PLSC 40601

Week 7: Reviewing advances estimating techniques building on semiparemtric theory: DML/TMLE.

Molly Offer-Westort

Department of Political Science,
University of Chicago

Spring 2023

Housekeeping

- ?

Objective

- Get point estimates, confidence intervals for a (potentially causal) low-dimensional parameter θ_0 in the presence of high-dimensional nuisance parameters, η_0.

Objective

- Get point estimates, confidence intervals for a (potentially causal) low-dimensional parameter θ_0 in the presence of high-dimensional nuisance parameters, η_0.
- η_0 may be so high-dimensional that traditional constraints on complexity of the parameter space, e.g., that estimators $\hat{\eta}$ lie in a space constrained by Donsker conditions, break down.

Objective

- Get point estimates, confidence intervals for a (potentially causal) low-dimensional parameter θ_0 in the presence of high-dimensional nuisance parameters, η_0.

- η_0 may be so high-dimensional that traditional constraints on complexity of the parameter space, e.g., that estimators $\hat{\eta}$ lie in a space constrained by Donsker conditions, break down. (example: number of parameters grows with sample size)

Objective

- Get point estimates, confidence intervals for a (potentially causal) low-dimensional parameter θ_0 in the presence of high-dimensional nuisance parameters, η_0.
- η_0 may be so high-dimensional that traditional constraints on complexity of the parameter space, e.g., that estimators $\hat{\eta}$ lie in a space constrained by Donsker conditions, break down. (example: number of parameters grows with sample size)
- We would like to use machine learning methods to deal with the high-dimensional setting.

ML methods

- ML methods relax reliance on assumptions about functional form;

ML methods

- ML methods relax reliance on assumptions about functional form; particularly useful with high dimensional data based on e.g., interactions, where we might not have strong substantive motivations for specific form.

ML methods

- ML methods relax reliance on assumptions about functional form; particularly useful with high dimensional data based on e.g., interactions, where we might not have strong substantive motivations for specific form.

- ML methods good at using regularization to reduce variance

ML methods

- ML methods relax reliance on assumptions about functional form; particularly useful with high dimensional data based on e.g., interactions, where we might not have strong substantive motivations for specific form.

- ML methods good at using regularization to reduce variance $\rightarrow$ good performance on prediction.

ML methods

- ML methods relax reliance on assumptions about functional form; particularly useful with high dimensional data based on e.g., interactions, where we might not have strong substantive motivations for specific form.

- ML methods good at using regularization to reduce variance $\rightarrow$ good performance on prediction.

- Good prediction does not imply good performance for estimation and inference, particularly for causal parameters.

DML Solution

1. use Neyman-orthogonal scores that are less sensitive wrt nuisance parameters to estimate θ_0

DML Solution

1. use Neyman-orthogonal scores that are less sensitive wrt nuisance parameters to estimate θ_0
2. use cross-fitting to deal with additional bias from ML estimators

Overview of algorithm (DML2)

1. Partition data into K equal folds

Overview of algorithm (DML2)

1. Partition data into K equal folds (4-5 seems good).

Overview of algorithm (DML2)

1. Partition data into K equal folds (4-5 seems good).
2. Estimate nuisance parameters on each fold k, using appropriate ML estimators for the context.

Overview of algorithm (DML2)

1. Partition data into K equal folds (4-5 seems good).
2. Estimate nuisance parameters on each fold k, using appropriate ML estimators for the context. (lots of sparsity? $\rightarrow$ lasso. Well approximated by trees? $\rightarrow$ random forest. And/or use ensemble methods.)

Overview of algorithm (DML2)

1. Partition data into K equal folds (4-5 seems good).
2. Estimate nuisance parameters on each fold k, using appropriate ML estimators for the context. (lots of sparsity? $\rightarrow$ lasso. Well approximated by trees? $\rightarrow$ random forest. And/or use ensemble methods.)
3. Estimate $\hat{\theta}_0$ as the solution to:

$$\frac{1}{K}\sum_{k=1}^{K} \mathrm{E}_{n,k}[\psi(W, \hat{\theta}_0, \hat{\eta}_{0,k})] = 0$$

   where ψ is the Neyman orthogonal score, $\mathrm{E}_{n,k}$ is empirical expectation over k-th fold of the data, W is a random element that takes values in a measurable space.

TMLE Solution

DML approach might produce results that perform well asymptotically, but don’t have nice small sample properties.

TMLE Solution

DML approach might produce results that perform well asymptotically, but don't have nice small sample properties.

1. Estimate the density using some estimating model that will have nice behavior

TMLE Solution

DML approach might produce results that perform well asymptotically, but don't have nice small sample properties.

1. Estimate the density using some estimating model that will have nice behavior
2. Then update in the model space, avoiding predictions that will be outside the density of the data / parameter space.

Overview of algorithm (Focusing on ATE)

1. Split the data into folds.

Overview of algorithm (Focusing on ATE)

1. Split the data into folds. Within a given training + validation pair:

Overview of algorithm (Focusing on ATE)

1. Split the data into folds. Within a given training + validation pair:
2. Generate initial estimate of conditional mean outcome $\hat{Y}_A = \hat{\mathrm{E}}\left[Y|A, \mathbf{X}\right]$ on training data.

Overview of algorithm (Focusing on ATE)

1. Split the data into folds. Within a given training + validation pair:
2. Generate initial estimate of conditional mean outcome $\hat{Y}_A = \hat{\mathrm{E}}[Y|A, \mathbf{X}]$ on training data.
3. Generate estimate of exposure mechanism, $\hat{\pi}_a = \hat{P}[A = a|\mathbf{X}]$.

Overview of algorithm (Focusing on ATE)

1. Split the data into folds. Within a given training + validation pair:
2. Generate initial estimate of conditional mean outcome $\hat{Y}_A = \hat{\mathrm{E}}\left[Y|A, \mathbf{X}\right]$ on training data.
3. Generate estimate of exposure mechanism, $\hat{\pi}_a = \hat{P}[A = a|\mathbf{X}]$.
4. Update initial estimate of $\hat{\mathrm{E}}\left[Y|A, \mathbf{X}\right]$ on validation data.

Overview of algorithm (Focusing on ATE)

1. Split the data into folds. Within a given training + validation pair:
2. Generate initial estimate of conditional mean outcome $\hat{Y}_A = \hat{\mathrm{E}}[Y|A, \mathbf{X}]$ on training data.
3. Generate estimate of exposure mechanism, $\hat{\pi}_a = \hat{P}[A = a|\mathbf{X}]$.
4. Update initial estimate of $\hat{\mathrm{E}}[Y|A, \mathbf{X}]$ on validation data.
   - Calculate $H_a(A = a, \mathbf{X}) = \frac{\mathbb{I}(A=1)}{\hat{\pi}_1} - \frac{\mathbb{I}(A=0)}{\hat{\pi}_1}$

Overview of algorithm (Focusing on ATE)

1. Split the data into folds. Within a given training + validation pair:
2. Generate initial estimate of conditional mean outcome $\hat{Y}_A = \hat{\mathrm{E}}\,[Y|A, \mathbf{X}]$ on training data.
3. Generate estimate of exposure mechanism, $\hat{\pi}_a = \hat{P}[A = a|\mathbf{X}]$.
4. Update initial estimate of $\hat{\mathrm{E}}\,[Y|A, \mathbf{X}]$ on validation data.
   - Calculate $H_a(A = a, \mathbf{X}) = \frac{\mathbb{I}(A=1)}{\hat{\pi}_1} - \frac{\mathbb{I}(A=0)}{\hat{\pi}_1}$
   - Regress Y on H_a, specifying $\hat{Y}_a$ as offset.

Overview of algorithm (Focusing on ATE)

1. Split the data into folds. Within a given training + validation pair:
2. Generate initial estimate of conditional mean outcome $\hat{Y}_A = \hat{\mathrm{E}}[Y|A, \mathbf{X}]$ on training data.
3. Generate estimate of exposure mechanism, $\hat{\pi}_a = \hat{P}[A = a|\mathbf{X}]$.
4. Update initial estimate of $\hat{\mathrm{E}}[Y|A, \mathbf{X}]$ on validation data.
   - Calculate $H_a(A = a, \mathbf{X}) = \frac{\mathbb{I}(A=1)}{\hat{\pi}_1} - \frac{\mathbb{I}(A=0)}{\hat{\pi}_1}$
   - Regress Y on H_a, specifying $\hat{Y}_a$ as offset.
   - Generate counterfactual predictions from this new regression model, $\hat{Y}_1^*, \hat{Y}_0^*$.

Overview of algorithm (Focusing on ATE)

1. Split the data into folds. Within a given training + validation pair:
2. Generate initial estimate of conditional mean outcome $\hat{Y}_A = \hat{\mathrm{E}}[Y|A, \mathbf{X}]$ on training data.
3. Generate estimate of exposure mechanism, $\hat{\pi}_a = \hat{P}[A = a|\mathbf{X}]$.
4. Update initial estimate of $\hat{\mathrm{E}}[Y|A, \mathbf{X}]$ on validation data.
   - Calculate $H_a(A = a, \mathbf{X}) = \frac{\mathbb{I}(A=1)}{\hat{\pi}_1} - \frac{\mathbb{I}(A=0)}{\hat{\pi}_1}$
   - Regress Y on H_a, specifying $\hat{Y}_a$ as offset.
   - Generate counterfactual predictions from this new regression model, $\hat{Y}_1^*, \hat{Y}_0^*$.
5. Averaging across folds:

$$\hat{\theta}_{\text{CV-TMLE}} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{E}_{n,k} \left[\hat{Y}_1^{k*} - \hat{Y}_0^{k*} \right]$$

Overview of algorithm (Focusing on ATE)

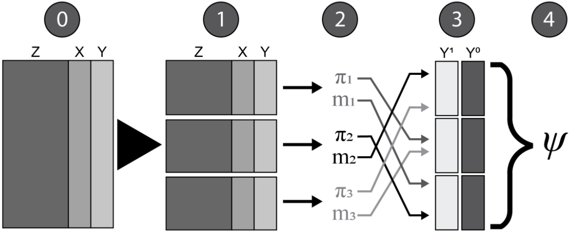
1. Split the data into folds. Within a given training + validation pair:
2. Generate initial estimate of conditional mean outcome $\hat{Y}_A = \hat{\mathrm{E}}[Y|A, \mathbf{X}]$ on training data.
3. Generate estimate of exposure mechanism, $\hat{\pi}_a = \hat{P}[A = a|\mathbf{X}]$.
4. Update initial estimate of $\hat{\mathrm{E}}[Y|A, \mathbf{X}]$ on validation data.
   - Calculate $H_a(A = a, \mathbf{X}) = \frac{\mathbb{I}(A=1)}{\hat{\pi}_1} - \frac{\mathbb{I}(A=0)}{\hat{\pi}_1}$
   - Regress Y on H_a, specifying $\hat{Y}_a$ as offset.
   - Generate counterfactual predictions from this new regression model, $\hat{Y}_1^*, \hat{Y}_0^*$.
5. Averaging across folds:

$$\hat{\theta}_{\text{CV-TMLE}} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{E}_{n,k}\left[\hat{Y}_1^{k*} - \hat{Y}_0^{k*}\right]$$

Technically you can do TMLE without machine learning OR crossfitting.

Cross-fitting

Figure 1: General double cross-fit procedure for doubly-robust estimators



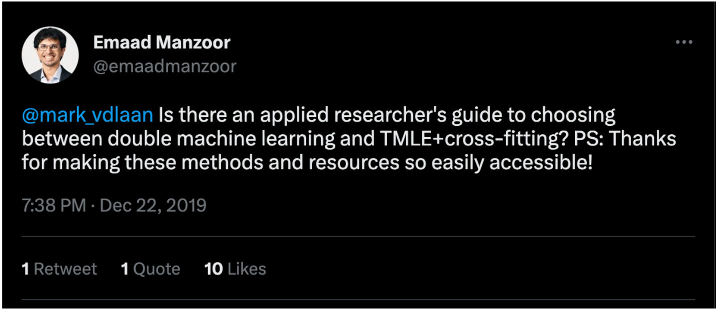
Step 0) The exposure (X), outcome (Y), and minimally sufficient adjustment set for identification (Z) are selected and collected.
Step 1) The data is partitioned into three approximately equal-sized sample splits.
Step 2) The treatment nuisance model and the outcome nuisance model are fit in each sample 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.

Zivich and Breskin (2021)

Cross-fitting

- As with DML, there are some different approaches to cross-fitting; e.g., updating step can be pooled across folds.

Relative benefits of TMLE vs. DML



Relative benefits of TMLE vs. DML

Response: https://vanderlaan-lab.org/2019/12/24/cv-tmle-and-double-machine-learning/

Relative benefits of TMLE vs. DML

Response: `https://vanderlaan-lab.org/2019/12/24/cv-tmle-and-double-machine-learning/`

Some arguments:

Relative benefits of TMLE vs. DML

Response: https://vanderlaan-lab.org/2019/12/24/cv-tmle-and-double-machine-learning/

Some arguments:

- TMLE "respects" constraints of the model, especially useful with, e.g., rare outcomes (Balzer et al., 2016)

Relative benefits of TMLE vs. DML

Response: `https://vanderlaan-lab.org/2019/12/24/cv-tmle-and-double-machine-learning/`

Some arguments:

- TMLE "respects" constraints of the model, especially useful with, e.g., rare outcomes (Balzer et al., 2016)

- No issues with multiple solutions

Relative benefits of TMLE vs. DML

Response: `https://vanderlaan-lab.org/2019/12/24/cv-tmle-and-double-machine-learning/`

Some arguments:

- TMLE "respects" constraints of the model, especially useful with, e.g., rare outcomes (Balzer et al., 2016)
- No issues with multiple solutions
- "We also note that the TMLE is the only estimator that actually generalizes the MLE – if the MLE is well-defined and used as initial estimator, then the TMLE is exactly equivalent to the MLE (i.e., the targeting step will select zero fluctuation)."

Application to data

Zivich and Breskin (2021)

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

September 1, 2020

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 complications for inference. Doubly-robust cross-fit estimators have been proposed to 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-computation, inverse probability weighting) and doubly-robust estimators (augmented inverse probability weighting, targeted maximum likelihood estimation). Nuisance functions were estimated with parametric models and ensemble machine learning, separately. We further assessed doubly-robust cross-fit estimators.

With correctly specified parametric models, all of the estimators were unbiased and confidence intervals achieved nominal coverage. When used with machine learning, the doubly-robust cross-fit estimators substantially outperformed all of the other estimators in terms of bias, variance, and 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.

Application to data

Zivich and Breskin (2021)

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

	Bias	RMSE	ASE	ESE	CLD	Coverage
G-computation						
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%
IPW						
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	0.085	95.2%
Main-effects	-0.015	0.026	0.027	0.022	0.106	92.4%
Machine learning	-0.001	0.020	0.021	0.020	0.082	95.6%
DC-TMLE						
True	0.001	0.020	0.021	0.020	0.084	95.8%
Main-effects	-0.018	0.025	0.024	0.018	0.094	91.4%
Machine learning	0.000	0.020	0.020	0.020	0.079	95.2%

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.
IPW: 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.

The goal

- Estimation of a causal parameter

The goal

- Estimation of a causal parameter
  - unbiased

The goal

- Estimation of a causal parameter
    - unbiased
    - consistent

The goal

- Estimation of a causal parameter
    - unbiased
    - consistent

- Possible sources of bias

The goal

- Estimation of a causal parameter
  - unbiased
  - consistent

- Possible sources of bias
  - identification
  - estimation

The goal

- Estimation of a causal parameter
  - unbiased
  - consistent

- Possible sources of bias
  - identification
  - estimation

- Possible uses of covariates

The goal

- Estimation of a causal parameter
    - unbiased
    - consistent
- Possible sources of bias
    - identification
    - estimation
- Possible uses of covariates
    - Required for causal model
    - Covariate adjustment for precision

References I

Balzer, L., Ahern, J., Galea, S., and van der Laan, M. (2016). Estimating effects with rare outcomes and high dimensional covariates: knowledge is power. *Epidemiologic methods*, 5(1):1–18.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters: Double/debiased machine learning. *The Econometrics Journal*, 21(1).

Schuler, M. S. and Rose, S. (2017). Targeted maximum likelihood estimation for causal inference in observational studies. *American Journal of Epidemiology*, 185(1):65–73.

Zivich, P. N. and Breskin, A. (2021). Machine learning for causal inference: on the use of cross-fit estimators. *Epidemiology (Cambridge, Mass.)*, 32(3):393.