Does sample splitting help when estimating causal effects using double machine learning methods?

TJ Tang, Qin He

2022-04-24

1. Introduction

Athey and Imbens (2016) proposed an "honest" approach to estimate causal effects using classification and regression trees (CART) that separate samples used for constructing the partition and estimating effects within leaves of the partition. Their study shows that honest estimation can remove the bias and improve in the coverage of confidence intervals. The intuition behind the honest criterion is to avoid using same data or information twice for different tasks, more specifically, to select a model structure and to make estimation given a mode structure (Athey and Imbens 2016). Sample splitting follows the honesty criterion and is indeed one of the key issues when adapting machine learning methods to causal inference problems. However, some anecdotes saying that sample splitting will not help in finite samples. Therefore, this paper study will first go through Chernozhukov et al. (2018) especially about an example that sample splitting reduces the bias, and then will conduct a few simulation studies to figure out the effect of sample splitting when estimating causal effects.

2. Double Machine Learning and Sample Splitting

Chernozhukov et al. (2018) establish that in very high dimension cases, machine learning methods will employ regularization to reduce variance but cause the regularization bias and overfitting on the parameters of interest or causal treatment effect parameter, for example average treatment effect. They show that the bias and overfit can be removed by double machine learning methods with sample splitting and cross fitting. Applying double machine learning methods can overcome the regularization bias because it employs an orthogonalized formulation. Meanwhile, sample splitting and cross fitting can reduce the bias due to the overfitting because we can avoid using the same data twice, to train the estimator and predict the missing potential outcomes. The details for orthogonalization and why double machine learning methods reduce the regularization bias is not the focus of this project. Instead, this project will focus more on the sample splitting and the effect to reduce overfitting bias.

Wager and Athey (2018) illustrated double-sample trees as a motivation to reduce bias. They divided training subsamples into two halves and used one half for split selection and the other for within-leaf estimation. Though sampling splitting is sometimes criticized as inefficient as they do not use the full training data at each estimation steps; however, they argued that by aggregating over multiple trees, each sample will be used for split selection

or leaf estimation. By doing so, therefore, we can efficiently achieve honesty criterion without wasting half of the samples.

Wager and Athey's ideas are similar to sample splitting and cross fitting in double machine learning. To perform the sample splitting to control for bias, we need to first split data into main sample and auxiliary sample, and then use the main sample to train and the auxiliary sample to estimate. Next, we swap the role of main sample and auxiliary sample to re-estimate. Finally, we combine the results by averaging over the two estimators. By doing so, we can achieve a good estimator without losing efficiency. The idea of sample splitting is an analogy to K-fold cross validation and can be also generalized to K-fold splitting and cross-fitting. For K-fold cases, we take a K-fold random partition of data rather than splitting into half, and at the end we aggregate over K machine learning estimators so that we use the full data. In terms of the choice of K, Chernozhukov et al. (2018) recommends using moderate K values such as 4 or 5.

3. Set up of Double Machine Learning with Partial Linear Model

Consider the partial linear model (using the notations in Chernozhukov et al. (2018):

$$Y = D\theta_0 + g_0(x) + U, \qquad E[U|X, D] = 0,$$
$$D = m_0(X) + V, \qquad E[V|X] = 0,$$

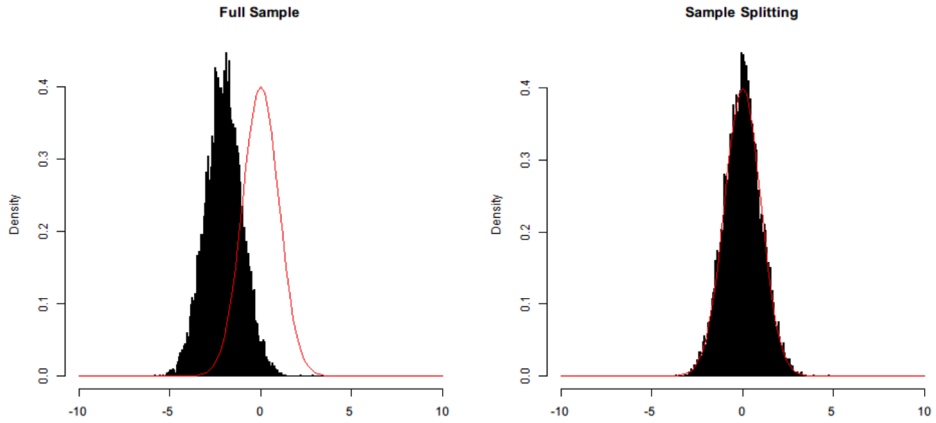
where Y is the outcome variable, D is the treatment indicator, X is a vector of confounders, U and V are disturbances, and θ_0 is the causal estimand we would like to infer.

Next, consider the double machine learning procedure with sample splitting, following Chernozhukov et al. (2018):
- Take a K-fold random partition $(I_k)_{k=1}^K$ of observation indices $[N] = \{1, \cdots, N\}$ such that the size of each fold I_k is N/K. And define $I_k^c = \{1, \cdots, N\} \setminus I_k$ for each k.
- For $k = 1$, using machine learning methods to estimate $\widehat{g_0}$ and $\widehat{m_0}$ using samples $I_{k=1}^c$.
- Predict Y and D using X by $\mathbb{E}[\widehat{Y|X}])$ and $\mathbb{E}[\widehat{D|X}])$ respectively on $I_{k=1}$.
- Get residuals $\widehat{U} = Y - \mathbb{E}[\widehat{Y|X}])$ and $\widehat{V} = D - \mathbb{E}[\widehat{D|X}])$
- Regress $\widehat{U}$ on $\widehat{V}$ to get the estimate $\widehat{\theta_{0,k=1}}$ using samples $I_{k=1}$.
- Repeat, obtain the K estimators for $k = 1, \cdots, K$, average them, and get $\widehat{\theta_0} = \frac{1}{K}\sum_{k=1}^K \widehat{\theta_{0,k}}$

Following the procedures, the estimator $\widehat{\theta_0}$ will be a consistent and asymptotic normal estimator. In other words, the distribution of $\widehat{\theta_0} - \theta_0$ will follow a normal distribution and center around zero.

4. Replicate Chernozhukov et al. (2018)

We found the codes for generating the Figure 2 in Chernozhukov et al. (2018) on GitHub. The purpose of the figures is to compare full sample with sample splitting and cross fitting procedures. We replicated and rewrote the codes in R and generated a similar plot.



To illustrate that sample splitting can combat overfitting bias, Chernozhukov et al. (2018) uses an artificial example. Let $X \sim N(0,1)$ and fix true θ_0 be 1. And the data is generated by $Y = D + X + U$ and $D = X + V$, where U and V are error terms and follow $N \sim (0,1)$. The overfit is specified as $(y_i - x_i)/N^{1/2-\epsilon}$. More specifically, $\hat{x}_i = x_i + (y_i - x_i)/N^{1/2-\epsilon}$. Then two estimators $\widehat{\theta_0}$ are calculated, one using the full sample, and the other adopting two-fold sample splitting and cross fitting. The double machine learning estimator is formulated as

$$\widehat{\theta_0} = \left(\frac{1}{n}\sum_{i\in I} \hat{V}_i D_i\right)^{-1} \frac{1}{n}\sum_{i\in I} \hat{V}_i (Y_i - \widehat{g_0}(X_i)).$$

Note that, when we use two-fold sample splitting and cross-fitting and if we partition our data into I and I^C, then $\widehat{\theta_0}$ for I is calculated using the $\widehat{g_0}$ estimated with observations in I^C, and vice versa. And we simply average over $\widehat{\theta_0}$ for I and $\widehat{\theta_0}$ for I^C to get our causal estimand using the full data. For the above plot, we fix $\epsilon = 0.1$. The histograms represent the distribution of the studentized $\widehat{\theta_0}$ calculated as $\frac{\widehat{\theta_0} - \theta_0}{s.e.(\widehat{\theta_0})}$, which can reflect the bias. And the red curve is the density for $N \sim (0,1)$. From the plots, we may observe that using the full sample and without sample splitting (left) clearly induce large bias as the distribution is shifted to the left and not centered around 0; however sample splitting with cross fitting (right) can remove the bias very well.

5. Simulation Study Setup

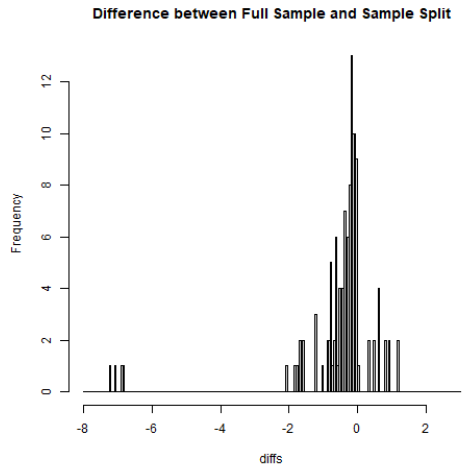
We explored 128 different scenarios in the data generation process with following configurations.

- Number of covariates: 5 or 100
- Sample size N: 50 or 1000
- Number of folds K: 2, 5
- Standard deviation of the error term U and V: 0.1 or 1
- Distribution of coefficients for covariates g_0: $N \sim (1, 0.5^2)$ or $U \sim (0.5,1.5)$
- Outcome model: linear or non-linear with sine functions

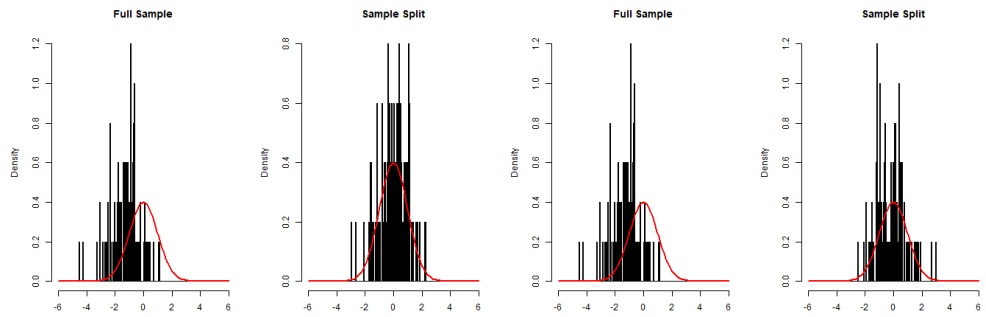
We fixed the true causal effect θ_0 be 1. And the machine learning methods we considered are LASSO and Extreme Gradient Boosting (XGBoost). We fit LASSO using `glmnet` package and the value of λ for LASSO is determined by 5-fold cross validation, where we compared λ from 0.001 to 100. We fit XGBoost using `xgboost` package. We did not tune XGBoost because we found the loss is small using the parameters provided in the documentation. For each scenario, we generated 100 different data sets and conduct double machine learning using the partial linear models with sample splitting and without sample splitting. The codes for simulations and the simulation results are provided.

6. Simulation Results

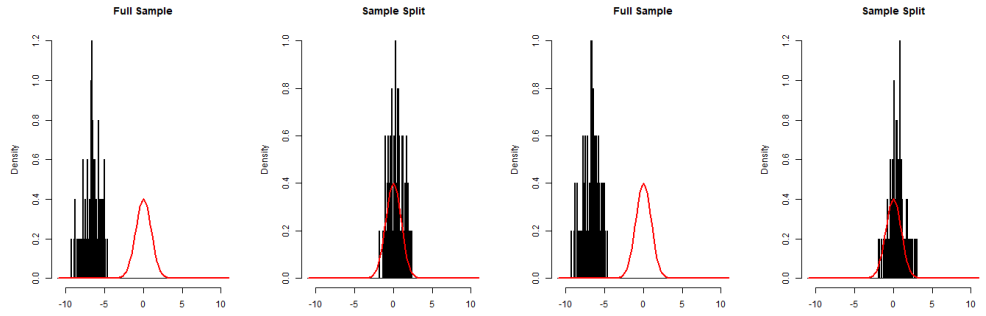
We first made a plot of 128 mean differences in two studentized estimators from using the full sample and using the sample splitting correspond to 128 different scenarios.



We may observe that most of the differences are around zero, suggesting that the difference between using sample splitting or not is small under most scenarios. However, we can still notice that under some scenarios the difference is huge, especially those on the left tail, suggesting that sample splitting may sometimes lead to larger bias. The conclusion is surprising, and we decided to look into the scenarios in the tail regions. It is notable that all four scenarios that lead to large differences correspond to data with high dimensions and models fitted with LASSO. For scenarios where all parameters for data generating process are identical but with model fitted with XGBoost, we do not observe such extreme differences. Hence, we believe those large differences could be caused by lack of converge of LASSO when dimension increases.



The above two plots are scenario 1 and 2 respectively, where the machine learning method is XGBoost, number of covariates is 5, sample size is 50, coefficients for covariates follow normal distribution, and the outcome model is generated from linear functions. The left plot is 2-fold sample splitting and the right plot is 5-fold. Under these two scenarios, we can see that using the full sample may cause large bias, and sample splitting can reduce the bias. We have calculated the difference between averaged bias of full sample and sample splitting across all 128 scenarios. In fact, under 66 out 128 scenarios, sample splitting method obtains lower average bias than full sample approach. This does not support that sample splitting approach is superior to full sample as it only out-perform full sample in around half of the scenarios. However, we think it is more meaningful to focus on extreme cases when full sample results bias significantly larger than that from sample splitting.



The above two plots correspond to the scenario where the machine learning method is LASSO, number of covariates is 100, sample size is 1000, the outcome model is generated from linear function, and the sample splitting strategy is 2-fold. The only difference is that the plot at left corresponds to data generated with coefficients for covariates following uniform distribution, and the right plot corresponds to data generated with coefficients for covariates following normal distribution. We can observe that under circumstances when both sample size and number of covariates are large, sample splitting can effectively reduce bias that would have been created if full sample are used instead.

In addition to plots, we have calculated some summary statistics including bias and variances for full sample, 2-fold sample splitting and 5-fold sample splitting. For the following tables, all the data are from the scenarios where the variance for the error term is 0.01, coefficients for covariates follow normal distribution, and the outcome model is

generated from linear functions. We adjust the sample size and number of covariates and make comparisons.

For the Table 1, the number of covariates is 5 and the sample size is 50.

Table 1

Method	Full	Twofold	Fivefold
XGBoost	-0.395(0.32)	-0.011(0.587)	-0.111(0.486)
LASSO	-0.072(0.047)	0.055(0.1)	0.018(0.07)

For the Table 2, the number of covariates is 100 and the sample size is 50. This is the case where the number of covariates is relatively higher than the sample size.

Table 2

Method	Full	Twofold	Fivefold
XGBoost	-0.695(2.641)	0.086(3.861)	-0.49(3.762)
LASSO	-0.36(2.117)	-0.229(2.944)	0.982(11.506)

From the two tables above, we can see that under both scenario, 2-fold sample splitting is better than full sample, but 5-fold sample splitting does not necessarily reduce bias from full sample.

For the Table 3 and 4, we compared the effect of sample splitting under different combinations of sample size the number of covariates.

Table 3

	N=50, Ncov=5		N=50, Ncov=100	
	Full	Twofold	Full	Twofold
XGBoost	-0.395(0.32)	-0.011(0.587)	-0.695(2.641)	0.086(3.861)
LASSO	-0.072(0.047)	0.055(0.1)	-0.36(2.117)	-0.229(2.944)

Table 4

	N=1000, Ncov=5		N=1000, Ncov=100	
	Full	Twofold	Full	Twofold
XGBoost	-0.068(0.077)	-0.023(0.082)	-0.029(0.61)	-0.025(0.609)
LASSO	-0.002(0.006)	0.003(0.008)	-0.095(0.014)	0.013(0.027)

7. Conclusions

In this paper, we studied the effect of sample splitting in the estimation of causal effects. We have replicated the result presented by Chernozhukov et al. (2018), which indicates that sample splitting do remove bias. Furthermore, we extend the simulation study to wider range of possible data generating possible and machine learning methods. We have incorporated cases when data are generated with various sample size, number of covariates , and both linear and non-linear form. We also considered both parametric machine learning model (LASSO) and non-parametric model (Gradient Boosting Tree), and different value of folds for sample splitting. Through simulating over 128 scenarios, we observe that with sample splitting, we can achieve estimates with smaller bias under most circumstances. Especially, when both sample size and dimension are large and the machine learning algorithm may suffer from lack of convergence, our simulations showed that estimations obtained using full sample approach may lead to significant deviation from the true value, and such bias can be removed by sample splitting. Our conclusion is that in finte sample study, sample splitting may not be superior to full sample under all scenarios, but it do provide more robust estimations under certain high dimensional cases.

References

- Athey, Susan, and Guido Imbens. "Recursive Partitioning for Heterogeneous Causal Effects." Proceedings of the National Academy of Sciences, vol. 113, no. 27, 2016, pp. 7353–7360., https://doi.org/10.1073/pnas.1510489113.
- Chernozhukov, Victor, et al. "Double/Debiased Machine Learning for Treatment and Structural Parameters." The Econometrics Journal, vol. 21, no. 1, 2018, https://doi.org/10.1111/ectj.12097.
- Wager, Stefan, and Susan Athey. "Estimation and Inference of Heterogeneous Treatment Effects Using Random Forests." Journal of the American Statistical Association, vol. 113, no. 523, 2018, pp. 1228–1242., https://doi.org/10.1080/01621459.2017.1319839.
- Seo, Michael, et al. "Comparing Methods for Estimating Patient-Specific Treatment Effects in Individual Patient Data Meta-Analysis." Statistics in Medicine, vol. 40, no. 6, 2020, pp. 1553–1573., https://doi.org/10.1002/sim.8859.
- Codes for figures in Section 4: https://github.com/VC2015/DMLonGitHub/blob/master/Figure2.m
- Lecture Notes provided by Professor Fan Li (Assessed via Sakai)
- Chernozhukov, Victor DML Slides, "Double/Debiased Machine Learning for Causal and Treatment Effects," May 31, 2018. (Accessed via Sakai)