Machine learning for causal inference: on the use of cross-fit estimators

Paul N Zivich[1,2], Alexander Breskin[3]

[1]Department of Epidemiology, Gillings School of Global Public Health, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA

[2]Carolina Population Center, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA

[3]NoviSci, Durham, NC, USA

Abstract

Modern causal inference methods allow machine learning to be used to weaken parametric modeling assumptions. However, the use of machine learning may result in bias and incorrect inferences due to overfitting. Cross-fit estimators have been proposed to eliminate this bias and yield better statistical properties.

We conducted a simulation study to assess the performance of several different estimators for the average causal effect (ACE). The data generating mechanisms for the simulated treatment and outcome included log-transforms, polynomial terms, and discontinuities. We compared singly-robust estimators (g-formula, inverse probability weights) and doubly-robust estimators (augmented inverse probability weights, targeted maximum likelihood estimation). Nuisance functions were estimated with parametric models and ensemble machine learning, separately. We further assessed cross-fit doubly-robust estimators.

With correctly specified parametric models, all of the estimators were unbiased and confidence intervals achieved nominal coverage. When used with machine learning, only the cross-fit estimators were unbiased and had nominal confidence interval coverage.

Due to the difficulty of properly specifying parametric models in high dimensional data, doubly-robust estimators with ensemble learning and cross-fitting may be the preferred approach for estimation of the ACE in most epidemiologic studies. However, these approaches may require larger sample sizes to avoid finite-sample issues.

INTRODUCTION

Most causal inference methods for observational data utilize so-called nuisance functions. These functions are not of primary interest but are used as inputs into estimators [1]. For instance, the propensity score function is the nuisance function for the inverse probability of treatment weighted estimator. In low-dimensional settings, these nuisance functions are possible to estimate nonparametrically. However, in more realistic settings, such as those that involve continuous covariates or in which the minimal sufficient adjustment set of confounders is large, parametric nuisance models are often used. Proper specification of these nuisance models is then required for the resulting estimator to be consistent. Specifically, a properly specified model contains the true function as a possible realization. Given the complex underlying relationships between variables, such parametric models are often difficult to specify correctly.

Data-adaptive supervised machine learning methods (which fit to a function using data-driven tuning parameters) [2, 3], have been suggested as an alternative approach to estimate nuisance functions in high-dimensional settings while not imposing overly restrictive parametric functional form assumptions [4, 5, 6, 7, 8]. Despite this optimism, issues in the use of machine learning for nuisance function estimation have become apparent [9, 10]. Notably, some machine learning algorithms converge to the true answer at a slow rate, leading to substantial under-coverage of corresponding confidence intervals. The use of doubly-robust estimators allow for the use of slower converging nuisance models; but issues of smoothness and overfitting remain which may preclude the use of complex machine learning approaches.

Cross-fit estimators have been developed to reduce overfitting and impose less restrictive smoothness assumptions on machine learning algorithms used to estimate nuisance functions [10, 11]. These estimators relax the convergence conditions to allow for a richer set of algorithms to be used. Cross-fit estimators share similarities with double machine learning [10], cross-validated estimators [12], and sample splitting [13]. An extension referred to as double cross-fitting has recently been proposed as an even less restrictive approach [11]. In this work, we detail a general procedure for double cross-fit estimators and demonstrate their performance in a simulation study. We compare a wide range of estimators in the context of a simulated study of statins and subsequent atherosclerotic cardiovascular disease (ASCVD).

METHODS

Data generating mechanism

Suppose we observe an independent and identically distributed sample $(O_1, ..., O_n)$ where $O_i = (X_i, Y_i, Z_i) \sim F$. Let X indicate statin use, Y indicate the observed incidence of ASCVD, and Y^x indicate the potential value Y would have taken if, possibly counter to fact, an individual received treatment x. The ACE is then

$$\psi = E[Y^1 - Y^0] = E[Y^1] - E[Y^0]$$

Note that the potential outcomes are not necessarily directly observed, so a set of conditions are needed to identify the ACE from observable data. Specifically, we assume the following conditions hold:

1. Counterfactual consistency [14]
$$Y_i = X_i Y_i^1 + (1 - X_i) Y_i^0$$

2. Conditional exchangeability [15]
$$Y^x \amalg X | Z \ \forall \ x \in \{0, 1\}$$

3. Positivity [16]
$$\text{if } \Pr(Z = z) > 0 \text{ then } \Pr(X = x | Z = z) > 0 \ \forall \ z \in \mathcal{Z}, x \in \{0, 1\}$$

Together, these conditions allow the ACE to be identified as:

$$\int_{z \in \mathcal{Z}} \left(E[Y|X = 1, Z = z] - E[Y|X = 0, Z = z] \right) dF_Z$$

We considered the following confounders in Z for the simulated data generating process: age (A), low-density lipoprotein (L), ASCVD risk scores (R), and diabetes (D). Therefore, $Z = (A, L, R, D)$ in the following simulated data. These factors were chosen based on the 2018 primary prevention guidelines for the management of blood cholesterol [17]. Full details on the data generating mechanism are provided in the Appendix. The incidence of statin use was chosen to be similar to reported empirical trends in US adults [18], and generated from the following model inspired by the 2018 primary prevention guidelines:

$$\begin{aligned} \Pr(X = 1|Z) = \text{Bernoulli}(\text{expit}(&-3.471 + 1.390 D_i + 0.112 L_i + 0.973\ I(L_i > \log(60)) \\ &- 0.046(A_i - 30) + 0.003(A_i - 30)^2 + 0.273\ I(0.05 \leq R_i < 0.075) \\ &+ 1.592\ I(0.075 \leq R_i < 0.2) + 2.461\ I(R_i \geq 0.2))) \end{aligned}$$

The ASCVD potential outcomes under each potential value of X were generated from the following model:

$$\begin{aligned} \Pr(Y^x = 1|Z) = \text{Bernoulli}(\text{expit}(&-6.250 - 0.75x + 0.35x(5 - L_i)\ I(L_i < \log(130)) + 0.45(A_i - 39)^{0.5} \\ &+ 1.75 D_i + 0.29 \exp(R_i + 1) + 0.14 L_i^2\ I(L_i > \log(120))))) \end{aligned}$$

The observed outcome was calculated as $Y_i = Y_i^1 X_i + (1 - X_i)Y_i^0$. The nuisance functions considered are:

$$\Pr(X|Z; \gamma) = \gamma Z \tag{1}$$

$$\Pr(Y|X, Z; \beta, \eta) = \beta X + \eta Z \tag{2}$$

As the name implies, these nuisance functions are not of interest but are used for the estimation of the ACE. Unlike in simulated data, the correct specification of these models is often unknown; and in the context of parametric models, must be *a priori* specified.

Nuisance function estimators

Before estimating the ACE, the nuisance model parameters (γ, β, η) need to be estimated. Much of the previous work in causal inference has relied on parametric regression models. However, these models must be sufficiently flexible to capture the true nuisance functions. In our simulations we consider two different parametric model specifications. First, we consider the correct model specification as described previously. This is the best-case scenario for researchers. Unfortunately, this case is unlikely to occur. Second, we considered a main-effects model, where all variables were assumed to be linearly related to the outcome and no interaction terms were included in the model. The main-effects model is quite rigid and does not contain the true density function.

As an alternative to the parametric models, we consider several data-adaptive machine learning algorithms. There are a variety of potential supervised machine learning algorithms and there is no guarantee on which algorithm will perform best in all scenarios [19]. Therefore, we utilize super-learner with 10-fold cross-validation to estimate the nuisance functions. Super-learner is a generalized ensemble algorithm that allows for the combination of multiple predictive algorithms into a single prediction function [20, 21]. Super-learner has been previously shown to asymptotically perform as well as the best performing algorithm included within the super-learner procedure [20], with studies of finite sample performance indicating similar results. Within super-learner, we included the following algorithms: the empirical mean, main-effects logistic regression without regularization, generalized additive model with 4 splines and a ridge penalty of 0.6 [22], generalized additive model with 6 splines, random forest with 500 trees and

a minimum of 20 individuals per leaf [23], and a neural network with a single hidden layer consisting of four nodes. Only non-processed main-effects variables were provided to each learner.

Estimators for the ACE

After estimation of the nuisance functions, the predictions can be used in estimators for the ACE. We considered 4 estimators: g-computation, an inverse probability weighted (IPW) estimator, an augmented inverse probability weighted (AIPW) estimator, and a targeted maximum likelihood estimator (TMLE). IPW only requires nuisance function from Equation 1 and g-computation only requires the nuisance function from Equation 2. Due to their reliance on a single nuisance function, these methods are said to be singly-robust. However, these singly-robust estimators require fast convergence of nuisance models, severely limiting which algorithms can be used. AIPW and TMLE instead use both nuisance functions from both Equations 1 and 2, and have the property of double robustness; such that if either nuisance function is correctly estimated, then the point estimate will be consistent [4, 24, 25, 26]. Perhaps more important in the context of machine learning, these doubly-robust estimators allow for slower convergence of nuisance function estimators. However, all of these estimators require the nuisance models to be smooth, in the sense that the belong to the so-called Donsker class [10]. Intuitively, members of this class are less prone to overfitting than models outside the class. For models that do not belong to the Donsker class, confidence intervals may be overly narrow and result in misleading inference. Recent work has demonstrated that cross-fitting weakens the smoothness conditions for the nuisance function estimators, which allows for a more diverse set of algorithms. A double cross-fit procedure allows for further theoretical improvements [11]. Therefore, we additionally considered double cross-fit alternatives for AIPW (DC-AIPW) and TMLE (DC-TMLE). We briefly outline each estimator, with further details and formulas provided in the Appendix.

G-formula

We used the g-computation procedure described by Snowden et al. 2011 to estimate the ACE [27]. Briefly, the outcome nuisance model is fit using the observed data. From the fit outcome nuisance model, the probability of Y is predicted under $X = 1$ and under $X = 0$ for all individuals. The ACE is calculated by taking the average of the differences in the estimated Y under each treatment plan. Wald-type 95% confidence intervals were generated using the standard deviation of 250 bootstrap samples with replacement, each of size n.

IPTW

In contrast to the g-formula, IPTW rely on estimation of the treatment nuisance models. From the predicted probabilities of X, weights are constructed by taking the inverse of the predicted probabilities of the observed X. These weights are then used to calculate the weighted average of Y among subjects with each value of X. We used robust standard errors that ignore the estimation of the nuisance model, which results in conservative variance estimates for the ACE [28]. Therefore, confidence interval coverage is expected to be at least 95% when the nuisance model is properly specified.

AIPW

AIPW uses both the treatment and outcome nuisance model predictions to estimate the ACE. Predicted probabilities of the treatment and outcome are combined via a single equation to generate predictions under each value of X, with confidence intervals calculated from the influence curve [24, 25].

Figure 1: General double cross-fit procedure for doubly-robust estimators
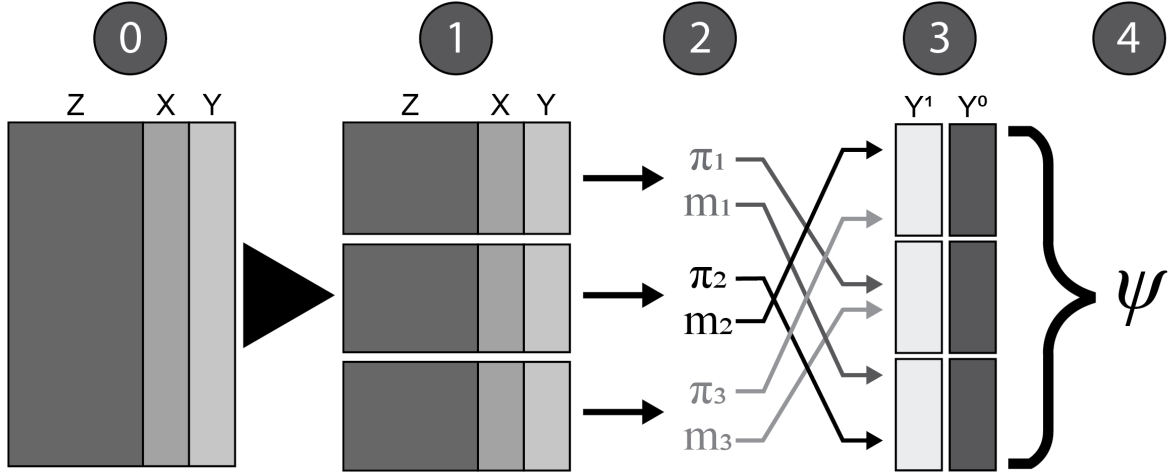
Step 0) The exposure (X), outcome (Y), and minimally sufficient adjustment set for identification (Z) are selected and collected.

Step 1) The data is split into three approximately equal-sized sample splits.

Step 2) Algorithms for the treatment nuisance model and the outcome nuisance model are fit to each sample split.

Step 3) Predicted outcomes under each treatment are estimated using the nuisance models estimated using discordant data sets. For example, sample split 1 uses the treatment nuisance model from sample split 3 and the outcome nuisance model from sample split 2.

Step 4) The target parameter is calculated from the mean of the predictions across all splits. The variance for that particular sample split is calculated as the mean of variances for each split.

Steps 1-4 are repeated a number of times to reduce sensitivity to particular sample splits. The overall point estimate is calculated as the median of the point estimates for all of the different splits. The estimated variance consists of two parts: the variability of the ACE within a particular split and the variance of the ACE point estimate between each split.

TMLE

TMLE similarly uses both the treatment and outcome nuisance model predictions to construct a single estimate. Unlike AIPW, TMLE uses a 'targeting' step that corrects the bias-variance tradeoff in the estimation [4]. This is accomplished by fitting parametric working model, where the observed Y is modeled as a function of the so-called clever covariate with the outcome nuisance model predictions included as an offset. The targeted predictions under each value of X from this model are averaged, and their difference provides an estimate of the ACE. Confidence intervals are calculated from the influence curve.

Double cross-fit

A visualization of the double cross-fit procedure is provided in Figure 1. This process is compatible with both doubly-robust estimators previously described. First, the data set is split into three approximately equal-sized groups (although this can be generalized to numbers larger than three [10, 11]). Note that the splits are non-overlapping, so that each subject occurs in only 1 split. Second, treatment and outcome nuisance model parameters are estimated in individual sample splits. This involves using the super-learner fitting procedure independently for each split (for a total of six times). Third, predicted treatment probabilities and expected values for outcomes are calculated from the nuisance models in the discordant splits, such that the predictions from super-learner do not come from the data used to fit the algorithms. For example, sample split 1 has the probability of treatment predicted with the treatment nuisance model from split 3; and the expected value of the outcome predicted with the outcome nuisance model from split 2. For each of the splits, the estimator (e.g. AIPW, TMLE, etc.) is calculated from the treatment and outcome predictions. In a final step, the estimates from all splits are averaged together to produce a final point estimate.

Since the ACE is dependent on a particular splitting of the input data, the procedure is repeated a number of times with different possible splits. In our implementation, we used 100 different splits as is recommended in other cross-fitting procedures [10]. Results were unstable when only using 10 splits (results not shown). The overall point

Table 1: Descriptive characteristics for a single sample

	Statin (n=776)	No statin (n=2224)
Age (SD)	58 (9.5)	53 (7.6)
Diabetes	31%	1%
log(LDL) (SD)	4.92 (0.2)	4.86 (0.2)
Risk score (SD)	0.15 (0.2)	0.06 (0.1)
ASCVD	37%	29%

Descriptive statistics for a single sample from mthe data generating mechanism. Continuous variables are preseted as mean (standard deviation)
SD: standard deviation, LDL: low-density lipoproteins, ASCVD: atherosclerotic cardiovascular disease.

estimate of the ACE is calculated as the median of the ACE for all splits s

$$\widetilde{ACE} = \text{median}(ACE_s)$$

While the mean can also be chosen, it is more susceptible to outliers and may require a larger number of different splits [10]. The estimated variance for the ACE consists of two parts: the variability of the ACE within a particular split and the variance of the ACE point estimate between each split. The variance for the s different splits, is the median of

$$Var(\widetilde{ACE}) = \text{median}(Var(ACE_s) + (ACE_s - \widetilde{ACE})^2)$$

Performance metrics

To compare the performance of estimators for each of the 18 different combinations, we calculated bias, empirical standard error (ESE), root mean squared error (RMSE), average standard error (ASE), average confidence limit difference (CLD), and 95% confidence limit coverage over 2000 simulated samples. Bias was defined as mean of the estimated ACE from each simulation minus the true population ACE. ESE was calculated as the standard error of the estimate from each simulation. RMSE was defined the square root of bias squared plus ESE squared. ASE was the mean of the estimated standard error from each simulation. CLD was the mean of the upper confidence limit minus the lower confidence limit. 95% confidence interval coverage was calculated as the proportion of confidence intervals containing the true population ACE.

All simulations were conducted using Python 3.5.1 (Python Software Foundation, Wilmington, DE) with the sklearn implementations of the previously described algorithms [29]. Outside of the specified parameters, the defaults of the software were used. The true ACE was calculated as the average difference in potential outcomes for a population of 10,000,000 individuals. Simulations for estimation of the ACE were repeated 2000 times for n=3000. The sample size for simulations was chosen such that when split into three equally sized groups (n=1000), the true parametric model could be fit and correctly estimate the true ACE.

RESULTS

Before presentation of the simulations, we present estimators in the context of a single data set. Characteristics of the single study sample are displayed in Table 1. Results for the estimators are presented in Table 2. Nuisance models estimated with machine learning led to substantially narrower confidence intervals as indicated by the CLD. Differences were less stark for the double cross-fit estimators. Broadly, run-times for estimators of ACE were short. The double cross-fit estimators had substantially longer run times due to the repeated sample splitting procedure. As reference, a single estimation of the DC-AIPW with super-learner required 660 different predictive models to be fit.

Table 2: Estimated risk differences for a single sample from the data generating mechanism with main-terms

|	RD	SD(RD)	95% CL	CLD	Run-time*
G-formula					
Main-effects	-0.14	0.016	-0.17, -0.11	0.06	0.9
Machine learning	-0.09	0.015	-0.12, -0.06	0.06	82.3
IPTW					
Main-effects	-0.13	0.039	-0.20, -0.05	0.15	0.0
Machine learning	-0.11	-0.028	-0.16, -0.05	0.11	0.3
AIPW					
Main-effects	-0.08	0.038	-0.16, -0.01	0.15	0.0
Machine learning	-0.11	0.016	-0.14, -0.08	0.06	0.7
TMLE					
Main-effects	-0.12	0.029	-0.18, -0.06	0.11	0.0
Machine learning	-0.12	0.016	-0.15, -0.09	0.06	0.7
DC-AIPW					
Main-effects	-0.09	0.040	-0.16, -0.01	0.16	1.3
Machine learning	-0.11	0.022	-0.16, -0.07	0.09	128.1
DC-TMLE					
Main-effects	-0.12	0.028	-0.18, -0.07	0.11	1.3
Machine learning	-0.11	0.021	-0.15, -0.07	0.08	129.9

RD: risk difference, SD(RD): standard deviation for the risk difference, 95% CL: 95% confidence limits, CLD: confidence limit difference defined as the upper confidence limit minus the lower confidence limit.

Machine learning estimators were super-learner with 10-fold cross validation. Algorithms included were the empirical mean, main-effects logistic regression without regularization, generalized additive model with 4 splines and a ridge penalty of 0.6, generalized additive model with 6 splines, random forest with 500 trees and a minimum of 20 individuals per leaf, and a neural network with a single hidden layer consisting of four nodes.

Double cross-fit procedures included 100 different sample splits.

* Run times are based on a server running on a single 2.5 GHz processor with 5 GB of memory allotted. Run times are indicated in minutes. G-formula run-times are large due to the use of a bootstrap procedure to calculate the variance for the risk difference. IPTW used robust variance estimators. AIPW, TMLE, DC-AIPW, and DC-TMLE variances were calculated using influence curves.

Figure 2: Bias and confidence interval coverage of estimators of the average causal effect
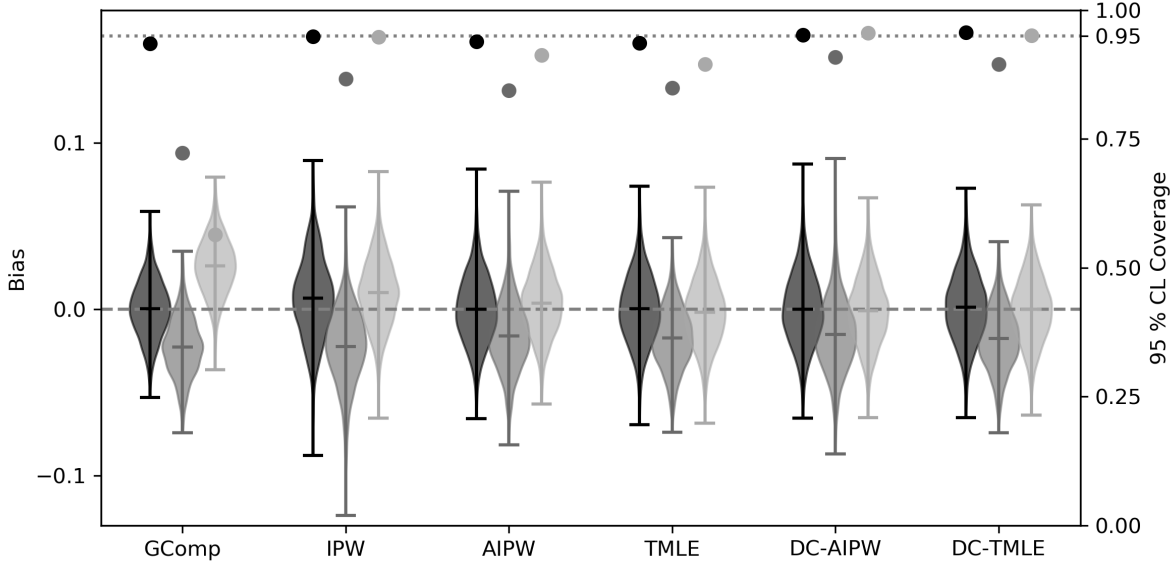
GComp: g-formula, IPW: inverse probability of treatment weights, AIPW: augmented inverse probability weights, TMLE: targeted maximum likelihood, DC-AIPW: double cross-fit AIPW, DC-TMLE: double cross-fit TMLE

Simulations

As expected, the performance of correctly specified parametric nuisance models for all ACE estimators were unbiased and confidence intervals resulted in near-95% coverage (Figure 2, Table 3). The most efficient estimator was g-formula (ESE=0.017), followed by TMLE (ESE=0.021), AIPW (ESE=0.021), and IPTW (ESE=0.023). DC-TMLE and DC-AIPW were comparable to their non cross-fit counterparts (0.021 and 0.021, respectively). Confidence interval coverage was higher for double cross-fit estimators.

For main-effects parametric nuisance models, all ACE estimators were biased from the true target parameter. Increased RMSE was primarily a result of the occurrence of bias. The double cross-fit procedure did not improve estimates in terms of bias due to model misspecification. Confidence interval coverage is likely greater solely due to the penalty in estimated variance due to variation between sample splits.

For singly-robust estimators with machine learning, point estimates were biased (Table 3, Figure 2). Non-cross-fit doubly robust estimators with machine learning resulted in unbiased estimates of the ACE, but confidence interval coverage was below expected levels for AIPW (91.1%) and TMLE (89.5%). Confidence interval coverage of DC-AIPW and DC-TMLE were near nominal levels (95.6% and 95.0%, respectively).

DISCUSSION

Under the unlikely scenario in which parametric nuisance model specifications correctly capture the true density, the ACE estimates are unbiased and subsequent inference is valid for all estimators considered. In these scenarios, confidence intervals were wider for double cross-fit estimators due to the variance between splits being incorporated from the sample splitting procedure. This increase in variance highlights the bias-variance trade-off made when choosing a less-restrictive estimator. However, it is often unreasonable to assume correct parametric model specification in high-dimensional data with weak background information or theory. In these scenarios, the pursuit of weaker parametric assumptions for nuisance model specification is worthwhile, with machine learning being a viable approach. However, nave use of machine learning may lead to bias and incorrect inference. As highlighted in our

Table 3: Simulation results for estimators under different approaches to estimation of the nuisance functions

	Bias	RMSE	ASE	ESE	CLD	Coverage
G-formula						
True	0.000	0.017	0.017	0.017	0.065	93.5%
Main-effects	-0.023	0.029	0.017	0.018	0.067	72.3%
Machine learning	0.026	0.031	0.015	0.017	0.058	56.5%
IPTW						
True	0.007	0.025	0.025	0.024	0.097	94.9%
Main-effects	-0.022	0.032	0.023	0.023	0.091	86.6%
Machine learning	0.010	0.023	0.023	0.021	0.090	94.8%
AIPW						
True	0.000	0.021	0.020	0.021	0.077	93.9%
Main-effects	-0.016	0.026	0.020	0.020	0.076	84.4%
Machine learning	0.004	0.020	0.017	0.019	0.066	91.3%
TMLE						
True	0.000	0.021	0.020	0.021	0.077	93.6%
Main-effects	-0.017	0.025	0.019	0.018	0.075	84.9%
Machine learning	-0.002	0.020	0.017	0.020	0.065	89.5%
DC-AIPW						
True	0.000	0.021	0.022	0.021	0.084	95.2%
Main-effects	-0.015	0.026	0.025	0.022	0.096	90.9%
Machine learning	-0.001	0.020	0.020	0.020	0.080	95.6%
DC-TMLE						
True	0.001	0.020	0.021	0.020	0.083	95.6%
Main-effects	-0.018	0.025	0.022	0.018	0.088	89.5%
Machine learning	0.000	0.020	0.020	0.020	0.077	95.0%

RMSE: root mean squared error, ASE: average standard error, ESE: empirical standard error, CLD: confidence limit difference, Coverage: 95% confidence limit coverage of the true value.
IPTW: inverse probability of treatment weights, AIPW: augmented inverse probability of treatment weights, TMLE: targeted maximum likelihood estimator, DC-AIPW: double cross-fit AIPW, DC-TMLE: double cross-fit TMLE.
True: correct model specification. Main-effects: all variables were assumed to be linearly related to the outcome and no interaction terms were included in the model. Machine learning: super-learner with 10-fold cross-validation including empirical mean, main-effects logistic regression without regularization, generalized additive models, random forest, and a neural network.

simulation, doubly robust estimators with double cross-fit and machine learning outperformed both estimators with incorrectly specified parametric nuisance models and non-cross-fit estimators with machine learning. As such, double cross-fit estimators with machine learning may be the preferred approach for ACE estimation in most epidemiologic studies.

Cross-fitting has had a long history in statistics [30, 31, 32], but recent emphasis has focused on its use for nonparametric nuisance function estimation [10, 11, 33, 34]. Broadly, double cross-fit procedures can be seen as an approach to avoid the overfitting of nuisance models. Single cross-fit procedures uncouple the nuisance model estimation from the corresponding predicted values, preventing so-called own observation bias [11]. However, the treatment nuisance model and outcome nuisance model are estimated using the same data in single cross-fit procedures. Double cross-fit procedures decouple these nuisance models by using separate splits, removing so-called nonlinearity bias [11]. As demonstrated in the simulations, reductions of these bias terms results in tangible benefits regarding point estimation and inference with machine learning algorithms.

While cross-fitting has tangible benefits, these benefits are not without cost. First, run-times for the double cross-fit estimators are substantially longer due to the repetition of fitting algorithms to a variety of different splits. We note that the double cross-fit procedure can easily be made to run in parallel, substantially reducing run-times. Computational costs may limit cross-fit procedures to estimators with closed-form variances, since bootstrapping would require considerable computational resources. A second, and potentially more problematic, cost is that sample splitting procedures reduce the amount of data available with which to fit algorithms. Because of the reduction in available data in each sample split, some complex algorithms may be precluded from use. The problem of finite data is furthered with the use of k-fold super-learner, further stretching the available data. For small data sets, increasing the number of folds in super-learner may aid in alleviating this issue [35]. For cross-fitting, single cross-fit procedures may be used. Instead of splitting into three samples, a single cross-fit at minimum requires splitting the data in half. However, the machine learning advantage of flexibility may be limited in these small data sets to begin with [9]. Whether single cross-fit with machine learning or highly flexible parametric models is preferred in these scenarios is an area for future work.

The problems of sample splitting can manifest themselves as random violations of the positivity assumption [36]. As detailed in previous work by Yu et al. 2019, confounders that are strongly related to the exposure may result in positivity violations [37]. Due to the flexibility of machine learning algorithms, these positivity issues may result in highly variable estimates. Furthermore, positivity issues may not be easy to diagnose, especially in procedures like double cross-fitting. Similar to previous recommendations [37], using multiple approaches to triangulate estimates may be helpful. For example, researchers may want to compare a flexible parametric AIPW and a double cross-fit AIPW with super-learner.

While our results support the use of machine learning algorithms, machine learning is not panacea for causal inference. Rather, machine learning can be seen as weakening a single assumption, namely the assumption of proper model specification. Prior substantive knowledge to justify counterfactual consistency, conditional exchangeability, and positivity remain necessary for causal inference [9, 35]. For super learner and other ensemble approaches to provide the maximal benefit in terms of specification, a diverse set of algorithms should be included [21]. Furthermore, multiple tuning parameters, sometimes referred to as hyperparameters, should be included. While the program defaults are often used, these hyperparameters can dramatically change performance of algorithms [38]. Therefore, super learner should not only include a diverse set of algorithms, but also those same algorithms under a diverse set of hyperparameters. Our simulations did not extensively explore hyperparameters; with the inclusion of only two hyperparameter specifications for generalized additive models. Because double cross-fit procedures scale poorly in terms of run-time with the addition of algorithms, including more algorithms with different hyperparameters can have substantial cost in terms of run-time. Depending on the complexity of machine learning algorithms used, alternative approaches may be required for hyper-parameter tuning within the cross-fitting procedure [39]. Despite these concerns, a wide variety of hyperparameters should be explored in applications of double cross-fitting. Lastly,

data processing, such as calculation of interaction terms between variables, may be necessary for performance and should be done in practice [40].

Future work should include diagnostics for double cross-fitting and the addition of other nuisance functions. Due to the repetition of sample splitting, standard diagnostics (e.g. examining the distributions of predicted treatment probabilities [37]) may more difficult to interpret. Often, realistic analyses further have additional issues that must be addressed, such as missing data and loss-to-follow-up. Therefore, additional nuisance functions (like inverse probability weights for informative loss-to-follow-up) are often needed and cross-fit procedures for these scenarios should be assessed.

Conclusions

Machine learning is not a panacea for the monumental task of causal inference. However, these algorithms do impose less restrictive assumptions regarding the possible forms of the nuisance functions used for estimation. Double cross-fit estimators should be seen as an approach to flexibly fitting nuisance models while retaining valid inference. In practice, double cross-fit estimators should be used regularly with a super-learner that includes a diverse library of learners.

Acknowledgments

References

[1] E. H. Kennedy, "Semiparametric theory and empirical processes in causal inference," in *Statistical causal inferences and their applications in public health research*, pp. 141–167, Springer, 2016.

[2] S. J. Mooney and V. Pejaver, "Big data in public health: terminology, machine learning, and privacy," *Annual review of public health*, vol. 39, pp. 95–112, 2018.

[3] Q. Bi, K. E. Goodman, J. Kaminsky, and J. Lessler, "What is machine learning? a primer for the epidemiologist," *American journal of epidemiology*, 2019.

[4] M. S. Schuler and S. Rose, "Targeted maximum likelihood estimation for causal inference in observational studies," *American journal of epidemiology*, vol. 185, no. 1, pp. 65–73, 2017.

[5] S. Watkins, M. Jonsson-Funk, M. A. Brookhart, S. A. Rosenberg, T. M. O'Shea, and J. Daniels, "An empirical comparison of tree-based methods for propensity score estimation," *Health services research*, vol. 48, no. 5, pp. 1798–1817, 2013.

[6] R. Pirracchio, M. L. Petersen, and M. van der Laan, "Improving propensity score estimators' robustness to model misspecification using super learner," *American journal of epidemiology*, vol. 181, no. 2, pp. 108–119, 2015.

[7] B. K. Lee, J. Lessler, and E. A. Stuart, "Improving propensity score weighting using machine learning," *Statistics in medicine*, vol. 29, no. 3, pp. 337–346, 2010.

[8] D. Westreich, J. Lessler, and M. J. Funk, "Propensity score estimation: neural networks, support vector machines, decision trees (cart), and meta-classifiers as alternatives to logistic regression," *Journal of clinical epidemiology*, vol. 63, no. 8, pp. 826–833, 2010.

[9] A. P. Keil and J. K. Edwards, "You are smarter than you think:(super) machine learning in context," *European journal of epidemiology*, vol. 33, no. 5, pp. 437–440, 2018.

[10] V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins, "Double/debiased machine learning for treatment and structural parameters," *The Econometrics Journal*, vol. 21, pp. C1–C68, 01 2018.

[11] W. K. Newey and J. R. Robins, "Cross-fitting and fast remainder rates for semiparametric estimation," *arXiv preprint arXiv:1801.09138*, 2018.

[12] W. Zheng and M. J. van der Laan, "Cross-validated targeted minimum-loss-based estimation," in *Targeted Learning*, pp. 459–474, Springer, 2011.

[13] S. Athey and G. Imbens, "Recursive partitioning for heterogeneous causal effects," *arXiv preprint arXiv:1504.01132*, 2015.

[14] S. R. Cole and C. E. Frangakis, "The consistency statement in causal inference: a definition or an assumption?," *Epidemiology*, vol. 20, no. 1, pp. 3–5, 2009.

[15] M. A. Hernán and J. M. Robins, "Estimating causal effects from epidemiological data," *Journal of Epidemiology & Community Health*, vol. 60, no. 7, pp. 578–586, 2006.

[16] D. Westreich and S. R. Cole, "Invited commentary: positivity in practice," *American journal of epidemiology*, vol. 171, no. 6, pp. 674–677, 2010.

[17] S. M. Grundy, N. J. Stone, A. L. Bailey, C. Beam, K. K. Birtcher, R. S. Blumenthal, L. T. Braun, S. de Ferranti, J. Faiella-Tommasino, D. E. Forman, *et al.*, "2018 aha/acc/aacvpr/aapa/abc/acpm/ada/ags/apha/aspc/nla/pcna guideline on the management of blood cholesterol: a report of the american college of cardiology/american heart association task force on clinical practice guidelines," *Journal of the American College of Cardiology*, vol. 73, no. 24, pp. e285–e350, 2019.

[18] J. A. Salami, H. Warraich, J. Valero-Elizondo, E. S. Spatz, N. R. Desai, J. S. Rana, S. S. Virani, R. Blankstein, A. Khera, M. J. Blaha, *et al.*, "National trends in statin use and expenditures in the us adult population from 2002 to 2013: insights from the medical expenditure panel survey," *JAMA cardiology*, vol. 2, no. 1, pp. 56–65, 2017.

[19] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural computation*, vol. 8, no. 7, pp. 1341–1390, 1996.

[20] M. J. Van der Laan, E. C. Polley, and A. E. Hubbard, "Super learner," *Statistical applications in genetics and molecular biology*, vol. 6, no. 1, 2007.

[21] S. Rose, "Mortality risk score prediction in an elderly population using machine learning," *American journal of epidemiology*, vol. 177, no. 5, pp. 443–452, 2013.

[22] T. J. Hastie, "Generalized additive models," in *Statistical models in S*, pp. 249–307, Routledge, 2017.

[23] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] M. J. Funk, D. Westreich, C. Wiesen, T. Stürmer, M. A. Brookhart, and M. Davidian, "Doubly robust estimation of causal effects," *American journal of epidemiology*, vol. 173, no. 7, pp. 761–767, 2011.

[25] J. K. Lunceford and M. Davidian, "Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study," *Statistics in medicine*, vol. 23, no. 19, pp. 2937–2960, 2004.

[26] H. Bang and J. M. Robins, "Doubly robust estimation in missing data and causal inference models," *Biometrics*, vol. 61, no. 4, pp. 962–973, 2005.

[27] J. M. Snowden, S. Rose, and K. M. Mortimer, "Implementation of g-computation on a simulated data set: demonstration of a causal inference technique," *American journal of epidemiology*, vol. 173, no. 7, pp. 731–738, 2011.

[28] M. A. Hernán, B. A. Brumback, and J. M. Robins, "Estimating the causal effect of zidovudine on cd4 count with a marginal structural model for repeated measures," *Statistics in medicine*, vol. 21, no. 12, pp. 1689–1709, 2002.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[30] P. J. Bickel and Y. Ritov, "Estimating integrated squared density derivatives: sharp best order of convergence estimates," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 381–393, 1988.

[31] J. Pfanzagl, "Estimation in semiparametric models," in *Estimation in Semiparametric Models*, pp. 17–22, Springer, 1990.

[32] J. Hájek, "Asymptotically most powerful rank-order tests," *The Annals of Mathematical Statistics*, pp. 1124–1147, 1962.

[33] J. Robins, L. Li, E. Tchetgen, A. van der Vaart, *et al.*, "Higher order influence functions and minimax estimation of nonlinear functionals," in *Probability and statistics: essays in honor of David A. Freedman*, pp. 335–421, Institute of Mathematical Statistics, 2008.

[34] P. J. Bickel, "On adaptive estimation," *The Annals of Statistics*, pp. 647–671, 1982.

[35] A. I. Naimi and L. B. Balzer, "Stacked generalization: an introduction to super learning," *European journal of epidemiology*, vol. 33, no. 5, pp. 459–464, 2018.

[36] M. L. Petersen, K. E. Porter, S. Gruber, Y. Wang, and M. J. Van Der Laan, "Diagnosing and responding to violations in the positivity assumption," *Statistical methods in medical research*, vol. 21, no. 1, pp. 31–54, 2012.

[37] Y.-H. Yu, L. M. Bodnar, M. M. Brooks, K. P. Himes, and A. I. Naimi, "Comparison of parametric and nonparametric estimators for the association between incident prepregnancy obesity and stillbirth in a population-based cohort study," *American journal of epidemiology*, vol. 188, no. 7, pp. 1328–1336, 2019.

[38] F. Dominici, A. McDermott, S. L. Zeger, and J. M. Samet, "On the use of generalized additive models in time-series studies of air pollution and health," *American journal of epidemiology*, vol. 156, no. 3, pp. 193–203, 2002.

[39] J. Wong, T. Manderson, M. Abrahamowicz, D. L. Buckeridge, and R. Tamblyn, "Can hyperparameter tuning improve the performance of a super learner?: A case study," *Epidemiology (Cambridge, Mass.)*, vol. 30, no. 4, p. 521, 2019.

[40] A. I. Naimi and E. H. Kennedy, "Nonparametric double robustness," *arXiv preprint arXiv:1711.07137*, 2017.

Appendix

Section 1: Data generating mechanism

Let A indicate age, L indicate the natural-log transformed low-density lipo-protein, D indicate diabetes, F indicate frailty, and R indicate the risk score. All observations are indexed by i, such that A_i refers to the age of individual i. All variables were generated from the following distributions in their respective order.

$$A_i = \begin{cases} 75 - \sqrt{30(v_i - 60)} & \text{if } v_i > 60 \\ v_i & \text{if } v_i \le 60 \end{cases}, \quad v = \frac{55 \times \text{Uniform}(0,1) + 80}{2}$$

$$L_i = 0.005 A_i + \text{Normal}(\log(100), 0.18)$$

$$D_i = \text{Bernoulli}(\text{expit}(-4.23 + 0.03 L_i - 0.02 A_i + 0.0009 A_i^2))$$

$$F_i = \text{expit}(-5.5 + 0.05(A_i - 20) + 0.001 A_i^2 + \text{Normal}(0,1)$$

$$R_i = \text{expit}(4.299 + 3.501 D_i - 2.07 \log(A_i) + 0.051 \log(A_i)^2 + 4.090 L_i - 1.04 L_i \log(A_i) + 0.01 F_i)$$

where $\text{expit}(x) = \frac{\exp(x)}{1+\exp(x)}$. Statin (X) and atherosclerotic cardiovascular disease (Y) were generated from the following models

$$\begin{aligned} \Pr(X = 1|Z) = \text{Bernoulli}(\text{expit}(& -3.471 + 1.390 D_i + 0.112 L_i + 0.973 \, I(L_i > \log(60)) \\ & - 0.046(A_i - 30) + 0.003(A_i - 30)^2 + 0.273 \, I(0.05 \le R_i < 0.075) \\ & + 1.592 \, I(0.075 \le R_i < 0.2) + 2.461 \, I(R_i \ge 0.2))) \end{aligned}$$

$$\begin{aligned} \Pr(Y^x = 1|Z) = \text{Bernoulli}(\text{expit}(& -6.250 - 0.75x + 0.35x(5 - L_i) \, I(L_i < \log(130)) + 0.45(A_i - 39)^{0.5} \\ & + 1.75 D_i + 0.29 \exp(R_i + 1) + 0.14 L_i^2 \, I(L_i > \log(120)))) \end{aligned}$$

From each of the potential outcomes, the observed outcomes were calculated based on the potential outcome under the observed treatment.

$$Y_i = X_i Y_i^1 + (1 - X_i) Y_i^0$$

Across all simulations the estimand of interest, the population average causal effect, was defined as

$$\psi = E[Y^1 - Y^0]$$

The true value for ψ was calculated directly from the potential outcomes of 10,000,000 individuals. Let $\hat{\psi}$ indicate the sample-specific estimate of the population average causal effect.

Section 2: Estimators

We consider two nuisance models, where $m(Z_i; \beta, \eta)$ is the outcome nuisance model and $\pi(Z_i; \gamma)$ is the treatment nuisance model. The estimated outcome nuisance model is expressed as:

$$m_x(Z_i; \hat{\beta}, \hat{\eta}) = \hat{E}[Y|X = x, Z] = \hat{\beta} x + \hat{\eta} Z_i$$

and the estimated treatment nuisance model is expressed as:

$$\pi(Z_i; \hat{\gamma}) = \widehat{\Pr}(X = 1|Z) = \hat{\gamma} Z_i$$

G-formula

The g-computation algorithm consists of the outcome nuisance model predictions only. The estimated parameters of the outcome nuisance model are used to predict outcome values. The average of the predicted outcome values under each treatment plan are contrasted. The average causal effect is estimated via:

$$\hat{\psi}_G = n^{-1} \sum_i m_1(Z_i; \hat{\beta}, \hat{\eta}) - n^{-1} \sum_i m_0(Z_i; \hat{\beta}, \hat{\eta})$$

Wald-type confidence intervals were calculated with the standard error estimated using a bootstrapping procedure. Briefly, n samples are drawn with replacement from the observed sample. Nuisance models are fit to the newly generated sample and used to calculate the average causal effect. This procedure was repeated 250 times, with the bootstrap standard error defined as the standard error of the 250 re-estimated average causal effects.

Inverse probability weighed estimator

IPW consists of the treatment nuisance model predictions only. The estimated parameters of the treatment nuisance model are used to predict probabilities of treatment. The inverse of these estimated probabilities are used to construct the inverse probability weights. The average causal effect is estimated via:

$$\hat{\psi}_{IPTW} = n^{-1} \sum_i \frac{Y_i X_i}{\pi(Z_i; \hat{\gamma})} - n^{-1} \sum_i \frac{Y_i(1 - X_i)}{1 - \pi(Z_i; \hat{\gamma})}$$

Confidence intervals can similarly be calculated using a bootstrapping procedure. Instead, we used a robust variance estimator, which ignores the estimation of the nuisance model. Confidence intervals constructed from the robust variance result in conservative variance estimates for the ACE, with coverage expected to be at least 95% when the nuisance model is properly specified.

Augmented inverse probability weights

AIPW consists of the treatment and outcome nuisance model predictions. These predictions are combined via the following formula to estimate the average causal effect:

$$
\begin{aligned}
\hat{\psi}_{AIPW} = & n^{-1} \sum_i \frac{Y_i X_i}{\pi(Z_i; \hat{\gamma})} + \frac{m_1(Z_i; \hat{\beta}, \hat{\eta})(\pi(Z_i; \hat{\gamma}) - X_i)}{\pi(Z_i; \hat{\gamma})} \\
& - n^{-1} \sum_i \frac{Y_i(1 - X_i)}{1 - \pi(Z_i; \hat{\gamma})} + \frac{m_0(Z_i; \hat{\beta}, \hat{\eta})(X_i - \pi(Z_i; \hat{\gamma}))}{1 - \pi(Z_i; \hat{\gamma})}
\end{aligned}
\tag{3}
$$

As indicated by their name, the AIPW estimator can be seen to include the IPW and an augmentation term. For inference, Wald-type confidence intervals were constructed from the following variance empirical influence curve-based estimator

$$
\begin{aligned}
\widehat{Var}(\hat{\psi}_{AIPW}) = & \frac{1}{n-1} \sum_i \left(\left(\frac{Y_i X_i}{\pi(Z_i; \hat{\gamma})} + \frac{m_1(Z_i; \hat{\beta}, \hat{\eta})(\pi(Z_i; \hat{\gamma}) - X_i)}{\pi(Z_i; \hat{\gamma})} \right) \right. \\
& \left. - \left(\frac{Y_i(1 - X_i)}{1 - \pi(Z_i; \hat{\gamma})} + \frac{m_0(Z_i; \hat{\beta}, \hat{\eta})(X_i - \pi(Z_i; \hat{\gamma}))}{1 - \pi(Z_i; \hat{\gamma})} \right) - \hat{\psi}_{AIPW} \right)^2
\end{aligned}
$$

This estimated variance is directly used to calculate Wald-type confidence intervals via

$$\hat{\psi}_{AIPW} \pm 1.96 \sqrt{\frac{\widehat{Var}(\hat{\psi}_{AIPW})}{n}}$$

Targeted maximum likelihood estimation

TMLE similar consists of the treatment and outcome nuisance models. For ease of later notation, we define the so-called clever covariate as:

$$H(X_i, Z_i; \hat{\gamma}) = \frac{X_i}{\pi(Z_i; \hat{\gamma})} - \frac{1 - X_i}{1 - \pi(Z_i; \hat{\gamma})}$$

Using the clever covariate, we target (or update) the predictions from the outcome nuisance model. The targeting step is done by first estimating ϵ in the following parametric working model

$$\text{logit}(Y_i) = \text{logit}(m_X(Z_i; \hat{\beta}, \hat{\eta})) + \epsilon H(X_i, Z_i; \hat{\gamma})$$

The estimate $\hat{\epsilon}$ is then used to update the untargeted estimates via the following formulas:

$$m_1^*(m_1, H(1, Z_i; \hat{\gamma}), \hat{\epsilon}) = \text{expit}\left(\text{logit}(m_1(Z_i; \hat{\beta}, \hat{\eta}) + \hat{\epsilon}\, H(1, Z_i; \hat{\gamma}))\right)$$

$$m_0^*(m_0, H(0, Z_i; \hat{\gamma}), \hat{\epsilon}) = \text{expit}\left(\text{logit}(m_0(Z_i; \hat{\beta}, \hat{\eta}) + \hat{\epsilon}\, H(0, Z_i; \hat{\gamma}))\right)$$

Therefore, we can use the targeted model predictions to calculate the average causal effect as:

$$\hat{\psi}_{TMLE} = \frac{1}{n} \sum_i m_1^*(m_1, H(1, Z_i; \hat{\gamma}), \hat{\epsilon}) - \frac{1}{n} \sum_i m_0^*(m_0, H(0, Z_i; \hat{\gamma}), \hat{\epsilon})$$

For inference, Wald-type confidence intervals were constructed from the following variance estimator:

$$\widehat{Var}(\hat{\psi}_{TMLE}) = \frac{1}{n-1} \sum_i \left(\left(\frac{Y_i X_i}{\pi(Z_i; \hat{\gamma})} + \frac{m_1^*(m_1, H(1, Z_i; \hat{\gamma}), \hat{\epsilon})(\pi(Z_i; \hat{\gamma}) - X_i)}{\pi(Z_i; \hat{\gamma})} \right) \right.$$
$$\left. - \left(\frac{Y_i(1 - X_i)}{1 - \pi(Z_i; \hat{\gamma})} + \frac{m_0^*(m_0, H(0, Z_i; \hat{\gamma}), \hat{\epsilon})(X_i - \pi(Z_i; \hat{\gamma}))}{1 - \pi(Z_i; \hat{\gamma})} \right) - \hat{\psi}_{TMLE} \right)^2$$