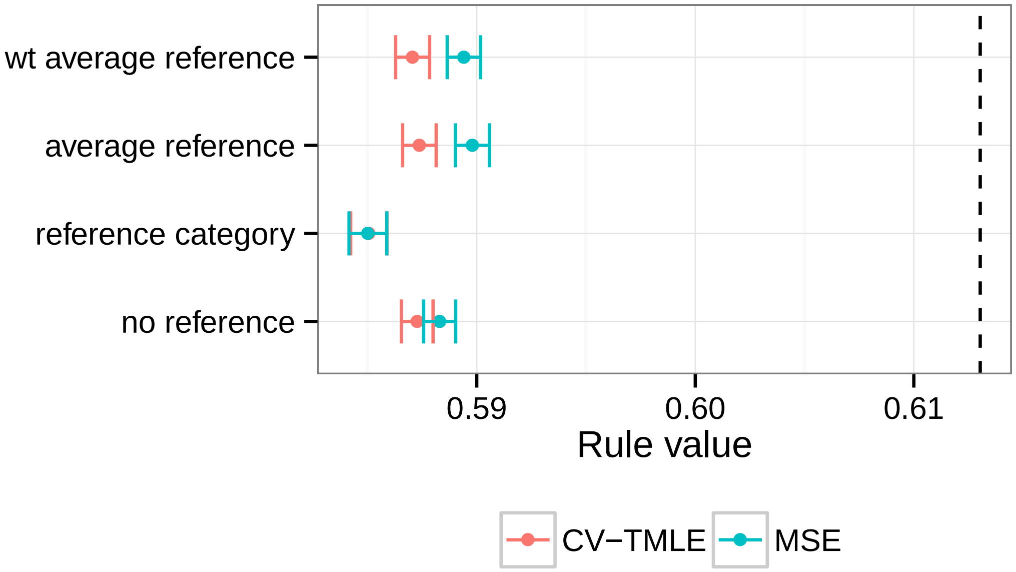


Figure 3.1: Pseudo-blip Comparison. The dashed line indicates the mean outcome under the true optimal rule. Rules learned using a squared error loss are indicated in blue, and rule learned using a CV-TMLE estimate of the rule value are indicated in red.

3.7 Variable Importance

An optimal rule that is learned using Super Learner as described above is somewhat opaque: it can be thought of as a black box function that takes a vector of covariates V as input and returns a treatment decision A. One could imagine implementing such a function in software on a computer or mobile platform, enabling researchers and practitioners to make optimal treatment decisions in a clinical setting. However, this black box approach presents several barriers to practical implementation. First, without some insight into how a rule is making treatment decisions, clinicians are unlikely to trust a black box over their own clinical judgment. Second, if a rule is based on a large number of covariates, it might be infeasible to collect and enter all the required covariates before a treatment decision needs to be made. We hope to illuminate the black box and improve the feasibility of implementing a treatment rule by determining which covariates in particular are influential in making treatment decisions. To gain insight into the learned optimal treatment rule, we appeal to variable importance measures (VIM). Specifically, we define variable importance in terms of a risk difference, as was done in Dudoit et al. [2003]. Consider a partition of the covariates on which the rule is based, $V = \{V_j : j \in J = \{1, \ldots, d\}\}$, into a subset of covariates V_S for some $S \subset J$ and its complement $\bar{V}_S = V \setminus V_S$. We can then evaluate the importance of the subset V_S by calculating a risk difference comparing the performance of estimators including and excluding this subset: $\psi^{\text{VIM}}_{0,V_S,L} = E_0[L(O,\bar{Q}_0(V)) - L(O,\bar{Q}_0(\bar{V}_S))]$. Here, we focus on univariate subsets (i.e. $S \in \{1,\ldots,d\}$, to determine individual variable importance for each covariate. This framework also allows us to evaluate the variable importance of sets of covariates.

Within this framework, we consider three possible loss functions from Luedtke and van der Laan [2016b], capturing different aspects of performance for learning an optimal rule. First, a (multivariate) regression-type loss function such as the squared error loss:

$$L(O,\bar{Q}^{\text{pblip}}_0(V)) = \frac{1}{n_A}\sum_{a\in\mathcal{A}}\left(D^{\text{pblip}}_{\bar{Q}_Y,g,a}(O) - \bar{Q}^{\text{pblip}}_{0,a}(V)\right)^2$$

.

A risk difference calculated using this loss function indicates how the learning of the blip function is impacted by omitting a particular covariate.

Second we can use a classification type loss function, such as the 0-1 loss:

$$L(O,\bar{Q}^{\text{pblip}}_0(V)) = I\left[\left(\arg\max_{a\in\mathcal{A}} D^{\text{pblip}}_{\bar{Q}_Y,g,a}(O)\right) \neq \left(\arg\max_{a\in\mathcal{A}} \bar{Q}^{\text{pblip}}_{0,a}(V)\right)\right]$$

A risk difference based on a classification loss indicates how treatment assignment changes when a particular covariate is omitted.

Finally, we can directly use the inverse of the rule value as a loss:

$$L(O,\bar{Q}^{\text{pblip}}_0(V)) = -\bar{Q}_{Y,0}(A = \arg\max_{a\in\mathcal{A}} \bar{Q}^{\text{pblip}}_{0,a}(V), W)$$

A risk difference based on rule value indicates how mean outcome is impacted by omitting a particular covariate. This is perhaps the most relevant loss function, as it directly informs the impact of each covariate on optimizing the outcome of interest.

Estimating this risk difference in turn for each of the covariates in V gives variable importance measures that can be compared to estimate the relative importance of the covariates comprising V. This requires Super Learning of the rule for each subset $\bar{V}_S$ that needs to be studied, a computationally demanding procedure. This increases the importance of creating an optimal treatment implementation that is computationally efficient.

The above loss functions, and therefore the corresponding variable importance parameters, are all defined in terms of the *true* blip functions ($\bar{Q}^{\text{pblip}}_0(V)$) for various subsets V_S. If instead we use the *estimated* blip functions ($\bar{Q}^{\text{pblip}}_n(V)$), we can define data-adaptive variable importance parameters. That is, instead of asking "how important is V_S for the true rule", we can ask "how important is V_S for the learned rule". While the parameters based on the true blip functions are relevant for understanding how covariates impact the truly optimal treatment rule, we argue that these data-adaptive parameters are more relevant in that they aid understanding of how covariates impact the rule that was actually learned. This allows practitioners to understand how the rule that they could implement is functioning, and understand how rules that rely on fewer variables compare. Therefore, the simulation in the next section focuses on our ability to learn the data-adaptive variable importance parameters.

Simulation

We studied the performance of estimators for the three different VIM parameters outlined above via simulation. We generated 1000 samples of size n=1000 from the following data generating distribution:

$$\begin{aligned}
O &= (W, A, Y) \\
P(O) &= P(Y=1|A,W)P(A=a_i|W)f(W) \\
f(W) &= \mathcal{N}(\mathbf{0}_5, \boldsymbol{I}_{5x5}) \\
P(A=a_i|W) &= \frac{\text{logit}^{-1}(W_i)}{\sum_{i=1}^{3}\text{logit}^{-1}(W_i)} \\
P(Y=1|A,W) &= 0.5\,\text{logit}^{-1}\left(3\sum_{i=1}^{3} I(A=a_i) * (i * W_i - 0.5)\right) \\
&\quad + 0.5\,\text{logit}^{-1}\left(W_2 W_3\right)
\end{aligned}$$

This simulated data generating distribution was also used to demonstrate the usage of the opttx package in appendix C. For each simulation iteration, we calculated the true values of the three data-adaptive variable importance measures using a large (n=1,000,000) test sample, and estimated the values using cross-validated risk difference estimators. We compared cross-validated estimates of the three VIM measures above to their true values, with the goal of evaluating which VIM parameters are easiest to estimate from data, as well as which most readily identify the important covariates for making optimal treatment decisions.

Results

Figure 3.2 depicts the true values and estimates values for the three data-adaptive variable importance parameters. The three measures generally agree as to the ordering of the covariates by importance, with W_3 being most important, followed by W_2, and W_1, and with W_4 and W_5 having no importance, as should be expected for pure noise covariates. There is generally good correspondence between estimated and true values.

Figure 3.3 compares the performance of estimators for the three variable importance measures. Of the three, the regression parameter has the lowest MSE. Therefore, although the regression parameter is the least interpretable of the three parameters, we suggest using it as it appears to be the easiest to estimate from data.

Figure 3.4 compares the rank-order of variables by importance (higher rank is more important) for the three methods. The ability to correctly order variables by importance is necessary for variable selection procedures like backward selection. We can see that all three approaches generally correctly separate the variables into two groups: the not important group (W_4 and W_5), and the important group (W_1, W_2, and W_3). This separation is most pronounced in the regression approach, further supporting our recommendation that this be the preferred approach. Within the important group, the ordering is less consistent than could be hoped. It's possible that is because the variables in this group are of similar importance, and it is hard to correctly estimate the ordering. Again, the regression approach appears to most consistently order the variables by importance.

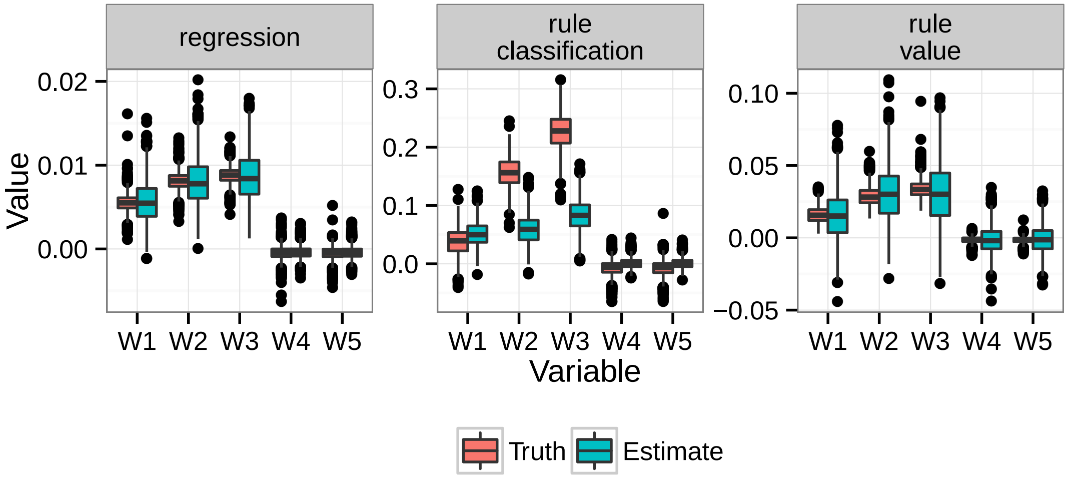


Figure 3.2: True and Estimated Data-Adaptive VIM Values. True values of the three data-adaptive VIM parameters were calculated using a large (n=1,000,000) test sample. Estimates are from cross-validated risk difference estimators.

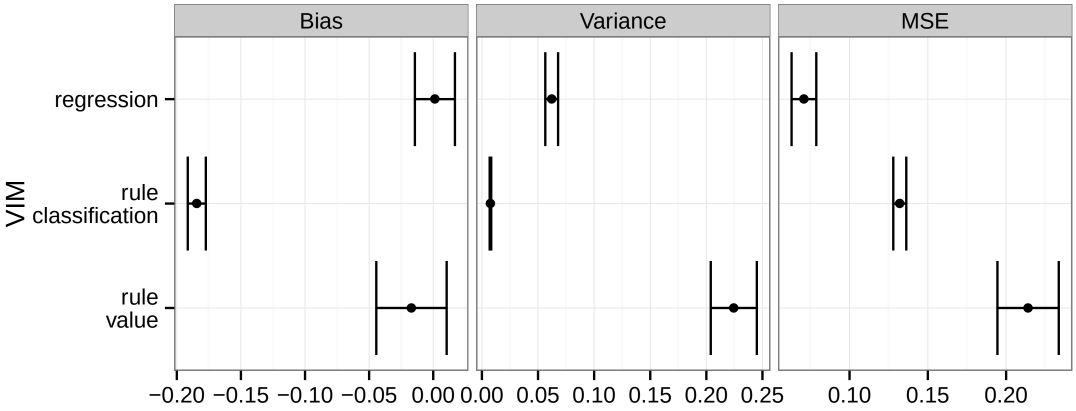


Figure 3.3: VIM Performance Comparison. To facilitate comparison between measures, performance was normalized by dividing by the mean true value of the variable importance measure.

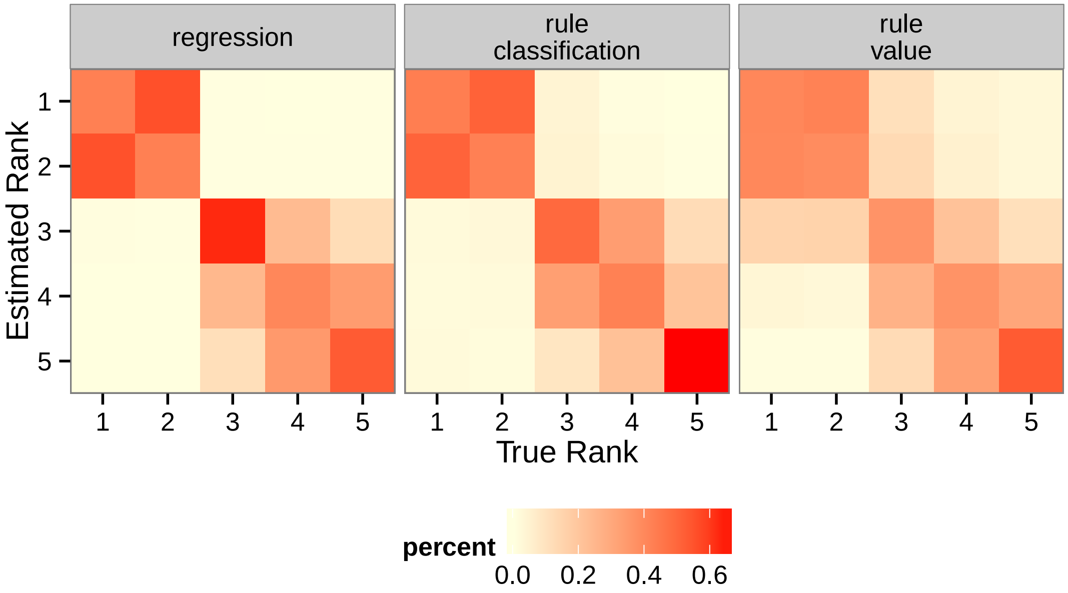


Figure 3.4: VIM Rank Comparison. A comparison of the rank-order of variables by importance (higher rank is more important) for the three methods. The ability to correctly order variables by importance is necessary for variable selection procedures like backward selection.

3.8 PROPPR Data Example

Having evaluated the performance of the implemented methodologies via simulation, we then applied the methods to the PROPPR data example [Holcomb et al., 2015]. Unfortunately, at many time points, there seemed to be very little treatment effect of Platelet:RBC ratio on the probability of achieving hemostasis for most strata of V. Therefore, for the sake of illustration, we present results from a time point with a particularly pronounced apparent treatment effect: the effect of Platelet:RBC ratio at one hour after admission to the hospital on the probability of achieving hemostasis at two hours after admission.

Figure 3.5 shows estimates of the mean outcome (probability of hemostasis) as observed, under the learned treatment rule, and under static treatments where everyone was assigned to each of the three treatment levels. These estimates were obtained using CV-TMLE. There is little evidence to support the conclusion that the learned rule outperforms the best static treatment (a low platelet to RBC ratio), or the observed treatment regime. Although we lack evidence to recommend the use of a dynamic treatment rule in this study, we proceeded to evaluate which variables were important to the rule we did learn. Figure 3.6 shows the results of this analysis, using the regression variable importance measure. By far the most important variables were injury type (blunt or penetrating) and pulse. Platelet count, lactate, and clotting factor were determined to be of middling importance. Unfortunately, it was difficult to find a setting in which treatment had a definitive impact on outcome in the PROPPR study. This made it difficult to effectively demonstrate optimal treatment methodologies using this study. It is possible that an optimal rule to assign blood transfusion regimes to patients could be learned to maximize survival, but learning such a rule might require a more comprehensive set of covariates, more carefully defined treatment and outcomes, and/or a larger sample size.

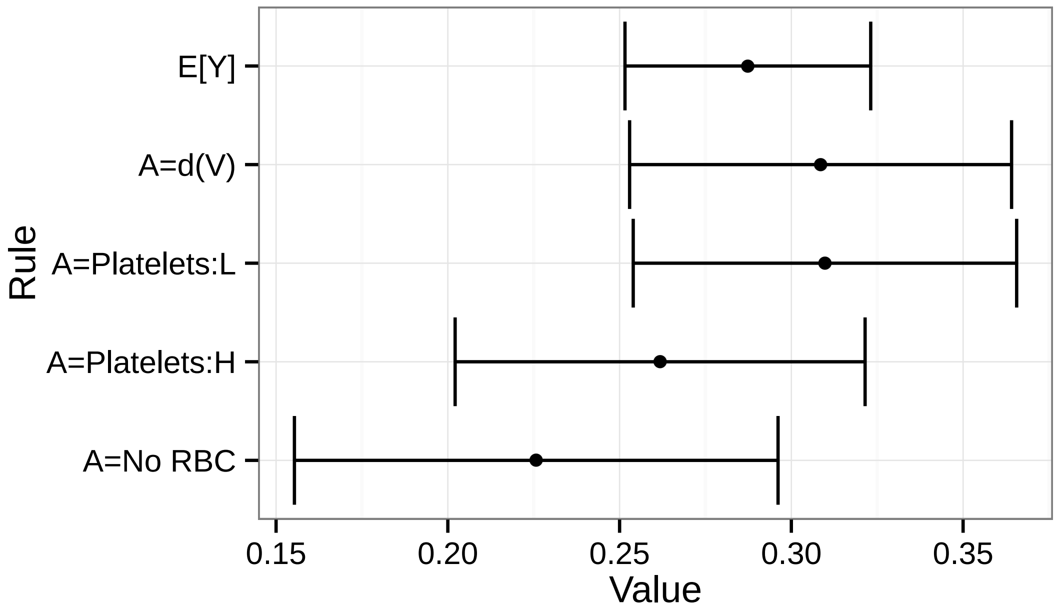


Figure 3.5: Mean Outcomes from the PROPPR Study

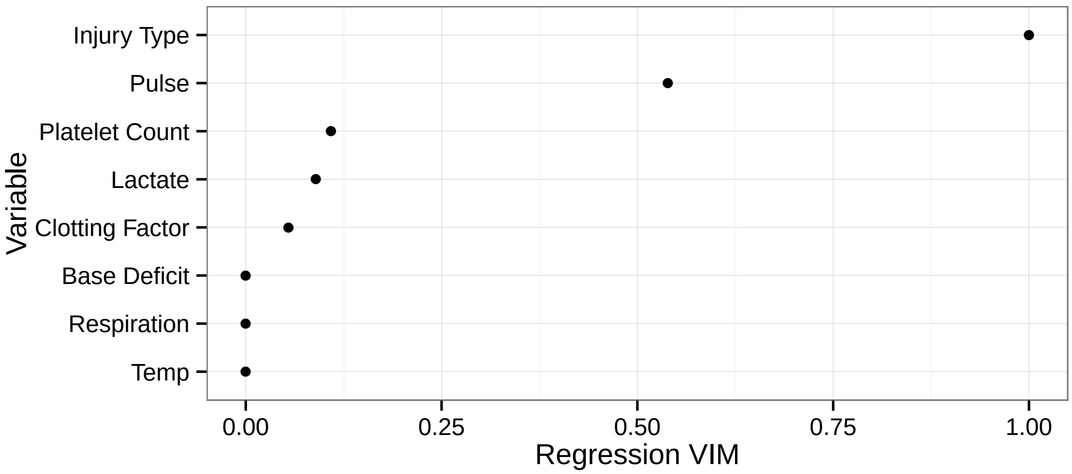


Figure 3.6: Variable Importance Measures for PROPPR Study

3.9 Conclusion

Although development of the `opttx` package is ongoing, with a number of important features still not implemented, several key challenges to a practical implementation of the optimal treatment methodology developed in Luedtke and van der Laan [2016b] have been overcome, with solutions implemented in software. These solutions include the Split Sequential Super Learner described in chapter 2, the use of categorical blip functions described in section 3.4, and novel variable importance measures described in section 3.7. As it currently exists, the `opttx` package can then be understood as a proof of concept for these solutions, with a focus on evaluating their performance using simulation.

As noted throughout the paper, there are many important components of the optimal treatment methodology not yet implemented in the `opttx` package. These include support for longitudinal optimal treatment problems, weighted classification and joint Super Learning approaches for the learning of the rule, support for continuous outcomes (Y), support for right censored outcomes, and support for the on-line Super Learner to fix inference in the case of strata of V with no treatment effect. Due to these limitations the `opttx` package is rather limited in its applicability to a range of important datasets. These features are all planned extensions to the current package, and will be implemented as part of an upcoming software development effort. We therefore expect that a more complete and accessible implementation of optimal treatment will be available shortly.

Appendix A

Targeted Bootstrap Theory

This appendix contains details of the theoretical results for the Targeted Bootstrap[1].

A.1 TMLE for the asymptotic variance of the target parameter

Asymptotic Efficiency

As with any TMLE we have the following general theorem for this TMLE $\tilde{P}_n^*$.

Theorem 4. *Consider $O \sim P_0 \in \mathcal{M}$. Let $\Sigma^2 : \mathcal{M} \to I\!R$ be defined by $\Sigma^2(P) = PD^*(P)^2$, where $D^*(P)$ is the efficient influence curve of $\Psi : \mathcal{M} \to I\!R$ at P. Let $D^*_\Sigma(P)$ be the efficient influence curve of Σ^2 at P. Let $R_\Sigma(P, P_0)$ be defined by*

$$\Sigma^2(P) - \Sigma^2(P_0) = -P_0 D^*_\Sigma(P) + R_\Sigma(P, P_0).$$

Using the representation above, $R_\Sigma(P, P_0)$ involves second order differences $Q_\Sigma(P) - Q_\Sigma(P_0)$ and $g_\Sigma(P) - g_\Sigma(P_0)$

Assume that $P_n D^*_\Sigma(\tilde{P}_n^*) = o_P(1/\sqrt{n})$, $D^*_\Sigma(\tilde{P}_n^*)$ falls in a P_0-Donsker class with probability tending to 1, $P_0\{D^*_\Sigma(\tilde{P}_n^*) - D^*_\Sigma(P_0)\}^2 \to 0$ in probability as $n \to \infty$, and $R_\Sigma(\tilde{P}_n^*, P_0) = o_P(1/\sqrt{n})$. Then, $\Sigma^2(\tilde{P}_n^*)$ is an asymptotically efficient estimator of $\Sigma^2(P_0)$ at P_0:

$$\Sigma^2(\tilde{P}_n^*) - \Sigma^2(P_0) = (P_n - P_0) D_{\Sigma^2}(P_0) + o_P(1/\sqrt{n}).$$

In particular, the confidence interval $\Sigma^2(\tilde{P}_n^*) \pm 1.96\sigma_n/\sqrt{n}$ with σ_n^2 being a consistent estimator of $P_0 D_{\Sigma^2}(P_0)$, such as $\sigma_n^2 = P_n D_{\Sigma^2}(\tilde{P}_n^*)$, contains $\Sigma^2(P_0)$ with probability tending to 0.95, under the assumptions of the above theorem.

In order to be able to apply this theorem $\tilde{P}_n^*$ will need to be a probability distribution compatible with both $Q^*_{\Sigma,n}$ and $g_{\Sigma,n}$ in the above definition of the TMLE in terms of $(Q^*_{\Sigma,n}, g_{\Sigma,n})$. As we will see for consistency of the targeted bootstrap we only need that $\tilde{P}_n^*$ is compatible with $Q^*_{\Sigma,n}$.

[1] All results in this section due to Mark van der Laan

Application to the treatment specific mean

Specifically applying theorem 4 to the case where the target parameter is the treatment specific mean:

Theorem 5. *Consider $O \sim P_0 \in \mathcal{M}$ and $\mathcal{M}$ is the nonparametric model. Let $\Sigma^2 : \mathcal{M} \to I\!R$ be defined by $\Sigma^2(P) = PD^*(P)^2$, where $D^*(P)$ is the efficient influence curve of $\Psi : \mathcal{M} \to I\!R$ at P defined by $\Psi(P) = P\bar{Q}(P)$.*

Let $\tilde{P}_n^*$ be the above defined TMLE targeting $\Sigma^2(P_0)$, implied by $(Q_n^*, \bar{g}_n^*)$: i.e. $\tilde{P}_n^*(W = w) = Q_{W,n}(w)$, $\tilde{P}_n^*(A = 1 \mid W) = \bar{g}_n^*(W)$, and $\tilde{P}_n^*(Y = 1 \mid A, W) = \bar{Q}_n^*(W)$. Assume that $P_n D_\Sigma^*(\tilde{P}_n^*) = o_P(1/\sqrt{n})$, $D_\Sigma^*(\tilde{P}_n^*)$ falls in a P_0-Donsker class with probability tending to 1, $P_0\{D_{\Sigma^2}(\tilde{P}_n^*) - D_\Sigma^*(P_0)\}^2 \to 0$ in probability as $n \to \infty$, and $R_\Sigma(\bar{g}_n^*, Q_n^*, \bar{g}_0, Q_0) = o_P(1/\sqrt{n})$. Then, $\Sigma^2(\tilde{P}_n^*)$ is an asymptotically efficient estimator of $\Sigma^2(P_0)$ at P_0:

$$\Sigma^2(\tilde{P}_n^*) - \Sigma^2(P_0) = (P_n - P_0)D_{\Sigma^2}(P_0) + o_P(1/\sqrt{n}).$$

A.2 Joint TMLE for target parameter and its asymptotic variance

Asymptotic Efficiency

Analogous to theorem 4, we can prove asymptotic efficiency for the joint TMLE:

Theorem 6. *Consider $O \sim P_0 \in \mathcal{M}$. Let $\Psi : \mathcal{M} \to I\!R$ be a parameter with efficient influence curve $D^*(P)$. Let $\Sigma^2 : \mathcal{M} \to I\!R$ be defined by $\Sigma^2(P) = PD^*(P)^2$ and denote its efficient influence curve with $D_\Sigma^*(P)$.*

Let $\tilde{P}_n^*$ be aTMLE targeting $(\Psi(P_0), \Sigma^2(P_0))$, so that $(\Psi(\tilde{P}_n^*, \Sigma^2(\tilde{P}_n^*))$ is the TMLE of (ψ_0, σ_0^2), satisfying $P_n D_{\Sigma^2}(\tilde{P}_n^*) = o_P(1/\sqrt{n})$ and $P_n D^*(\tilde{P}_n^*) = o_P(1/\sqrt{n})$. In addition, we assume 1) $D^*(\tilde{P}_n^*)$ and $D_\Sigma^*(\tilde{P}_n^*)$ falls in a P_0-Donsker class with probability tending to 1, 2) $P_0\{D_\Sigma^*(\tilde{P}_n^*) - D_\Sigma^*(P_0)\}^2 \to 0$ and $P_0\{D^*(\tilde{P}_n^*) - D^*(P_0)\}^2 \to 0$ in probability as $n \to \infty$, 3) $R_\psi(\tilde{P}_n^*, P_0) = o_P(1/\sqrt{n})$, and $R_\Sigma(\tilde{P}_n^*, P_0) = o_P(1/\sqrt{n})$.

Then, $(\Psi(\tilde{P}_n^*), \Sigma^2(\tilde{P}_n^*)$ is an asymptotically efficient estimator of $(\Psi(P_0), \Sigma^2(P_0))$ at P_0:

$$\begin{aligned}\Psi(\tilde{P}_n^*) - \Psi(P_0) &= (P_n - P_0)D^*(P_0) + o_P(1/\sqrt{n})\\ \Sigma^2(\tilde{P}_n^*) - \Sigma^2(P_0) &= (P_n - P_0)D_\Sigma^*(P_0) + o_P(1/\sqrt{n}).\end{aligned}$$

Application to the treatment specific mean

Specifically applying theorem 6 to the case where the target parameter is the treatment specific mean:

Theorem 7. *Consider $O \sim P_0 \in \mathcal{M}$ and $\mathcal{M}$ is the nonparametric model. Let $\Psi : \mathcal{M} \to I\!R$ be defined by $\Psi(P) = P\bar{Q}(P)$ and let $D^*(P)$ be its efficient influence curve. Let $\Sigma^2 : \mathcal{M} \to I\!R$*

be defined by $\Sigma^2(P) = PD^*(P)^2$, and let $D^*_\Sigma(P)$ be its efficient influence curve. In addition, let $R_\psi(P, P_0)$ and $R_\Sigma(P, P_0)$ be the corresponding second order terms as explicitly defined above.

Let $\tilde{P}^*_n$ be the above defined TMLE targeting $(\Psi(P_0), \Sigma^2(P_0))$, so that $(\Psi(\tilde{P}^*_n, \Sigma^2(\tilde{P}^*_n))$ is the TMLE of (ψ_0, σ^2_0). Here $\tilde{P}^*_n$ is the probability distribution implied by $(Q^*_n, \bar{g}^*_n)$: i.e. $\tilde{P}^*_n(W = w) = Q_{W,n}(w)$, $\tilde{P}^*_n(A = 1 \mid W) = \bar{g}^*_n(W)$, and $\tilde{P}^*_n(Y = 1 \mid A, W) = \bar{Q}^*_n(W)$.

Assume that $P_n D^*_\Sigma(\tilde{P}^*_n) = o_P(1/\sqrt{n})$ and $P_n D^*(\tilde{P}^*_n) = o_P(1/\sqrt{n})$. In addition, we assume 1) $D^*(\tilde{P}^*_n)$ and $D_\Sigma(\tilde{P}^*_n)$ falls in a P_0-Donsker class with probability tending to 1, 2) $P_0\{D_\Sigma(\tilde{P}^*_n) - D_\Sigma(P_0)\}^2 \to 0$ and $P_0\{D^*(\tilde{P}^*_n) - D^*(P_0)\}^2 \to 0$ in probability as $n \to \infty$, 3) $R_\psi(\tilde{P}^*_n, P_0) \equiv P_0 \frac{\bar{g}^*_n - \bar{g}_0}{\bar{g}^*_n}(\bar{Q}^*_n - \bar{Q}_0) = o_P(1/\sqrt{n})$, and $R_\Sigma(\bar{g}^*_n, Q^*_n, \bar{g}_0, Q_0) = o_P(1/\sqrt{n})$.

Then, $(\Psi(\tilde{P}^*_n), \Sigma^2(\tilde{P}^*_n)$ is an asymptotically efficient estimator of $(\Psi(P_0), \Sigma^2(P_0))$ at P_0:

$$\begin{aligned} \Psi(\tilde{P}^*_n) - \Psi(P_0) &= (P_n - P_0)D^*(P_0) + o_P(1/\sqrt{n}) \\ \Sigma^2(\tilde{P}^*_n) - \Sigma^2(P_0) &= (P_n - P_0)D_{\Sigma^2}(P_0) + o_P(1/\sqrt{n}). \end{aligned}$$

Sufficient conditions for the assumptions 1), 2), and 3) are that $\parallel \bar{g}^*_n - \bar{g}_0 \parallel_{P_0} = o_P(n^{-1/4})$, $\parallel \bar{Q}^*_n - \bar{Q}_0 \parallel_{P_0} = o_P(n^{-1/4})$, $\bar{g}_0 > \delta > 0$ a.e., $Pr(\parallel \bar{g}^*_n, \bar{Q}^*_n \parallel^*_v < M) \to 1$ for some $M < \infty$, where $\parallel h \parallel_{P_0} = \sqrt{\int h^2(o) dP_0(o)}$, and $\parallel h \parallel^*_v$ is the uniform sectional variation norm for a multivariate real valued function h.

A.3 Targeted Bootstrap

Performance of cross-validation selector on targeted bootstrap samples.

The following theorem can be applied to establish the behavior of the super-learner $\hat{P}$ that plays the role as initial estimator in the definition of the TMLE $\hat{P}^*$, but now as estimator of $\tilde{P}^*_n$ when applied to bootstrap samples from $\tilde{P}^*_n$. In the definition of the TMLE $\hat{P}^*$ one typically on constructs a targeted estimator of a parameter $Q : \mathcal{M} \to Q(\mathcal{M})$ so that $\Psi(P) = \Psi_1(Q(P))$ for some Ψ_1, and the TMLE will then possibly also rely on a nuisance parameter $g : \mathcal{M} \to Q(\mathcal{M})$ chosen so that $D^*(P) = D^*(Q(P), g(P))$ only depends on P through $Q(P), g(P)$. In that case, the TMLE only relies on initial estimator of both Q, g, and the following theorem can then be applied to establish the behavior of this initial estimator on bootstrap samples, assuming that the estimators are based on using cross-validation to select among many candidate estimators, as in the super-learner. So in the following theorem Q plays the role of either P, Q, g in the above.

Theorem 8. *Suppose we observe n independent identically distributed copies of $O^{\#}_i \sim \tilde{P}^*_n$, given P_n. Let $\mathcal{M}$ be the statistical model for P_0. Let $Q : \mathcal{M} \to Q(\mathcal{M})$ be the parameter mapping of interest and let $Q^*_n = Q(\tilde{P}^*_n)$ be the true value For a given random split $B_n \in \{0, 1\}^n$ with $p = 1/n \sum_{i=1}^n B_n(i)$, let $P^{\#0}_{n,B_n}, P^{\#1}_{n,B_n}$ be the empirical distributions of* *the training and validation sample. Let* $(O, Q) \rightarrow L(Q)(O)$ *be a loss function so that* $\tilde{P}_n^* L(Q_n^*) = \min_{Q \in Q(\mathcal{M})} \tilde{P}_n^* L(Q)$. *Let* $P_{n,\#}$ *be the empirical probability distribution of* $O_i^{\#}$, $i = 1, \ldots, n$. *Let* $\hat{Q}_k : \mathcal{M}_{np} \rightarrow Q(\mathcal{M})$, $k = 1, \ldots, K(n)$, *be a set of given estimators of* Q_0. *Suppose that for each* k, $\hat{Q}_k(P_{n,\#}) \in Q(\mathcal{M})$ *with probability 1.*

Let $k_n^{\#} = \hat{K}(P_{n,\#}) = argmin_k E_{B_n} P_{n,B_n}^{\#1} L(\hat{Q}_k(P_{n,B_n}^{\#0}))$ *be the cross-validation selector, and let* $\tilde{k}_n^{\#} = \tilde{K}(P_{n,\#}) = argmin_k E_{B_n} \tilde{P}_n^* L(\hat{Q}_k(P_{n,B_n}^{\#0}))$ *be the comparable benchmark (oracle) selector.*

We define a loss-based dissimilarity

$$d_n(Q, Q_n^*) \equiv E_{B_n} P_0 \{L(Q) - L(Q_n^*)\}.$$

11 ***A****ssumptions* 12 ***A****1. There exist a* $M_1 < \infty$ *so that*

$$\sup_{Q \in Q(\mathcal{M})} \sup_{O} \mid L(Q)(O) - L(Q_n^*)(O) \mid \leq M_1,$$

where the supremum over O *is taken over a support of the distribution* $\tilde{P}_n^*$ 13 *.****A****2. There exist a* $M_2 < \infty$ *so that*

$$\sup_{Q \in Q(\mathcal{M})} \frac{VAR_{\tilde{P}_n^*}\left[L(Q) - L(Q_n^*)\right](O)}{\tilde{P}_n^*\left[L(Q) - L(Q_n^*)\right](O)} \leq M_2. \tag{A.1}$$

F*inite Sample Result. For any* $\delta > 0$, *there exists a* $C(M_1, M_2, \delta) \leq C/\delta$ *for some* $C(M_1 M_2)$ *so that*

$$\begin{aligned} E_{\tilde{P}_n^*} d_n(\hat{Q}_{\hat{K}(P_{n,\#})}(P_{n,B_n}^{\#0}), Q_n^*) &\leq (1+\delta) E_{\tilde{P}_n^*} d_n(\hat{Q}_{\tilde{K}(P_{n,\#})}(P_{n,B_n}^{\#0}), Q_n^*) \\ &\quad + C(M_1, M_2, \delta) \frac{\log(K(n))}{np}. \end{aligned}$$

A*symptotic equivalence of cross-validation selector and oracle selector on bootstrap samples: Suppose that, for almost all* $(P_n : n)$,

$$\frac{E_{\tilde{P}_n^*} d_n(\hat{Q}_{\tilde{K}(P_{n,\#})}(P_{n,B_n}^{\#0}), Q_n^*)}{\log K(n)/(np)} \rightarrow 0,$$

as $n \rightarrow \infty$. *Then, for almost all* $(P_n : n)$, *we have*

$$\frac{E_{\tilde{P}_n^*} d_n(\hat{Q}_{\hat{K}(P_{n,\#})}(P_{n,B_n}^{\#0}), Q_n^*)}{E_{\tilde{P}_n^*} d_n(\hat{Q}_{\tilde{K}(P_{n,\#})}(P_{n,B_n}^{\#0}), Q_n^*)} \rightarrow 1.$$

As a consequence, under the same conditions as needed for the oracle inequality for the cross-validation selector on the original sample, but making sure to use bounds M_1, M_2 that are uniformly in $P_0 \in \mathcal{M}$, we have the same oracle inequality for the cross-validation selector on the bootstrap sample. As a consequence, the cross-validation selector behaves as well on the bootstrap samples as it does on the original sample. This is very important, since one of our key conditions was that the second order term $R(P_n^{\#*}, \tilde{P}_n^*)$ for the TMLE on the bootstrap sample is still $o_P(1/\sqrt{n})$, given that it is true for $R(\tilde{P}_n^*, P_0)$.

Consistency

Let $O_1^{\#},\ldots,O_n^{\#} \sim_{iid} \tilde{P}_n^*$, given P_n, $P_{n,\#}$ is the corresponding empirical distribution, and $\tilde{P}_n^*$ is such that $\Sigma^2(\tilde{P}_n^*)$ is a TMLE of $\Sigma^2(P_0)$. The bootstrap distribution is defined as the probability distribution $F_n^{\#}$ of $\sqrt{n}(\Psi(P_{n,\#}^*)-\Psi(\tilde{P}_n^*))$, given P_n, where $P_{n,\#}^*$ is the TMLE targeting ψ_0 applied to the bootstrap empirical distribution $P_{n,\#}$. Let F_n be the probability distribution of $\sqrt{n}(\Psi(P_n^*)-\Psi(P_0))$. For simplicity, let $F_n, F_n^{\#}$ be cumulative distribution functions of these probability distributions. One views $F_n^{\#}$ as an estimate of the distribution F_n. Assume the regularity conditions guaranteeing that F_n converges to $F_0=N(0,\Sigma^2(P_0))$. Asymptotic consistency of the bootstrap is defined by $F_n^{\#}$ converging to F_0, given $(P_n:n)$. We like to prove that indeed our targeted bootstrap distribution $F_n^{\#}$ converges to F_0, under appropriate regularity conditions. In this manner, one can use $F_n^{\#}$ as a finite sample bootstrap distribution (which usually picks up second order terms/finite sample variability) of $\Psi(P_n^*)$, while one still guarantees that we obtain an asymptotically valid confidence interval. Of course, one could also use $\Psi(P_n^*)\pm 1.96\sigma_n^0/\sqrt{n}$ for a consistent estimator σ_n^{20} of $P_0D^*(P_0)^2$, such as $\sigma_n^*=\sqrt{\Sigma^2(\tilde{P}_n^*)}$, but a bootstrap confidence interval is based on a finite sample probability distribution and that can improve finite sample coverage.

By definition of the TMLE $P_{n,\#}^*$ targeting ψ_0, we have

$$P_{n,\#}D^*(P_{n,\#}^*)=0.$$

By application of our general identity $P_0D^*(P)=\Psi(P_0)-\Psi(P)+R_\psi(P,P_0)$, we have

$$\tilde{P}_n^*D^*(P_{n,\#}^*)=\Psi(\tilde{P}_n^*)-\Psi(P_{n,\#}^*)+R_\psi(P_{n,\#}^*,\tilde{P}_n^*).$$

As a consequence, we have

$$\Psi(P_{n,\#}^*)-\Psi(\tilde{P}_n^*)=(P_{n,\#}-\tilde{P}_n^*)D^*(P_{n,\#}^*)+R_\psi(P_{n,\#}^*,\tilde{P}_n^*).$$

Suppose that $\sqrt{n}R_\psi(P_{n,\#}^*,\tilde{P}_n^*)\to 0$ in probability, given almost $(P_n:n)$. This relies on the TMLE $\hat{P}^*:\mathcal{M}_{np}\to\mathcal{M}$ targeting ψ_0 to behave as well under sampling from P_0 as under sampling from $\tilde{P}_n^*$, or, at least, not much worse, so that the second order term still converges to zero at a faster rate than $1/\sqrt{n}$. In the next section we show that this can be expected when using the super-leaner or other cross-validation based estimators. In fact, since $\tilde{P}_n^*$ is itself a super-learner based fit (with a parametric model based extension) it is most likely that the super-learner will do a better job to fit $\tilde{P}_n^*$ under sampling from $\tilde{P}_n^*$ than that it estimates P_0 under sampling from P_0. So, we expect that $R_\psi(P_{n,\#}^*,\tilde{P}_n^*)$ be be $o_P(1/\sqrt{n})$ since we assumed $R_\psi(\tilde{P}_n^*,P_0)=o_P(1/\sqrt{n})$.

Suppose $D^*(P_{n,\#}^*)-D^*(\tilde{P}_n^*)$ falls in a uniform Donsker class with probability tending to 1, given $(P_n:n)$, where the uniformity of the Donsker class is over $\mathcal{M}$. If we know that $D^*(P_n^*)$ falls in a uniform Donsker class such as the class of functions with multivariate sectional variation norm bounded by a $M<\infty$, then we expect this to also hold for $D^(P_{n,\#}^*)$ if $P_{n,\#}^*$ is as smooth as P_n^*. So this condition would be problematic if $P_{n,\#}^*$ over fits the bootstrap sample relative to the TMLE P_n^* fitting the original sample. Again, due to our uniform oracle inequality of the cross-validation selector established in the next section, one does not expect this to happen at all.

In addition, assume $\tilde{P}_n^*\{D^*(P_{n,\#}^*) - D^*(\tilde{P}_n^*)\}^2 \to 0$ in probability, given almost $(P_n : n)$. This is a weak condition relative to the second order condition above. Then, by empirical process theory (check, tightness of the process uniformly in P should give this, and that is what a uniform Donsker class gives us), we have

$$\sqrt{n}(P_{n,\#} - \tilde{P}_n^*)(D^*(P_{n,\#}^*) - D^*(\tilde{P}_n^*)) \to 0,$$

in probability as $n \to \infty$, given almost every $(P_n : n)$. So, then, given almost every $(P_n : n)$,

$$\Psi(P_{n,\#}^*) - \Psi(\tilde{P}_n^*) = (P_{n,\#} - \tilde{P}_n^*)D^*(\tilde{P}_n^*) + o_P(1/\sqrt{n}).$$

By the uniform standard CLT for sums of real valued independent identically distributed random variables, we have that, if $\tilde{P}_n^*\{D^*(\tilde{P}_n^*)\}^2 \to P_0\{D^*(P_0)\}^2$, then, given $(P_n : n)$, $\sqrt{n}(P_{n,\#} - \tilde{P}_n^*)D^*(\tilde{P}_n^*)$ converges to $N(0, \Sigma^2(P_0))$. So this condition relies on $\Sigma^2(\tilde{P}_n^*)$ to be consistent for $\Sigma^2(P_0)$. In our case, we have that $\Sigma^2(\tilde{P}_n^*)$ is even an efficient estimator of $\Sigma^2(P_0)$ so that this condition will hold very nicely. Thus, this proves consistency of the bootstrap for almost all $(P_n : n)$, under the above stated conditions.

To summarize, one assumption is that our TML estimator $\hat{P}^* : \mathcal{M}_{np} \to \mathcal{M}$ (targeting Ψ) under sampling from $\tilde{P}_n^*$ still satisfies that $\sqrt{n}R_\psi(\hat{P}^*(P_{n,\#}), \tilde{P}_n^*)$ converges to zero, which means that our estimator $\hat{P}^*(P_{n,\#})$ needs to approximate the true $\tilde{P}_n^*$ under sampling from $\tilde{P}_n^*$ at the same or good enough rate as $\hat{P}^*(P_n)$ converges to P_0. This is supported by our uniform oracle inequality for the cross-validation selector in the next section, and the fact that $\tilde{P}_n^*$ will be much easier to estimate than it is to estimate the true P_0: e..g, if $\tilde{P}_n^* = P_n^*$ (i.e., we use the TMLE targeting both ψ_0, σ_0^2), then $\tilde{P}_n^*$ is itself a realization of $\hat{P}^*$ so that the bias $\hat{P}^*(P_{n,\#}) - P_n^*$ on bootstrap samples can be expected to be significantly less than the bias $\hat{P}^*(P_n) - P_0$ based on the original sample.

Once $P_{n,\#}^*$ behaves that way, one also expects that the Donsker class property for $D^*(P_{n,\#}^*) - D^*(\tilde{P}_n^*)$ holds as well (i.e., we do not want that $P_{n,\#}^*$ is an over fit of $\tilde{P}_n^*$ relative to $\tilde{P}_n^*$ as an estimator of P_0). Therefore, given the regularity conditions needed for the efficiency of the TMLE of ψ_0 under sampling from P_0, the only real assumption on $\tilde{P}_n^*$ is that $\Sigma^2(\tilde{P}_n^*))$ is consistent for $\Sigma^2(P_0)$, which is precisely the way $\tilde{P}_n^*$ is targeted by being a TMLE targeting $\Sigma^2(P_0)$.

So we have shown the following theorem stating that the targeted bootstrap is consistent almost everywhere, under appropriate regularity conditions.

Theorem 9. *We make the following assumptions on $\tilde{P}_n^*$: for almost all $(P_n : n)$,*

- *The TMLE $P_{n,\#}^*$ satisfies*

$$P_{n,\#}D^*(P_{n,\#}^*) = o_P(1/\sqrt{n}).$$

- *$\sqrt{n}R_\psi(P^*_{n,\#}, \tilde{P}^*_n) \to 0$ in probability.*
- *$D^*(P^*_{n,\#}) - D^*(\tilde{P}^*_n)$ falls in a uniform Donsker class, where the uniformity is over $\mathcal{M}$.*
- *$\tilde{P}^*_n\{D^*(P^*_{n,\#}) - D^*(\tilde{P}^*_n)\}^2 \to 0$ in probability.*

Then, given almost every $(P_n : n)$,

$$\Psi(P^*_{n,\#}) - \Psi(\tilde{P}^*_n) = (P_{n,\#} - \tilde{P}^*_n)D^*(\tilde{P}^*_n) + o_P(1/\sqrt{n}).$$

If also $\| D^*(\tilde{P}^*_n) \|_\infty < M$ with probability tending to 1, and $\tilde{P}^*_n\{D^*(\tilde{P}^*_n)\}^2 \to P_0\{D^*(P_0)\}^2$ (i.e., $\Sigma^2(\tilde{P}^*_n) \to \Sigma^2(P_0)$ in probability as $n \to \infty$), then, for almost all $(P_n : n)$,

$$\sqrt{n}(P_{n,\#} - \tilde{P}^*_n)D^*(\tilde{P}^*_n) \Rightarrow_d N(0, \Sigma^2(P_0)).$$

Thus, if $\Psi(\tilde{P}^*_n)$ is asymptotically efficient, so that $\sqrt{n}(\Psi(\tilde{P}^*_n) - \Psi(P_0)) \Rightarrow N(0, \Sigma^2(P_0))$, then, for almost all $(P_n : n)$, the bootstrap distribution of $\sqrt{n}(\Psi(P^*_{n,\#}) - \Psi(\tilde{P}^*_n))$ consistently estimates the limit distribution of $\sqrt{n}(\Psi(\tilde{P}^*_n) - \Psi(P_0))$.

Application to the treatment specific mean

Considering the assumptions of theorem 9 in the specific case of the treatment specific mean, we have the following:

Lemma 2. *Sufficient conditions for the last three assumptions of theorem 9are that $\| \hat{\bar{g}}(P^*_{n,\#}) - \bar{g}(\tilde{P}^*_n) \|_{\tilde{P}^*_n} = o_P(n^{-1/4})$, $\| \hat{\bar{Q}}(P^*_{n,\#}) - \hat{\bar{Q}}(\tilde{P}^*_n) \|_{\tilde{P}^*_n} = o_P(n^{-1/4})$, $\bar{g}_0 > \delta > 0$ a.e., $Pr(\| (\hat{\bar{g}}(P^*_{n,\#})\hat{\bar{Q}}(P^*_{n,\#})) \|^*_v < M) \to 1$ for some $M < \infty$.*

If also $\| D^*(\tilde{P}^*_n) \|_\infty < M$ with probability tending to 1, and $\tilde{P}^*_n\{D^*(\tilde{P}^*_n)\}^2 \to P_0\{D^*(P_0)\}^2$, then, for almost all $(P_n : n)$,

$$\sqrt{n}(P_{n,\#} - \tilde{P}^*_n)D^*(\tilde{P}^*_n) \Rightarrow_d N(0, \Sigma^2(P_0)).$$

Sufficient conditions for the latter assumptions are $\bar{g}_0 > \delta > 0$, $\| (\bar{g}, \bar{Q})(\tilde{P}^*_n) - (\bar{g}_0, \bar{Q}_0) \|_{P_0} \to 0$ in probability.

Thus, if $\Psi(P^*_n)$ is asymptotically efficient, so that $\sqrt{n}(\Psi(P^*_n) - \Psi(P_0)) \Rightarrow N(0, \Sigma^2(P_0))$, then, for almost all $(P_n : n)$, the bootstrap distribution of $\sqrt{n}(\Psi(P^*_{n,\#}) - \Psi(\tilde{P}^*_n))$ consistently estimates the limit distribution of $\sqrt{n}(\Psi(P^*_n) - \Psi(P_0))$. Sufficient conditions for the asymptotic efficiency of $\Psi(P^*_n)$ are $\| \bar{g}^*_n - \bar{g}_0 \|_{P_0} = o_P(n^{-1/4})$, $\| \bar{Q}^*_n - \bar{Q}_0 \|_{P_0} = o_P(n^{-1/4})$, $\bar{g}_0 > \delta > 0$, $Pr(\| \bar{g}^*_n, \bar{Q}^*_n \|^*_v < M) \to 1$ for some $M < \infty$.

Thus a corollary of this Theorem is the following.

Corollary 1. *Consider the distribution F_n of $\sqrt{n}(\Psi(P^*_n) - \Psi(P_0)$ and $F^{\#}_n$ of $\sqrt{n}(\Psi(P^*_{n,\#}) - \Psi(\tilde{P}^*_n))$ under sampling from $\tilde{P}^*_n$. Assume*

- $\parallel (\bar{g}, \bar{Q})(\tilde{P}_n^*)) - (\bar{g}_0, \bar{Q}_0) \parallel_{P_0} \rightarrow 0$ *in probability.*
- $\parallel \bar{g}_n^* - \bar{g}_0 \parallel_{P_0} = o_P(n^{-1/4})$*,* $\parallel \bar{Q}_n^* - \bar{Q}_0 \parallel_{P_0} = o_P(n^{-1/4})$*,* $\bar{g}_0 > \delta > 0$*,* $Pr(\parallel \bar{g}_n^*, \bar{Q}_n^* \parallel_v^* < M) \rightarrow 1$ *for some* $M < \infty$*.*
- $\parallel \hat{\bar{g}}(P_{n,\#}^*) - \bar{g}(\tilde{P}_n^*) \parallel_{\tilde{P}_n^*} = o_P(n^{-1/4})$*,* $\parallel \hat{\bar{Q}}(P_{n,\#}^*) - \hat{\bar{Q}}(\tilde{P}_n^*) \parallel_{\tilde{P}_n^*} = o_P(n^{-1/4})$*,* $Pr(\parallel (\hat{\bar{g}}(P_{n,\#}^*)\hat{\bar{Q}}(P_{n,\#}^*)) \parallel_v^* < M) \rightarrow 1$ *for some* $M < \infty$*.*

Then, F_n *converges weakly to* $F_0 = N(0, P_0\{D^*(P_0)\}^2)$*, and* $F_n^{\#}$ *converges to* F_0*, given almost every* $(P_n : n)$*. That is,* ψ_n^* *is asymptotically efficient and the* $\tilde{P}_n^*$*-targeted bootstrap is consistent almost everywhere.*

These conditions on $\hat{\bar{g}}$ and $\hat{\bar{Q}}$ on bootstrap samples from $\tilde{P}_n^*$ w.r.t. estimating $\bar{g}(\tilde{P}_n^*), \bar{Q}(\tilde{P}_n^*)$, relative to how they estimate $\bar{g}_0, \bar{Q}_0$ based on samples from P_0, warrant some discussion. The uniform oracle inequality (theorem 8) shows that if these estimators $\hat{\bar{g}}, \hat{\bar{Q}}$ utilize cross-validation (e.g., they are super-learners based on a specified library of candidate algorithms), then the cross-validation selector will still perform asymptotically in an optimal way. So any potential deterioration of these estimators on bootstrap samples must be due to $\tilde{P}_n^*$ being harder to approximate than P_0. Since $\tilde{P}_n^*$ it itself based on using the same super-learners (i.e, the super-learners are cheating in some sense relative to the super-learner of $(\bar{Q}_0, \bar{g}_0)$), we strongly suggest that in practice this might not be an issue at all.

On the other hand, this might not generally hold up. For example, if the estimator $\hat{\bar{g}}$ is a kernel regression estimator using a kernel that is orthogonal to polynomial powers up till some degree and bandwidth selected based on cross-validation, then the asymptotic rate of convergence of this estimator depends on the underlying smoothness of P_0. If now $\tilde{P}_n^*$ (which itself is close to a kernel regression estimator) is now less smooth than P_0 (at least for n large enough), then one expects that the kernel regression estimator based on the bootstrap samples will converge to their true counterparts at a slower rate. So in this case, it appears that some over smoothing of $\tilde{P}_n^*$ might be required (e.g. use initial estimators with a bandwidth larger than the one selected by cross-validation), which is consistent with results on bootstrapping in the nonparametric regression and density estimation literature. However, in most real data applications our covariates are high dimensional and we rely on using many candidate algorithms using different approximation strategies (often relying on extrapolation). Therefore, we suggest that in practice the advantages of $\tilde{P}_n^*$ to be of the same form as an actual realization of these super-learners outweighs the concerns for $\tilde{P}_n^*$ being less smooth than P_0. Nonetheless, it is crucial that if the actual model $\mathcal{M}$ makes certain smoothness assumptions, that these are enforced in $\tilde{P}_n^*$; i.e. $\tilde{P}_n^* \in \mathcal{M}$ with probability 1. It is also good to realize that the bootstrapped estimators are allowed to lose some performance relative to the original estimators as long as they still achieve the desired rates of convergence.

Appendix B

Dependent Cross-Validation Theory

This appendix contains details of the theoretical results for dependent cross-validation[1].

B.1 Cross-validation oracle inequality

Objective

Let Γ be some sub-collection of all functions that map from $\mathcal{O}$ to $\mathbb{R}$. For each $\gamma \in \Gamma$, let L^γ be some nonnegative loss function for estimating

$$\theta_0 \equiv \arg\min \theta \in \Theta E_P[L^{\gamma(P)}(O, \theta)],$$

where we will abuse notation and let $\gamma(P)$ represent both a parameter mapping on the distribution P and let a value γ represent a corresponding value of that parameter mapping, e.g. $\gamma(P)$ may be $Pr_P(A = 1|W)$. We will let $\gamma_0 = \gamma(P_0)$.

Suppose a cross-validation scheme B_n as discussed in Section 2.1 of van der Laan and Dudoit [2003a], and let $P^0_{B_n}$ and $P^1_{B_n}$ respectively represent the training and validation samples corresponding to the split B_n. Let n^1 represent a lower bound on the size of the training set resulting from the cross-validation split, e.g. in V-fold cross-validation $n^1 = \lfloor n/V \rfloor$.

Throughout we will assume that, for a given split B_n, γ_0 is estimated using only $P^0_{B_n}$, save for a parameter β_n that is selected from a set of size $N_\beta(n) < \infty$ values. We will denote this estimate $\hat{\gamma}^{\beta_n}_{B_n}$. For example, β_n could be the convex combination derived by running a super-learner algorithm to learn γ_0, but then using the candidate estimators only applied to the training set associated with B_n.

Also suppose that we aim to learn θ_0 using a super-learner with convex combination α_n that can take on $N_\alpha(n)$ values. For a given α and training sample $P^0_{B_n}$, denote the convex combination as $\hat{\theta}^\alpha_{B_n}$, where the candidates are learned on P and combined using the weights α. By restricting α_n to take finitely many values on the simplex, we can in fact cover

[1] All results in this section due to Alex Luedtke

many algorithms besides what we typically think of as super-learners. More succinctly, these algorithms can be viewed as a special case of the super-learner algorithm. For example, for bandwidth selection in a kernel density estimator, we could take the candidates to be the density estimators at different bandwidths, and the choices of α to be the "corners" of the simplex, i.e. the points that contain a one in exactly one coordinate and zeroes in all others.

We define the cross-validated risk at the true γ_0 as follows:

$$R_n(\alpha) \equiv E_{B_n} E_P \left[L^{\gamma_0}(O, \hat{\theta}^{\alpha}_{B_n}) \right].$$

We may want to consider the cross-validated risk under misspecification of the nuisance parameter, given by

$$R^{\beta}_n(\alpha) \equiv E_{B_n} E_P \left[L^{\hat{\gamma}^{\beta}_{B_n}}(O, \hat{\theta}^{\alpha}_{B_n}) \right].$$

Oracle does not know γ_0

We first consider an oracle who does not know the true nuisance function value γ_0. This oracle then seeks to minimize the cross-validated risk that treats the estimated nuisance function as the true nuisance function and aims to choose the convex combination α to minimize the risk with this estimated nuisance function treated as the truth. For each sample of size n, the oracle will treat a different estimated nuisance function as the true nuisance function, and thus aim to minimize an objective function with different nuisance parameters.

Theorem 10. *For each $\theta \in \Theta$ and $\gamma \in \Gamma$, suppose $o \mapsto L^{\gamma}(o, \theta)$ has Bernstein pair $(M(\theta, \gamma), v(\theta, \gamma))$. Further suppose that $R^{\gamma}(\theta) \equiv E_P L^{\gamma}(O, \theta) \geq 0$ for every $\theta \in \Theta, \gamma \in \Gamma$. Fix $\delta > 0$ and $1 \leq p \leq 2$. Then we have the following finite sample oracle ineqality:*

$$E_{P^n} R^{\beta_n}_n(\alpha_n) \leq (1 + 2\delta) E_{P^n} \min_{\alpha} R^{\beta_n}_n(\alpha) + Rem_n(p, \delta)$$

where P^n represents the distribution of an i.i.d. sample of size n from P and

$$\begin{aligned} Rem_n(p, \delta) \equiv & (1 + \delta) \frac{16 \log\left(1 + N_\alpha(n) + N_\beta(n)\right)}{(n^1)^{1/p}} \times \\ & \sup_{\theta \in \Theta, \gamma \in \Gamma} \left[\frac{M(\theta, \gamma)}{(n^1)^{1 - 1/p}} + \left(\frac{v(\theta, \gamma)}{R^{\gamma}(\theta)^{2-p}} \right)^{1/p} \left(\frac{1 + \delta}{\delta} \right)^{2/p - 1} \right]. \end{aligned}$$

Proof. For a given sample split B_n, let $\mathbb{G}^1_{B_n} = (P^1_{B_n} - P)$ represent the empirical process of the training sample. By the same calculations used to prove Lemma 2.1 in van der Vaart et al. [2006b], we have that for any α_1 in the set of size K_α:

$$\begin{aligned} R_n^{\beta_n}(\alpha_n) \leq & (1+2\delta) R_n^{\beta_n}(\alpha_1) \\ & + \frac{1}{\sqrt{n^1}} E_{B_n}\left[\max_{\alpha,\beta} \int L^{\hat{\gamma}^{\beta}_{B_n}}(o, \hat{\theta}^{\alpha}_{B_n}) d\left((1+\delta)\mathbb{G}^1_{B_n} - \delta\sqrt{n^1}P\right)(o)\right] \\ & + \frac{1}{\sqrt{n^1}} E_{B_n}\left[\max_{\alpha,\beta} \int -L^{\hat{\gamma}^{\beta}_{B_n}}(o, \hat{\theta}^{\alpha}_{B_n}) d\left((1+\delta)\mathbb{G}^1_{B_n} + \delta\sqrt{n^1}P\right)(o)\right] \\ \equiv & (1+2\delta) R_n^{\beta_n}(\alpha_1) + r(P_n, \delta). \end{aligned} \tag{B.1}$$

Because the above holds for any α_1, it holds for the minimum over all α in the set of size K_α. The remainder term $r(P_n, \delta)$ is the sum of two decentered empirical processes whose expectations under P^n, scaled by root-n^1 and averaged over B_n. For a given B_n, the decentered empirical processes can be controlled using Lemma 2.2 of van der Vaart et al. [2006b]. In particular, $E_{P^n} r(P_n, \delta) \leq Rem_n(p, \delta)$ for any $1 \leq p \leq 2$. Taking the expected value of the above with respect to P^n completes the proof. □

Suppose $\{L^\gamma : \gamma \in \Gamma\}$ satisfies the following conditions:

$$\sup_{\theta,\gamma} \sup_{o \in \mathcal{O}} |L^\gamma(o,\theta)| < \infty \tag{B.2}$$

$$\sup_{\theta,\gamma} \frac{Var_P\left(L^\gamma(O,\theta)\right)}{E_P\left[L^\gamma(O,\theta)\right]} < \infty. \tag{B.3}$$

In this case, we can derive the following oracle inequality:

Corollary 2. *Suppose $\{L^\gamma : \gamma \in \Gamma\}$ satisfies (B.2) and (B.3). Fix $\delta > 0$. Then we have the following finite sample oracle ineqality:*

$$E_{P^n} R_n^{\beta_n}(\alpha_n) \leq (1+2\delta) E_{P^n} \min_{\alpha} R_n^{\beta_n}(\alpha) + C(\delta) \frac{\log\left(1 + N_\alpha(n) + N_\beta(n)\right)}{n^1},$$

where P^n represents the distribution of an i.i.d. sample of size n from P and $C(\delta)$ is a constant that may rely on P and the loss L.

Proof. This result is an immediate consequence of Theorem 10, taking $p = 1$ and setting $M(\theta,\gamma) = \sup_{\theta,\gamma} \sup_{o \in \mathcal{O}} |L^\gamma(o,\theta)|$ and $v(\theta,\gamma) = 1.5 E_P\left[L^\gamma(O,\theta)^2\right]$ (see Section 8.1 of van der Vaart et al. [2006b]). □

We may instead be interested in whether or not the risk of the estimated α converges to the risk of the oracle in probability. While we could directly apply Markov's inequality to establish such a result, we will instead use the bounds on the Orlicz norm established previously in the literature to get a finite sample bound on this probability. In this way, we can say that, with probability at least $1 - K/n$, the risk of our estimator deviates from $(1+2\delta)$ times the risk of our oracle by a term that decays to 0 as $\log(1 + N_\alpha + N_\beta)\log n/n$. This result is reminiscent of the finite sample bounds given in the machine learning literature.

Theorem 11. *Suppose (B.2) and (B.3) hold. Fix $\delta > 0$ and $k > 0$. Then, for a constant C which may rely on P, the loss L, and δ, the following holds with probability at least $1 - 2/(n^k - 1)$:*

$$R_n^{\beta_n}(\alpha_n) - (1+2\delta)\min_{\alpha} R_n^{\beta_n}(\alpha) \leq \frac{Ck\log(1+N_\alpha+N_\beta)\log n}{n^1}.$$

If $n^1 = \Theta(n)$ and $N_\alpha + N_\beta = O(n^d)$ for $d > 0$, then again with probability at least $1-2/(n^k - 1)$:

$$R_n^{\beta_n}(\alpha_n) - (1+2\delta)\min_{\alpha} R_n^{\beta_n}(\alpha) \leq O\left(\frac{dk\log^2 n}{n}\right).$$

Proof. Note: to avoid adding extra notation, the constant C changes throughout this proof.

We will use the inequality in (B.1) with α_1 replaced by the minimum over all α in the set of size K_α. Consider the quantity:

$$E_{B_n}\left[r_1(P_{B_n}^0, P_{B_n}^1)\right] = E_{B_n}\left[\max_{\alpha,\beta}\int L^{\hat{\gamma}_{B_n}^{\beta}}(o, \hat{\theta}_{B_n}^{\alpha}) d\left(\mathbb{G}_{B_n}^1 - \frac{\delta}{1+\delta}\sqrt{n^1}P\right)(o)\right].$$

Let $\psi(x) \equiv \exp(x^p) - 1$. The Orlicz norm $\|X\|_\psi$ of a random variable $X \sim P_X$ is given by

$$\|X\|_\psi \equiv \inf\left\{C > 0 : E_{P_X}\psi\left(\frac{|X|}{C}\right) \leq 1\right\}.$$

We have:

$$\left\|E_{B_n}\left[r_1(P_{B_n}^0, P_{B_n}^1)\right]\right\|_\psi \leq E_{B_n}\left\|r_1(P_{B_n}^0, P_{B_n}^1)\right\|_\psi.$$

For $C > 0$ and a given B_n

$$E_{P^n}\psi\left(\frac{r_1(P_{B_n}^0, P_{B_n}^1)}{C}\right) = E_{P_{B_n}^0} E_{P_{B_n}^1}\left[\psi\left(\frac{r_1(P_{B_n}^0, P_{B_n}^1)}{C}\right)\right].$$

It follows that, for a given B_n and $P_{B_n}^0$, an upper bound u on $\left\|r_1(P_{B_n}^0, P_{B_n}^1)\right\|_\psi$ for fixed B_n, $P_{B_n}^0$ that does not rely on B_n, $P_{B_n}^0$ then the same upper bound u is valid for $a \equiv \left\|E_{B_n}\left[r_1(P_{B_n}^0, P_{B_n}^1)\right]\right\|_\psi$. For $t > 0$, Markov's inequality gives:

$$\begin{aligned} Pr\left(E_{B_n}\left[r_1(P_{B_n}^0, P_{B_n}^1)\right] > t\right) &\leq Pr\left(\psi\left(\frac{E_{B_n}\left[r_1(P_{B_n}^0, P_{B_n}^1)\right]}{a}\right) > \psi\left(\frac{t}{a}\right)\right) \\ &\leq \frac{1}{\psi(t/a)} \leq \frac{1}{\psi(t/u)} = \frac{1}{\exp(t/u) - 1}, \end{aligned} \tag{B.4}$$

so that we can establish exponential tail bounds for $E_{B_n}\left[r_1(P_{B_n}^0, P_{B_n}^1)\right]$.

Fix B_n and $P^0_{B_n}$ so that $P^1_{B_n}$ can be treated as an i.i.d. sample of size n^1 from P. Define:

$$\begin{aligned} s(\alpha,\beta) &\equiv \int L^{\hat{\gamma}^{\beta}_{B_n}}(o,\hat{\theta}^{\alpha}_{B_n})d\left(\mathbb{G}^1_{B_n} - \frac{\delta}{1+\delta}\sqrt{n^1}P\right)(o) \\ R(\alpha,\beta) &\equiv E_P[L^{\hat{\gamma}^{\beta}_{B_n}}(O,\hat{\theta}^{\alpha}_{B_n})], \end{aligned}$$

where the dependence on $P^1_{B_n}$ (only first definition), $P^0_{B_n}$, and B_n are omitted in the notation.

The proof of Lemma 8.2 in van der Vaart et al. [2006b] established that, for each α and β:

$$\begin{aligned} Pr\left(sI\left(s \le \sqrt{n^1}\left(\frac{v}{M}+\frac{\delta R}{1+\delta}\right)\right) > t\right) &\le \exp\left(-\frac{t\frac{\delta R}{1+\delta}\sqrt{n^1}+\left(\frac{\delta R}{1+\delta}\right)^2 n^1}{4v}\right) \\ Pr\left(sI\left(s > \sqrt{n^1}\left(\frac{v}{M}+\frac{\delta R}{1+\delta}\right)\right) > t\right) &\le \exp\left(-\sqrt{n^1}\frac{t+\frac{\delta R}{1+\delta}\sqrt{n^1}}{4M}\right), \end{aligned}$$

where we omit the dependence of s, R on (α,β) and v, M on $(\hat{\theta}^{\alpha}_{B_n},\hat{\gamma}^{\beta}_{B_n})$. Compared to the notation of van der Vaart et al., we have replaced $\lambda(f)$ with $\frac{\delta R}{1+\delta}\sqrt{n^1}$ and p,q with 1. By Lemma 2.2.1 in van der Vaart and Wellner [1996]:

$$\begin{aligned} \left\|sI\left(s \le \sqrt{n^1}\left(\frac{v}{M}+\frac{\delta R}{1+\delta}\right)\right)\right\|_{\psi} &\le \frac{4v}{R\sqrt{n^1}}\left(\frac{1+\delta}{\delta}\right)\left(1+e^{-\left(\frac{\delta R}{1+\delta}\right)^2\frac{n^1}{4v}}\right) \\ \left\|sI\left(s > \sqrt{n^1}\left(\frac{v}{M}+\frac{\delta R}{1+\delta}\right)\right)\right\|_{\psi} &\le \frac{4M}{\sqrt{n^1}}\left(1+e^{-\frac{\delta R}{1+\delta}\frac{n^1}{4M}}\right) \end{aligned}$$

where the norms are taken over the distribution of $P^1_{B_n}$. The convexity of $\|\cdot\|_{\psi}$ and the fact that $e^{-x} \le 1$ for $x \ge 0$ shows that

$$\begin{aligned} \|s(\alpha,\beta)\|_{\psi} &\le \frac{4M}{\sqrt{n^1}}\left(1+e^{-\frac{\delta R}{1+\delta}\frac{n^1}{4M}}\right)+\frac{4v}{R\sqrt{n^1}}\left(\frac{1+\delta}{\delta}\right)\left(1+e^{-\left(\frac{\delta R}{1+\delta}\right)^2\frac{n^1}{4v}}\right) \\ &\le \frac{8M}{\sqrt{n^1}}+\frac{8v}{R\sqrt{n^1}}\left(\frac{1+\delta}{\delta}\right). \end{aligned}$$

We can take the maximum over α,β and apply Lemma 2.2.2 of van der Vaart and Wellner [1996] to show that:

$$\begin{aligned} \left\|\max_{\alpha,\beta} s(\alpha,\beta)\right\|_{\psi} &\lesssim \log\left(1+N_\alpha+N_\beta\right)\max_{\alpha,\beta}\|s(\alpha,\beta)\|_{\psi} \\ &\lesssim \frac{1}{\sqrt{n^1}}\log\left(1+N_\alpha+N_\beta\right)\sup_{\theta\in\Theta,\gamma\in\Gamma}\left(M(\theta,\gamma)+\frac{v(\theta,\gamma)}{E_P L^{\gamma}(O,\theta)}\left(\frac{1+\delta}{\delta}\right)\right), \end{aligned}$$

where $\lesssim$ denotes less than or equal to up to a universal constant and we have used the fact that our estimators of θ_0 and γ_0 fall in Θ and Γ. Noting that $\max_{\alpha,\beta} s(\alpha,\beta) = r_1(P^0_{B_n}, P^1_{B_n})$

for our fixed $P^0_{B_n}$ and that the upper bound established above does not rely on B_n or $P^0_{B_n}$, we can return to (B.4) to show that

$$Pr\left(E_{B_n}\left[r_1(P^0_{B_n}, P^1_{B_n})\right] > t\right) \leq \left(\exp\left(Ct\frac{\sqrt{n^1}}{\log(1+N_\alpha+N_\beta)}\right)-1\right)^{-1},$$

where C now relies on the distibution P, the loss L, and the choice of $\delta > 0$. Thus

$$Pr\left(\frac{1}{\sqrt{n^1}}E_{B_n}\left[r_1(P^0_{B_n}, P^1_{B_n})\right] > t\right) \leq \left(\exp\left(Ct\frac{n^1}{\log(1+N_\alpha+N_\beta)}\right)-1\right)^{-1}.$$

Taking $t = k\log(1+N_\alpha+N_\beta)\log n/(Cn^1)$ shows that, with probability at most $1/(n^k-1)$:

$$\frac{1}{\sqrt{n^1}}E_{B_n}\left[r_1(P^0_{B_n}, P^1_{B_n})\right] > \frac{Ck\log(1+N_\alpha+N_\beta)\log n}{n^1}.$$

The same arguments give the same type of result for the second term in (B.1). The claim follows by a union bound. □

Oracle knows γ_0

Given a sample P_n, let the oracle be the selector which chooses α^*_n to minimize the cross-validated risk resulting from the actual loss of interest L^{γ_0}. That is:

$$\alpha^*_n = \arg\min \alpha R_n(\alpha).$$

We have the following oracle inequality:

Lemma 3. *Suppose the conditions of Theorem 10. Then:*

$$\begin{aligned} E_{P^n}R_n(\alpha_n) \leq& (1+2\delta)E_{P^n}R_n(\alpha^*_n) + 2\delta E_{P^n}\left[R^{\beta_n}_n(\alpha^*_n) - R_n(\alpha^*_n)\right] \\ &+ E_{P^n}\left[\left(R^{\beta_n}_n(\alpha^*_n) - R_n(\alpha^*_n)\right) - \left(R^{\beta_n}_n(\alpha_n) - R_n(\alpha_n)\right)\right] \\ &+ Rem_n(p,\delta), \end{aligned}$$

where $Rem_n(p,\delta)$ is defined in Theorem 10.

Proof. We have that:

$$\begin{aligned} E_{P^n}R_n(\alpha_n) &= E_{P^n}R^{\beta_n}_n(\alpha_n) + E_{P^n}\left[R_n(\alpha_n) - E_{P^n}R^{\beta_n}_n(\alpha_n)\right] \\ &\leq (1+2\delta)E_{P^n}R^{\beta_n}_n(\alpha^*_n) + E_{P^n}\left[R_n(\alpha_n) - E_{P^n}R^{\beta_n}_n(\alpha_n)\right] + Rem_n(p,\delta), \end{aligned}$$

where the inequality is an application of (B.1) at $\alpha_1 = \alpha^*_n$. The result follows by adding and subtracting $(1+2\delta)E_{P^n}R_n(\alpha^*_n)$ from the right-hand side. □

This leads to the following theorem, which gives conditions under which the oracle inequality holds up to the rate at which the nuisance function is estimated correctly.

Theorem 12. *Suppose the conditions of Theorem 10. Further suppose that*

$$E_{P^n}\left[\left(R_n^{\beta_n}(\alpha_n^*) - R_n(\alpha_n^*)\right) - \left(R_n^{\beta_n}(\alpha_n) - R_n(\alpha_n)\right)\right] = o\left(E_{P^n}\left[R_n^{\beta_n}(\alpha_n^*) - R_n(\alpha_n^*)\right]\right).$$

Then we have that:

$$\begin{aligned} E_{P^n}R_n(\alpha_n) \leq& (1+2\delta)E_{P^n}R_n(\alpha_n^*) + O\left(\delta E_{P^n}\left[R_n^{\beta_n}(\alpha_n^*) - R_n(\alpha_n^*)\right]\right) \\ &+ o\left(E_{P^n}\left[R_n^{\beta_n}(\alpha_n^*) - R_n(\alpha_n^*)\right]\right) + Rem_n(p,\delta). \end{aligned}$$

Proof. The result is an immediate consequence of the stated condition and the preceding lemma. □

It is now worth considering the plausibility of the stated condition in practical situations. For simplicity, suppose $\gamma_0 = \beta_0$ is a univariate nuisance parameter, and we are interested in learning $\alpha_0 = \theta_0 \in \mathbb{R}^1$. Note in this case there are no candidate estimators so cross-validation is not meaningful (we're just estimating α_0 and β_0 on the full sample). We would like to evaluate the performance of our estimate α_n with the risk function $\alpha \mapsto R^{\beta_0}(\alpha)$, but $\beta_0 \in \mathbb{R}^1$ is unknown so we use an estimate β_n of this nuisance parameter. Under mild regularity conditions, we have the following Taylor series expansion about (α_0, β_0):

$$\begin{aligned} &E_{P^n}\left[\left(R^{\beta_n}(\alpha_n) - R^{\beta_0}(\alpha_n)\right) - \left(R^{\beta_n}(\alpha_0) - R^{\beta_0}(\alpha_0)\right)\right] \\ &=E_{P^n}\left[(\alpha_n - \alpha_0)(\beta_n - \beta_0)\right] \frac{\partial}{\partial\alpha}\frac{\partial}{\partial\beta}R^\beta(\alpha)\bigg|_{\alpha=\alpha_0,\beta=\beta_0} \\ &\quad + o\left(E_{P^n}\left[(\alpha_n - \alpha_0)(\beta_n - \beta_0)\right]\right) \end{aligned}$$

Another Taylor series expansion shows that

$$E_{P^n}[R^{\beta_n}(\alpha_0) - R^{\beta_0}(\alpha_0)] = O(E_{P^n}[\beta_n - \beta_0]).$$

B.2 Cross-validated one-step estimator and cross-validated TMLE

Problem setup

Suppose we wish to estimate a univariate parameter Ψ evaluated at a distribution P, where Ψ takes as input a distribution in our model $\mathcal{M}$ and outputs a real number. All of the results in this section can be generalized to a finite dimensional multivariate parameter.

Let $\{P_t^h : t \in (-\delta, \delta),\ \text{some } \delta > 0\}$ be a parametric submodel of $\mathcal{M}$ such that $P_{t=0}^h = P$ and P_t^h has square integrable score $o \mapsto h(o)$ at $t = 0$. We suppose that there exists some mapping $D : \mathcal{M} \to \mathbb{R}$ which does not depend on h such that

$$\Psi(P_t^h) - \Psi(P) = \int D(P)(o)h(o)dP(o) + o(t)$$

for all such submodels and $D(P)(O)$ is mean zero under P. If this property holds then Ψ is called pathwise differentiable at P in the model $\mathcal{M}$. The function $D(P)$ is known as a gradient of Ψ at P in the model $\mathcal{M}$. In a general model $\mathcal{M}$ gradients are not necessarily unique, but the minimum variance gradient $D^*(P)$, known as the canonical gradient, is (P almost surely) unique. For simplicity we assume that Ψ is pathwise differentiable at all $P' \in \mathcal{M}$ in this section, with canonical gradient $D^*(P')$. We refer the reader to Bickel et al. [1993] for a detailed exposition of pathwise differentiability.

For some distribution $P' \in \mathcal{M}$, we can always write

$$\Psi(P') - \Psi(P) = -PD^*(P') + Rem(P, P') \tag{B.5}$$

for some $Rem(P, P')$, where we have used the notation $Pf = E_P[f(O)]$ for a function f. It follows from the pathwise differentiability of Ψ at P that $Rem(P, P')$ is small whenever P is close to P' in the right sense

The exposition of the cross-validated estimators in this section is similar to that in Zheng and van der Laan [2011]. Our results differ in that the estimate of the distribution P, used to estimate the parameter $\Psi(P)$ and the canonical gradient $D^*(P)$, does not have to rely only on the training sets in our setting. Our main motivation for this generalization is to allow one to use the entire data set to select between different candidate estimators when estimating the components of P needed to evaluate Ψ and D^*.

One-step estimator

The challenge of a non-cross-validated one-step estimator

First suppose one wishes to estimate Ψ using a one-step estimator. That is, one obtains an estimate $P_n^0 \in \mathcal{M}$ of P, or at least an estimate of the components of P needed to evaluate Ψ and D^*, and returns as estimate $\psi_n^{os} \equiv \Psi(P_n^0) + P_n D^*(P_n^0)$. Using (B.5), we have that

$$\begin{aligned} \psi_n^{os} - \Psi(P) &= (P_n - P)D^*(P_n^0) + Rem(P, P_n^0) \\ &= (P_n - P)D^*(P) + (P_n - P)\left[D^*(P_n^0) - D^*(P)\right] + Rem(P, P_n^0). \end{aligned} \tag{B.6}$$

Suppose that P_n^0 is a good estimate of P in the sense that $Rem(P, P_n^0) = o_P(n^{-1/2})$, i.e. converges to 0 in probability faster than $1/\sqrt{n}$. If $D^*(P_n^0)$ belongs to a Donsker class and $P(D^*(P_n^0) - D^*(P))^2$ converges to zero in probability, then Lemma 19.24 of van der Vaart [1998] yields

$$\sqrt{n}\left|(P_n - P)\left[D^*(P_n^0) - D^*(P)\right]\right| \to 0 \text{ in probability.} \tag{B.7}$$

Thus the middle term in (B.6) is $o_P(n^{-1/2})$. In this case $\sqrt{n}[\psi_n^{os} - \Psi(P)] = \sqrt{n}(P_n - P)D^*(P) + o_P(1)$, and the right-hand side converges to a normal limit distribution by the central limit theorem and Slutsky's theorem.

If P_n^0 is too data adaptive, $D^*(P_n^0)$ may not satisfy the Donsker class condition, or the Donsker class may be so large that (B.7) is of little use for reasonable sample sizes.

Cross-validated one-step estimator

For the cross-validated one-step and targeted minimum loss based estimators we focus on V-fold cross-validation. In the notation of Section B.1, this means that B_n takes on V values and that the elements of the training set $P_{B_n}^1$ are unique to each instance of B_n and all observations are contained in exactly one training set. That is, the training sets are mutually exclusive and exhaustive. We also assume that the training sets are of approximately equal size so that $n^1 \approx \lfloor n/V \rfloor$.

For a parameter β belonging to some index set Σ_n, let $\hat{P}_\beta$ be an estimator which takes as input a training sample $P_{B_n}^0$ and outputs an estimate of P which falls in $\mathcal{M}$. Note that we have allowed these index sets to rely on sample size, so that the estimator of P may borrow more information across training samples as the sample size grows. We assume that β_n is some choice of β that may rely on the entire observed sample of size n. It does not matter what criteria was used to select β_n.

Our main motivation in this section is when we have many candidate estimates of P, and the β subscript indexes the choice of candidates of P. This may involve running several super-learners for different condition expectations and conditional probabilities. The β subscript then indexes the discrete approximations to all of these super-learners.

The cross-validated one-step estimator of $\Psi(P)$ is defined as

$$\psi_n^{cvos} \equiv P_{B_n}\left[\Psi(\hat{P}_{\beta_n}(P_{B_n}^0)) + P_{B_n}^1 D^*(\hat{P}_{\beta_n}(P_{B_n}^0))\right].$$

We assume that our estimator of P satisfies the following consistency condition:

$$\max_{b_n} Rem(P, \hat{P}_{\beta_n}(P_{b_n}^0)) = o_P(n^{-1/2}), \tag{B.8}$$

where the maximum is over realizations of B_n. One could in principle replace P with some partially misspecified $P_1 \neq P$ representing the limit of our estimates of P, but we do not do so here.

We assume that, for some $M \in (0, \infty)$,

$$P^n\left(\sup_o \left| D^*(\hat{P}_{\beta_n}(P_{B_n}^0))(o) - D^*(P)(o)\right| < M\right) \to 1. \tag{B.9}$$

For sequence $\{\delta_n\}$ and training sample realization $P_{B_n}^0$, define the random class

$$\mathcal{D}_n(P_{B_n}^0) \equiv \left\{ d = \frac{D^*(\hat{P}_\beta(P_{B_n}^0)) - D^*(P)}{M} : \beta \in \Sigma_n,\ Pd^2 \leq \delta_n^2, \text{ and } \sup_o |d(o)| \leq 1 \right\}.$$

Entropy integral definition and an inequality. Let $\mathcal{D}$ be a class of functions mapping from $\mathcal{O}$ to $[-1,1]$. For a distribution P' and function f, define the $L_2(P')$ seminorm of f as $\|f\|_{P',2} = (P'f^2)^{1/2}$. For a finitely discrete distribution Q, let $N(\epsilon, \mathcal{D}, L_2(Q))$ represent the minimal number of $L_2(Q)$ balls of radius ϵ which cover $\mathcal{D}$. For $\delta > 0$, define the uniform entropy integral as

$$J(\delta, \mathcal{D}) = \sup_Q \int_0^\delta \sqrt{1 + \log N(\epsilon, \mathcal{D}, L_2(Q))} d\epsilon,$$

where the supremum is over finitely discrete distributions. Suppose that $\sup_{d\in\mathcal{D}} Pd^2 \leq \delta^2$ for some $\delta \in (0,1)$. From Theorem 2.1 in Van Der Vaart and Wellner [2011], we know that

$$E_{P^n}\sqrt{n}\sup_{d\in\mathcal{D}} |(P_n - P)d| \lesssim J(\delta, \mathcal{D})\left(1 + \frac{J(\delta,\mathcal{D})}{\delta^2\sqrt{n}}\right). \tag{B.10}$$

□

We now continue with our presentation of the conditions for the cross-validated one-step estimator theorem.

We assume that both of the following conditions hold for some deterministic sequence $\delta_n \to 0$:

$$\max_{b_n} P(D^*(\hat{P}_{\beta_n}(P^0_{b_n})) - D^*(P))^2 = o_P(\delta_n^2) \tag{B.11}$$

$$\max_{b_n} E_{P^n}\left[J(\delta_n, \mathcal{D}_n(P^0_{b_n}))\left(1 + \frac{\sqrt{V}J(\delta_n, \mathcal{D}_n(P^0_{b_n}))}{\delta_n^2\sqrt{n}}\right)\right] \overset{n\to\infty}{\longrightarrow} 0. \tag{B.12}$$

A deterministic sequence satisfying the first condition exists provided $\max_{b_n} P(D^*(\hat{P}_{\beta_n}(P^0_{b_n})) - D^*(P))^2 = o_P(1)$, but it can take some care for this sequence to also satisfy the latter condition if the index class Σ_n grows with n.

We have the following theorem.

Theorem 13. *Suppose that (B.8), (B.9), (B.11), and (B.12) hold. Then,*

$$\sqrt{n}\left[\psi_n^{cvos} - \Psi(P)\right] = \frac{1}{\sqrt{n}}\sum_{i=1}^n D^*(P)(O_i) + o_P(1) \rightsquigarrow \text{Normal}(0, \sigma^2),$$

where σ^2 is the variance of $D^*(P)(O)$.

Proof. First note that

$$\begin{aligned}\psi_n^{cvos} - \Psi(P) =& P_{B_n}\left[\Psi(\hat{P}_{\beta_n}(P^0_{B_n})) - \Psi(P_0) + P^1_{B_n}D^*(\hat{P}_{\beta_n}(P^0_{B_n}))\right] \\ =& P_{B_n}(P^1_{B_n} - P)D^*(P) + P_{B_n}(P^1_{B_n} - P)\left[D^*(\hat{P}_{\beta_n}(P^0_{B_n})) - D^*(P)\right] \\ &+ P_{B_n}Rem(P, \hat{P}_{\beta_n}(P^0_{B_n})).\end{aligned} \tag{B.13}$$

The final term is $o_P(n^{-1/2})$ by assumption. The first term simplifies to $(P_n - P)D^*(P)$ because the training sets are mutually exclusive and exhaustive.

We now consider the middle term. Let A_n represent the event

$$\frac{D^*(\hat{P}_{\beta_n}(P^0_{b_n})) - D^*(P)}{M} \in \mathcal{D}_n(P^0_{b_n}) \text{ for all instances } b_n \text{ of } B_n.$$

By our assumptions, A_n occurs with probability approaching 1. Hence, for any $t > 0$,

$$\begin{aligned}
&P^n\left\{\sqrt{n}\left|P_{B_n}(P^1_{B_n} - P)\left[D^*(\hat{P}_{\beta_n}(P^0_{B_n})) - D^*(P)\right]\right| \geq t\right\} \\
\leq& \sum_{b_n} P^n\left\{\sqrt{n}\left|(P^1_{b_n} - P)\left[D^*(\hat{P}_{\beta_n}(P^0_{b_n})) - D^*(P)\right]\right| \geq t\right\} \\
\leq& \sum_{b_n} P^n\left\{\sqrt{n}\left|(P^1_{b_n} - P)\left[D^*(\hat{P}_{\beta_n}(P^0_{b_n})) - D^*(P)\right]\right| \geq t \text{ and } A_n\right\} + o(1) \\
\leq& \sum_{b_n} P^n P^n\left\{\sqrt{n}\sup_{d\in\mathcal{D}_n(P^0_{b_n})}\left|(P^1_{b_n} - P)d\right| \geq \frac{t}{M}\,\middle|\, P^0_{b_n}\right\} + o(1) \\
\leq& \frac{M\sum_{b_n} P^n E_{P^n}\left[\sqrt{n}\sup_{d\in\mathcal{D}_n(P^0_{b_n})}\left|(P^1_{b_n} - P)d\right|\,\middle|\, P^0_{b_n}\right]}{t} + o(1). \qquad \text{(B.14)}
\end{aligned}$$

The final equality holds by Markov's inequality and the law of total expectation. Conditional on $P^0_{b_n}$, the class $\mathcal{D}_n(P^0_{b_n})$ is fixed and we can apply the results in Section B.2. In particular, we have the maximal inequality

$$E_{P^n}\left[\sqrt{n}\sup_{d\in\mathcal{D}_n(P^0_{b_n})}\left|(P^1_{b_n} - P)d\right|\,\middle|\, P^0_{b_n}\right] \lesssim J(\delta_n, \mathcal{D}_n(P^0_{b_n}))\left(1 + \frac{\sqrt{V}J(\delta_n, \mathcal{D}_n(P^0_{b_n}))}{\delta_n^2\sqrt{n}}\right),$$

where we used that the validation set $P^1_{b_n}$ contains approximately n/V observations. The expected value of this quantity over training samples times V converges to 0 by assumption. Plugging this into (B.14) and noting that $t > 0$ was arbitrary yields that $P_{B_n}(P^1_{B_n} - P)\left[D^*(\hat{P}_{\beta_n}(P^0_{B_n})) - D^*(P)\right] = o_P(n^{-1/2})$. The result follows by (B.13) and the central limit theorem, where we use that $D^*(P)$ is the canonical gradient and therefore mean zero and square integrable. □

We now give sufficient conditions under which (B.12) holds. First we suppose that the entropy of the class is uniformly bounded in the sense that

$$P^n\left\{\max_{b_n}\sup_{Q}\log N(\epsilon, \mathcal{D}_n(P^0_{b_n}), L_2(Q)) \leq G(\epsilon) \text{ for all } \epsilon > 0, n\right\} = 1 \qquad \text{(B.15)}$$

for some G which satisfies $\int_0^1\sqrt{1+G(\epsilon)}d\epsilon < \infty$. The only random quantities in the probability statement above are the training samples $P^0_{b_n}$. We have the following corollary.

Corollary 3. *The conclusion of Theorem 13 follows if the condition (B.12) is replaced by the stronger condition (B.15).*

Proof. Without loss of generality we can take δ_n so that $n^{-1/4} = o(\delta_n)$. There is no loss of generality because (B.11) holds for δ_n shrinking slower than $n^{-1/4}$ if (B.11) holds for δ_n shrinking faster than $n^{-1/4}$.

For any $\delta \in (0, 1)$, we have

$$\max_{b_n} J(\delta, \mathcal{D}_n(P^0_{b_n})) \leq \int_0^\delta \sqrt{1 + G(\epsilon)} d\epsilon \equiv UB(\delta) \text{ with probability 1.}$$

By the dominated convergence theorem, $UB(\delta_n)$ converges to 0 as $n \to \infty$. Using that $\lim_{n\to\infty} \delta_n^{-2} n^{-1/2} = 0$, we have that, for all n large enough,

$$\max_{b_n} J(\delta_n, \mathcal{D}_n(P^0_{b_n})) \left(1 + \frac{\sqrt{V} J(\delta_n, \mathcal{D}_n(P^0_{b_n}))}{\delta_n^2 \sqrt{n}}\right) \leq UB(\delta_n)\left(1 + UB(\delta_n)\right)$$

with probability 1. Taking the expectation of the left and taking the limit as $n \to \infty$ yields (B.12). □

Another possible scenario is that the estimators corresponding to each β are very different from each other so that using the covering number bound does not gain us much relative to just considering each candidate separately. Suppose that the index set Σ_n contains at most $K_\beta(n) < \infty$ elements. Then a trivial bound on the covering number $N(\epsilon, \mathcal{D}_n(P^0_{b_n}), L_2(Q))$ is given by $K_\beta(n)$. This yields the following result.

Corollary 4. *Suppose that each index set Σ_n for β at sample size n contains at most $K_\beta(n) < \infty$ elements. Then*

$$\left| P_{B_n}(P^1_{B_n} - P)\left[D^*(\hat{P}_{\beta_n}(P^0_{B_n})) - D^*(P)\right]\right| = O_P\left(\sqrt{\frac{1 + \log K_\beta(n)}{n}}\left(\delta_n + \sqrt{\frac{1 + \log K_\beta(n)}{n/V}}\right)\right).$$

Consequently, the condition (B.12) in Theorem 13 can be replaced by the condition that $(\delta_n + n^{-1/2})\sqrt{\log K_\beta(n)} \to 0$.

Proof. First note that $J(\delta, \mathcal{D}_n(P^0_{b_n})) \leq \delta\sqrt{1 + \log K_\beta(n)}$ for all instances of b_n and $P^0_{b_n}$. Consequently, the bound from Van Der Vaart and Wellner [2011] yields

$$\max_{b_n} E_{P^n}\left[\sqrt{n} \sup_{d \in \mathcal{D}_n(P^0_{b_n})} |P^1_{b_n} - P|d \,\middle|\, P_{b^0_n}\right] \lesssim \sqrt{1 + \log K_\beta(n)}\left(\delta_n + \sqrt{\frac{1 + \log K_\beta(n)}{n/V}}\right).$$

Replacing t in (B.7) with t times the upper bound above yields the first result. If $(\delta_n + n^{-1/2})\sqrt{\log K_\beta(n)} \to 0$, then the right-hand side above, which upper bounds the expectation in (B.12), converges to zero as $n \to \infty$. □

Suppose we let $K_\beta(n)$ increase as a polynomial in n. Then the corollary shows that the term $P_{B_n}(P^1_{B_n} - P)\left[D^*(\hat{P}_{\beta_n}(P^0_{B_n})) - D^*(P)\right]$ converges to zero at the rate

$$\max\left\{\delta_n\sqrt{\frac{\log n}{n}}, \frac{\log n}{n}\right\}.$$

This rate is optimal up to a log factor. To see that this is rate is nearly optimal, note that $(P_n - P)d_n = O_P(\sqrt{Pd_n^2/n})$ by Chebyshev's inequality when d_n is a deterministic sequence of functions with finite $L_2(P)$ norm. In the above rate, δ_n (essentially) plays the role of $\sqrt{Pd_n^2}$. We say essentially because (B.11) requires that δ_n is a little bit slower than this rate. If we had estimated $D^*(P)$ with a correctly specified parametric model then we would typically expect our estimate D^*_n to satifsy $P(D^*_n - D^*(P))^2 = O_P(1/n)$, thus suggesting that δ_n will converge to zero more slowly than $n^{-1/2}$, and most likely also more slowly then $n^{-1/2}\sqrt{\log n}$.

Cross-validated TMLE

Suppose now that β_n has been chosen to satisfy the additional property that

$$P_{B_n}P^1_{B_n}D^*(\hat{P}_{\beta_n}(P^0_{B_n})) = o_P(n^{-1/2}).$$

The left-hand side is a function of the data only so whether or not β_n satisfies this condition can be verified empirically. Typically β_n will be a high-dimensional vector where most of the dimensions are used to estimate the needed components of the likelihood, and few dimensions (often one dimension) are used to fluctuate these initial estimates to satisfy the above identity. One can then estimate $\Psi(P)$ with

$$\psi_n^{cvtmle} \equiv P_{B_n}\Psi(\hat{P}_{\beta_n}(P^0_{B_n})).$$

Such an estimator is known as a cross-validated targeted minimum loss based estimator (CV-TMLE). If the $o_P(n^{-1/2})$ term is equal to zero, then the results for the above estimator are identical to that of the cross-validated one-step estimator given in the previous subsection. Otherwise one has that

$$\begin{aligned}&\sqrt{n}\left(\psi_n^{cvtmle} - \Psi(P)\right)\\ &= \sqrt{n}\left(P_{B_n}P^1_{B_n}\left[D^*(\hat{P}_{\beta_n}(P^0_{B_n})) + \Psi(\hat{P}_{\beta_n}(P^0_{B_n}))\right] - \Psi(P)\right) + \sqrt{n}P_{B_n}P^1_{B_n}D^*(\hat{P}_{\beta_n}(P^0_{B_n})).\end{aligned}$$

The second term on the right is asymptotically negligible and the first term yields the same analysis as the cross-validated one-step estimator.

Appendix C

opttx Vignette

To illustrate the use of the opttx package, we simulated from the data generating distribution described in section 3.7.

```
#load opttx package
library(opttx,verbose=FALSE)
```

The format of the simulated data is as follows:

```
#load simulated example data
data(opttx_sim_data)
str(opttx_sim_data)

## 'data.frame': 1000 obs. of  11 variables:
##  $ W1  : num  -0.321 -0.335 -0.153 1.523 0.57 ...
##  $ W2  : num  -0.9513 0.0289 1.4711 0.88 -0.6791 ...
##  $ W3  : num  0.963 -0.929 -1.271 -1.105 0.834 ...
##  $ W4  : num  0.7469 -0.0845 -0.4186 1.5803 -3.2008 ...
##  $ W5  : num  0.5841 1.3042 0.4621 1.8003 0.0368 ...
##  $ A   : Factor w/ 3 levels "1","2","3": 2 2 2 1 3 2 3 2 3 2 ...
##  $ Y   : num  1 0 0 1 0 1 1 1 1 0 ...
##  $ d0  : int  3 2 1 1 3 1 2 1 2 1 ...
##  $ Yd0 : num  1 0 0 1 0 1 1 1 1 0 ...
##  $ g0W : num [1:1000, 1:3] 0.303 0.344 0.313 0.445 0.373 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr  "A1" "A2" "A3"
##  $ Q0aW: num [1:1000, 1:3] 0.182 0.284 0.128 0.615 0.457 ...
```

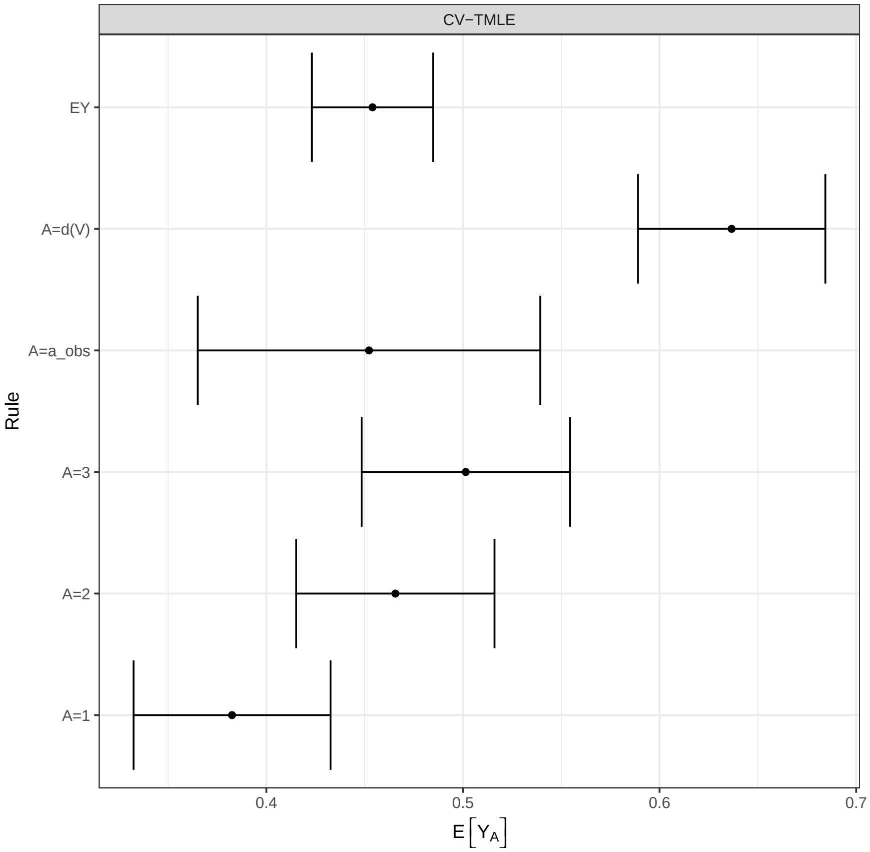
R Example C.0.1 shows the basic usage for the main opt_tmle function, which learns the optimal rule and estimates its performance using CV-TMLE. The most important arguments are the `data.frame`, the `SL.library` specification, and the relevant covariate, treatment, and outcome nodes (`Wnodes`, `Anode`, and `Ynode`, respectively). Unless otherwise specified (using the `Vnode` argument), all W nodes will be used to learn the rule.

R Example C.0.1.

```
#specify Super Learner Library for various components

SL.library <- list(Q = c("SL.glm", "SL.glmem", "SL.glmnet", "SL.glmnetem",
                         "SL.step.forward", "SL.gam", "SL.mean"),
                   g = c("mnSL.glmnet", "mnSL.multinom", "mnSL.mean"),
                   QaV = c("SL.glm", "SL.glmnet",
                           "SL.step.forward", "SL.gam", "SL.mean"))

#Important: convert library for QaV to a multivariate Super Learner library.
SL.library$QaV <- sl_to_mv_library(SL.library$QaV)


#Learn optimal rule and estimate its performance
opt_obj <- opt_tmle(opttx_sim_data, SL.library=SL.library,
                    Wnodes=c("W1","W2","W3","W4","W5"), Anode="A", Ynode="Y")

#show CV-TMLE estimates of rule performance
print(opt_obj)

## Treatment Assignments
##
##             optimal tx
## observed tx     1     2     3   Sum
##          1   0.077 0.131 0.119 0.327
##          2   0.084 0.096 0.158 0.338
##          3   0.060 0.119 0.156 0.335
##          Sum 0.221 0.346 0.433 1.000
##
##
## EYa Estimates
##
##    Estimate         SE CI Lower Bound CI Upper Bound Intervention
## 1 0.3825316 0.02556455      0.3324261      0.4326372          A=1
## 2 0.4656219 0.02572673      0.4151984      0.5160454          A=2
## 3 0.5014256 0.02703047      0.4484468      0.5544043          A=3
## 4 0.4522018 0.04445559      0.3650705      0.5393332      A=a_obs
## 5 0.6366442 0.02432419      0.5889696      0.6843187       A=d(V)
## 6 0.4540000 0.01575221      0.4231262      0.4848738           EY
##   Estimator
## 1   CV-TMLE
## 2   CV-TMLE
## 3   CV-TMLE
## 4   CV-TMLE
## 5   CV-TMLE
## 6   CV-TMLE
```

The CV-TMLE estimates of the mean outcomes under the various interventions can also be easily plotted:

```
plot(opt_obj)
```

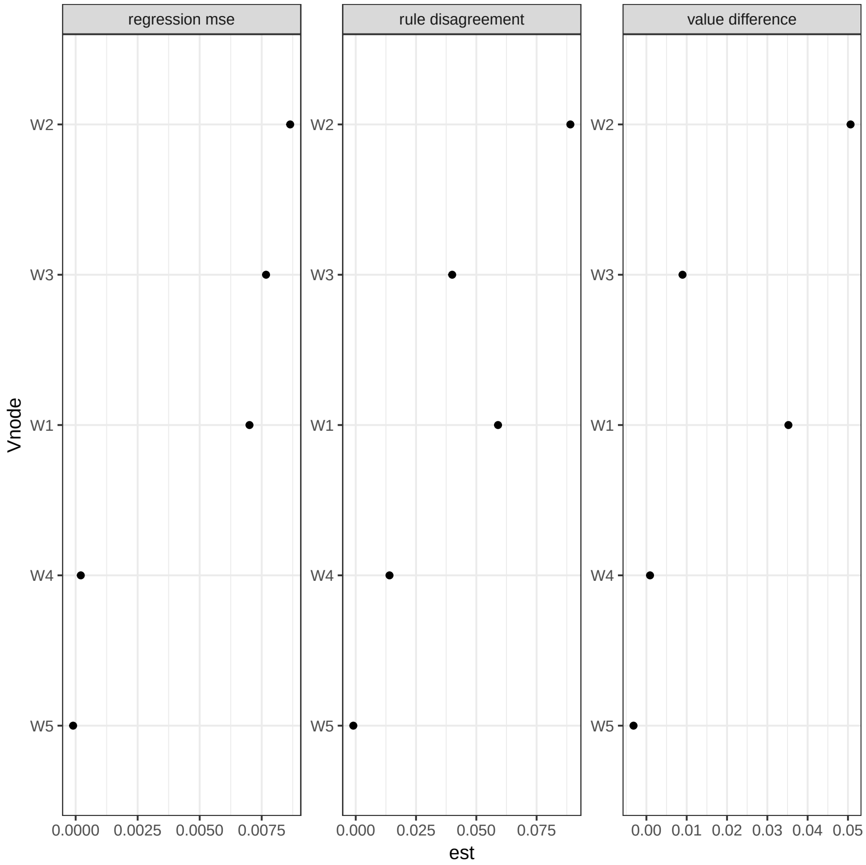


Having obtained an initial estimate of the optimal rule and its value, we can now estimate the importance of the covariates for this rule:

R Example C.0.2.

```
vimresult <- opttx:::backward_vim(opt_obj)

## Vnode W1
## Vnode W2
## Vnode W3
## Vnode W4
## Vnode W5
```

```
#plot VIM estimates
ggplot(vimresult$vimdf,aes(y=Vnode,x=est))+
  geom_point()+
  facet_wrap(~metric,scales="free")+theme_bw()
```



Bibliography

R. D. B and M. J. van der Laan. Statistical Issues and Limitations in Personalized Medicine Research with Clinical Trials. *The International Journal of Biostatistics*, 8(1):1–20, 2012.

P. J. Bickel, C. A. J. Klaassen, Y. Ritov, and J. A. Wellner. *Efficient and adaptive estimation for semiparametric models.* Johns Hopkins University Press, Baltimore, 1993.

P. J. Bickel, F. Götze, and W. R. van Zwet. Resampling fewer than n observations: gains, losses, and remedies for losses. *Statistica Sinica*, 7(1):1–31, 1997.

B. Chakraborty and E. Moodie. *Statistical methods for dynamic treatment regimes.* Springer, 2013.

F. S. Collins and H. Varmus. A New Initiative on Precision Medicine. *dx.doi.org*, 372(9): 793–795, Feb. 2015.

J. Coyle. origami. 2014.

A. C. Davison and D. V. Hinkley. *Bootstrap methods and their application*, volume 1 of *Cambridge Series in Statistical and Probabilistic Mathematics.* Cambridge University Press, Cambridge, Cambridge, 1997.

T. J. DiCiccio and J. P. Romano. A review of bootstrap confidence intervals. *Journal of the Royal Statistical Society Series B (. . .*, 1988.

T. J. DiCiccio and J. P. Romano. Nonparametric Confidence Limits by Resampling Methods and Least Favorable Families. *International Statistical Review / Revue Internationale de Statistique*, 58(1):59, Apr. 1990.

S. Dudoit and M. J. van der Laan. Asymptotics of cross-validated risk estimation in estimator selection and performance assessment. *Statistical Methodology*, 2(2):131–154, July 2005.

S. Dudoit, M. J. van der Laan, S. Keleş, A. M. Molinaro, S. E. Sinisi, and S. L. Teng. Loss-based estimation with cross-validation: Applications to microarray data analysis. *SIGKDD Explor. Newsl.*, 5(2):56–68, Dec. 2003. ISSN 1931-0145. doi: 10.1145/980972.980981. URL `http://doi.acm.org/10.1145/980972.980981`.

B. Efron. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 1987.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap, Monographs on Statistics and Applied Probability, Vol. 57.* New York and London: Chapman and Hall/CRC, 1993.

P. Hall. Theoretical comparison of bootstrap confidence intervals. *The Annals of Statistics*, 1988.

P. Hall. *The Bootstrap and Edgeworth Expansion.* Springer Series in Statistics. Springer-Verlag, New York, NY, 1992.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning* . Data Mining, Inference, and Prediction. Springer-Verlag, New York, NY, 2003.

J. B. Holcomb, B. C. Tilley, S. Baraniuk, E. E. Fox, C. E. Wade, J. M. Podbielski, D. J. del Junco, K. J. Brasel, E. M. Bulger, R. A. Callcut, M. J. Cohen, B. A. Cotton, T. C. Fabian, K. Inaba, J. D. Kerby, P. Muskat, T. O'Keeffe, S. Rizoli, B. R. H. Robinson, T. M. Scalea, M. A. Schreiber, D. M. Stein, J. A. Weinberg, J. L. Callum, J. R. Hess, N. Matijevic, C. N. Miller, J.-F. Pittet, D. B. Hoyt, G. D. Pearson, B. Leroux, and G. van Belle. Transfusion of Plasma, Platelets, and Red Blood Cells in a 1:1:1 vs a 1:1:2 Ratio and Mortality in Patients With Severe Trauma: The PROPPR Randomized Clinical Trial. *Jama*, 313(5): 471–482, Feb. 2015.

A. E. Hubbard, S. Kherad-Pajouh, and M. J. van der Laan. Statistical Inference for Data Adaptive Target Parameters. *The International Journal of Biostatistics*, 12(1), 2016.

A. R. Luedtke and M. J. van der Laan. Super-learning of an optimal dynamic treatment rule. Technical Report 326, available at `http://www.bepress.com/ucbbiostat/`, Division of Biostatistics, University of California, Berkeley, under review at JCI, 2014.

A. R. Luedtke and M. J. van der Laan. Statistical inference for the mean outcome under a possibly non-unique optimal treatment strategy. *The Annals of Statistics*, 44(2):713–742, Apr. 2016a.

A. R. Luedtke and M. J. van der Laan. Super-Learning of an Optimal Dynamic Treatment Rule. *The International Journal of Biostatistics*, 12(1), 2016b.

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

J. Pfanzagl. *Estimation in semiparametric models.* Springer, Berlin Heidelberg New York, 1990.

E. C. Polley and M. J. van der Laan. Super learner in prediction. 2010.

E. C. Polley and M. J. van der Laan. *SuperLearner: Super Learner prediction, Package Version 2.0-4*. Vienna: R Foundation for Statistical Computing, 2011.

J. M. Robins. Optimal structural nested models for optimal sequential decisions. In D. Y. Lin and H. P, editors, *Proc. Second Seattle Symp. Biostat.*, volume 179, pages 189–326, 2004a.

J. M. Robins. Optimal Structural Nested Models for Optimal Sequential Decisions. In *Proceedings of the Second Seattle Symposium in Biostatistics*, pages 189–326. Springer New York, New York, NY, 2004b.

R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction, 1998.

A. W. Vaart, S. Dudoit, and M. J. Laan. Oracle inequalities for multi-fold cross validation. *Statistics & Decisions*, 24(03):351–371, 2006.

M. J. van der Laan and S. Dudoit. Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: finite sample oracle inequalities and examples. Technical Report 130, Division of Biostatistics, University of California, Berkeley, 2003a.

M. J. van der Laan and S. Dudoit. Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities .... 2003b.

M. J. van der Laan and A. R. Luedtke. Targeted learning of an optimal dynamic treatment, and statistical inference for its mean outcome. 2014.

M. J. van der Laan and A. R. Luedtke. Targeted Learning of the Mean Outcome under an Optimal Dynamic Treatment Rule. *Journal of Causal Inference*, 3(1):61–95, 2015.

M. J. van der Laan and J. M. Robins. *Unified methods for censored longitudinal data and causality*. Springer Series in Statistics. Springer-Verlag, New York, New York, NY, 2003.

M. J. Van Der Laan and S. Rose. Targeted Learning: Causal Inference for Observational and Experimental Data - Mark J. van der Laan, Sherri Rose - Google Books. 2011.

M. J. van der Laan and S. Rose. *Targeted learning: causal inference for observational and experimental data*. 2011.

M. J. van der Laan and D. Rubin. Targeted Maximum Likelihood Learning. *The International Journal of Biostatistics*, 2(1), 2006.

M. J. van der Laan, S. Dudoit, and A. W. van der Vaart. The Cross-Validated Adaptive Epsilon-Net Estimator. *Stat. Decis.*, 24(3):373–395, Feb. 2006.

M. J. van der Laan, E. C. Polley, and A. E. Hubbard. Super learner. *Statistical applications in genetics and molecular biology*, 6(1):Article25, 2007a.

M. J. van der Laan, E. C. Polley, and A. E. Hubbard. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6:Art. 25–23 pp. (electronic), 2007b.

A. Van Der Vaart and J. A. Wellner. A local maximal inequality under uniform entropy. *Electronic Journal of Statistics*, 2011.

A. W. van der Vaart. *Asymptotic statistics*. Cambridge University Press, New York, 1998.

A. W. van der Vaart and J. A. Wellner. *Weak convergence and empirical processes*. Springer, Berlin Heidelberg New York, 1996.

A. W. van der Vaart, S. Dudoit, and M. J. van der Laan. Oracle inequalities for multi-fold cross validation. *Statistics & Decisions*, 24(3):351–371, Dec. 2006a.

A. W. van der Vaart, S. Dudoit, and M. J. van der Laan. Oracle Inequalities for Multi-Fold Cross-Validation. *Stat Decis*, 24(3):351–371, 2006b.

B. Zhang, A. A. Tsiatis, M. Davidian, M. Zhang, and E. Laber. Estimating optimal treatment regimes from a classification perspective. *Stat*, 1(1):103–114, Oct. 2012.

Y. Zhao, D. Zeng, A. J. Rush, and M. R. Kosorok. Estimating Individualized Treatment Rules Using Outcome Weighted Learning. *Journal of the American Statistical Association*, 107(499):1106–1118, June 2012.

Y.-Q. Zhao, D. Zeng, E. B. Laber, and M. R. Kosorok. New Statistical Learning Methods for Estimating Optimal Dynamic Treatment Regimes. *Journal of the American Statistical Association*, 110(510):583–598, July 2014.

W. Zheng and M. J. Van Der Laan. Asymptotic theory for cross-validated targeted maximum likelihood estimation. 2010.

W. Zheng and M. J. van der Laan. Targeted Maximum Likelihood Estimation of Natural Direct Effect. Technical Report 288, Division of Biostatistics, University of California, Berkeley, 2011.