Machine Learning Homework 2

Bingxue Li

8th, March, 2024

Part I: As shown in Table 1, $\hat{\beta}_{OLS} = 1.0691$ and $\hat{\beta}_{Lasso} = 1.0219$. Lasso estimators are biased downward towards 0 due to the penalty associated with non-zero coefficients. Although the estimated βs in these two cases are very close to the true $\beta_0 = 1$, the results are not reliable due to misspecification because we are fitting a linear model to data generated with nonlinear functions, i.e., sin() and cos(). We cannot interpret the results from the misspecified linear models as causal effects since conditional mean independence is not satisfied. Additionally, Lasso does not select variables in a consistent way; therefore, both are not interpretable in a causal or consistent sense.

With Post-lasso regression, we obtain $\hat{\beta}_{Post-Lasso} = 0.9451$, which is significantly biased downward. Recall that Lasso does not consistently select variables; several iterations could give $\beta_{Lasso,i-\text{iteration}} = 0$, and averaging across iterations gives us lower estimators. Indeed, Lasso would select a small set of variables that predict Y_i well. When γ_{0j}, the coefficient associated with X_{ij} is nonzero, or X_{ij} does contribute to the DGP of Y_i, and $Cov(D_i, X_{ij}) \neq 0$, we do not necessarily have X_{ij} selected by the Lasso algorithm, which leads to omitted variable bias.

Alternatively, one could consider applying regularized methods to learn the function $\beta \cdot D_i + g(X_i)$ following equation 14, allowing flexibility with $g()$. However, for estimators in equation 14, $\hat{g}()$ does not exhibit $\sqrt{n}$-consistency due to the bias necessarily introduced by any regularization that keeps the variance of the estimator from exploding, leading to a biased estimator of β. To overcome regularization bias, one should consider orthogonalization through conditioning on Xi for both Y_i and D_i and apply the thoughts of Frisch-Waugh Theorem, i.e., OLS regression with residuals. We want to achieve $\sqrt{n}$-consistency in the presence of the potentially high dimensional nuisance functions (g_0, m_0), therefore, nonlinear machine learning methods should be considered here. With Random Forest, we obtain $\hat{\beta}_{FWL-RF} = 1.0355$.

Although this would finally give us a consistent estimator of causal interpretation in theory, applying sample splitting and cross-fitting helps us to achieve the $\sqrt{n}$-consistency under mild conditions as it allows a reminder term to be $o_p(1)$ through independence of auxiliary and main sample (Appendix A.2). Finally, we consider the double machine learning estimator by Chernozhukov et al. (2018), using sample-splitting and cross-fitting and achieving full efficiency by rotating the role of subsamples, $\hat{\beta}_{DoubleML} = 0.9975$, the closest one to the true parameter.

Part II 1. We construct the K-fold double machine learning by first randomly splitting the full sample into k folds, denote the main sample as I_k, and the auxiliary sample, $\{W_i\}_{i=1}^{N} \backslash I_k$, as I_k^c. **2.** We train the nuisance functions, $(g(\cdot), m(\cdot))$ to make predictions $\hat{g}(X_i)$ and $\hat{m}(X_i)$ with ML method applied to I_k^c (considering that the data are not sparse in X_i and potential nonlinearity, better to apply nonlinear models). **3.** Obtain the residuals $Y_i - \hat{g}(X_i), D_i - \hat{m}(X_i)$ by applying trained nuisance functions to I_k. **4.** Perform OLS with residuals and obtain $\hat{\beta}_{0,k}$. **5.** Repeat 2 to 3 for all I_k fixed, obtain $\hat{\beta}_{0,1}, \hat{\beta}_{0,2}, ..., \hat{\beta}_{0,K}$, $\hat{\beta} = \frac{\sum_{i=1}^{K} \hat{\beta}_{0,k}}{K}$.

Firstly, one should notice that we recover the full efficiency by averaging across the k estimators just as 2-fold cases. Secondly, we apply the larger subsample ($\frac{k-1}{k} * N$) to perform prediction because learning the high-dimensional nuisance functions is usually more data demanding than the second step OLS regression. Thirdly, the choice of K does not affect the asymptotic performance of DML estimators as long as $N >> K$. K-fold cross-fitting double machine learning has the advantage of making the prediction model more consistent, $\hat{g}(\hat{m})(x_i) \xrightarrow{p} E[y_i(d_i)|x_i]$, as we are training with more observations, $\frac{k-1}{k} * N$, but the trade-off with respect to a second stage OLS regression, $\frac{1}{k} * N$, is immediate considering that the two subsamples are complementary. Just as any multiple stage plug-in estimation (i.e. 2SLS), first stage variance would impact the variance of final estimator, a precise first stage is hard to achieve and prioritized here. Additionally, from 2 to K, as the number of splits increases, the variability in the estimated performance metrics, accuracy, MSE, tends to decrease, implying estimates performance becomes more stable and reliable. However, concerns with estimator performance would arise once we consider a larger K value versus a small finite sample, in which case the K should be considered should not be too large.

Part III We try to replicate the paper on experiment with social pressure and voter turnout with double machine learning tool constructed above. Gerber, Green and Larimer (2008) shows that receiving mailings promising to publicize citizens' voting participation within neighborhoods increases turnouts rate in a large field-experiment. Here, instead of the randomized treatment, we use 44 covariates to be regularized to control for unobservables. We use different machine learning techniques with the k2dml function and different fold splits and show the estimation results in Table 4.

We firstly try linear models like elastic net with double machine learning, which gives estimates very close simple OLS. This is mainly due to the fact that our data is not sparse in the covariates and regularization with linear models improves very little in such a large experiment with much more observations than covariates. Then to account for nonlinearity, we try with random forest for different Ks (2 and 4), which indicate more modest treatment effects compared with naive OLS (around 9% rather than 14%). In the end, we find smaller variances associated DML with nonlinear method compared with simple OLS, this could be the merits of incorporating nonlinearity. In general, this is very close to the original papers' outcomes.

A Technical Appendix

A.1 Data Generating Process: Partially Linear Model

$$Y = \beta_0 \cdot D + g(X) + U \tag{1}$$

$$D = m(X) + V \tag{2}$$

$$X \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) \quad U \sim^{iid} \mathcal{N}(0,1) \quad V \sim^{iid} \mathcal{N}(0,1) \tag{3}$$

$$g(X) = \cos(X\boldsymbol{\beta})^2 \tag{4}$$

$$m(X) = \sin(X\boldsymbol{\beta}) + \cos(X\boldsymbol{\beta}) \tag{5}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \frac{1}{1} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{k} \end{bmatrix} \quad \boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6}$$

A.2 Section I: Comparison of Estimators

OLS

$$Y_i = \beta \cdot D_i + X_i' \cdot \gamma + e_i \tag{7}$$

Lasso

$$(\tilde{\beta}, \tilde{\gamma}) = \arg\min_{\beta,\gamma} \left\{ \sum_{i=1}^{N} (y_i - \beta \cdot d_i - x_i' \cdot \gamma)^2 + \lambda(|\beta| + \sum_{j=1}^{p} |\gamma_j|) \right\} \tag{8}$$

Tuning Parameter: λ.
Let $\hat{I} \equiv \text{support}(\tilde{\beta}, \tilde{\gamma})$ where $\tilde{\beta}$ and $\tilde{\gamma}$ denote the respective LASSO estimators obtained in 8. Furthermore, let W_i denote the vector of $\{D_i, X_{ij}\}$ for which $j \in \hat{I}$. Then, we run OLS such that,

$$Y_i = W_i'\gamma + \varepsilon_i \tag{9}$$

Post-double Selection Lasso
Based on Belloni, Chernozhukov, and Hansen (2014, ReStud),

$$\tilde{\beta_1} = \arg\min_{\beta_1} \left\{ \sum_{i=1}^{N} (D_i - \ X_i' \cdot \beta_1)^2 + \lambda_1 (\sum_{j=1}^{p} |\beta_{1j}|) \right\} \tag{10}$$

$$\tilde{\beta_2} = \arg\min_{\beta_2} \left\{ \sum_{i=1}^{N} (Y_i - \ X_i' \cdot \beta_2)^2 + \lambda_2 (\sum_{j=1}^{p} |\beta_{2j}|) \right\} \tag{11}$$

Tuning Parameters: λ_1, λ_2.
Let $\hat{I_1} \equiv \text{support}(\tilde{\beta_1})$ $\hat{I_2} \equiv \text{support}(\tilde{\beta_2})$ where $\tilde{\beta_1}$ and $\tilde{\beta_2}$ denote the respective LASSO estimators. Furthermore, let W_i denote the vector of X_{ij} for which $j \in \hat{I_1} \cup \hat{I_2}$. Then, we run OLS such that,

$$Y_i = \alpha D_i + W_i'\gamma + \varepsilon_i \tag{12}$$

Frisch-Waugh + ML method of choice
With the model $Y = \beta_0 \cdot D + g(X) + U$ and conditional mean independence, we construct the following transformed model:

$$Y_i - \mathbb{E}[Y_i|X_i] = \beta_0 \cdot (D_i - \mathbb{E}[D_i|X_i]) + \epsilon_i \tag{13}$$

Considering the nonlinearity of $g(\cdot)$ and $m(\cdot)$, we use random forest as the machine learning method to obtain a consistent prediction of Y_i, D_i conditional on X_i. We run OLS regression with the residual terms obtained after prediction.

Double Machine Learning

Based on Chernozhukov et al. (2018), first randomly split the sample by half to obtain I and I_c. Compute $\hat{\beta}_{0,1}$ with I as main sample using I_c as the auxiliary sample to obtain consistent prediction conditional on X_i. Interchanging the role of I_c and I repeat the procedure to obtain $\hat{\beta}_{0,2}$. The final estimator is given by $\hat{\beta} = \frac{\hat{\beta}_{0,1}+\hat{\beta}_{0,2}}{2}$.

Double Machine Learning: Why Orthogonalization?

$$\widehat{\beta}_0 = \left(\frac{1}{n}\sum_{i\in I} D_i^2\right)^{-1} \frac{1}{n}\sum_{i\in I} D_i(Y_i - \widehat{g}_0(X_i)) \tag{14}$$

As shown in Chernozhukov et al. (2018), $\hat(\beta_0)$ will have a convergence speed lower than $\sqrt{n}$ due to the fact any regularization that keep variance of high-dimensional nuisance function low necessarily introduces bias in $\hat{g}_0$. We therefore orthogonalize D with respect to X and remove the contamination introduced by directly subtracting $\hat{g}_0()$.

Double Machine Learning: Why Sample Splitting and Cross-fitting?

$$\check{\beta}_0 = \left(\frac{1}{n}\sum_{i\in I} \hat{V}_i^2\right)^{-1} \frac{1}{n}\sum_{i\in I} \hat{V}_i(Y_i - \hat{g}_0(X_i)) \tag{15}$$

where $\hat{V}_i = D_i - \hat{m}_0(X_i)$, with $\hat{m}_0(X_i)$ obtained with ML methods applied to I_c. To obtain $\sqrt{n}$-consistency with proper central limit theorem, we decompose the estimator as following:

$$\begin{aligned}
\sqrt{n}(\check{\beta}_0 - \beta_0) &= \left(\frac{1}{n}\sum_{i\in I} \hat{V}_i^2\right)^{-1} \frac{1}{\sqrt{n}}\sum_{i\in I} \hat{V}_i(D_i\beta_0 + g(X_i) + U_i - \hat{g}_0(X_i) - \hat{V}_i\beta_0) && (16)\\
&= \left(\frac{1}{n}\sum_{i\in I} \hat{V}_i^2\right)^{-1} \frac{1}{\sqrt{n}}\sum_{i\in I} \hat{V}_i((\hat{V}_i + \hat{m}(x_i))\beta_0 + g(X_i) + U_i - \hat{g}_0(X_i) - \hat{V}_i\beta_0) && (17)\\
&= \overbrace{(\mathbb{E}[V_i^2])^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I} V_iU_i}^{\equiv a} && (18)\\
&\quad \underbrace{+ \left(\mathbb{E}[V_i^2]\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}(m_0(X_i) - \hat{m}_0(X_i))(g_0(X_i) - \hat{g}_0(X_i))}_{\equiv b} && (19)\\
&\quad \underbrace{+ (\mathbb{E}[V_i^2])^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I} V_i(\hat{g}_0(X_i) - g_0(X_i))}_{\equiv c} && (20)
\end{aligned}$$

Note that a is an object we can apply CLT with with regularity conditions on error terms, b is $o_p(1)$ by consistency of machine learning model's prediction. What makes sample splitting desirable is c, we want to achieve $o_p(1)$ with c, we know that $\hat{g}()$ is achieved with I_c. If we do not use a sample splitting and cross fitting methods, the model error V_i will correlate with estimation error $\hat{g}_0(X_i) - g_0(X_i)$ due to the fact that estimation is done with the full sample, leading to bias of estimator when simply plug-in $\hat{g}()$. We get rid of this correlation and obtain $\mathbb{E}[V_i(\hat{g}_0(X_i) - g_0(X_i))] = 0$ through LIE conditioning on X_i. Finally, $\frac{1}{\sqrt{n}}\sum_{i\in I} V_i(\hat{g}_0(X_i) - g_0(X_i)) \xrightarrow{p} \mathcal{N}(0, \mathbb{E}[V_i^2 \cdot (\hat{g}_0(X_i) - g_0(X_i))^2])$ with the variance converge to 0 in probability by the same argument with $b = o_p(1)$. The rest arguments go with Slutsky Theorem. Therefore, we achieve $\sqrt{n}$-consistency with very mild conditions when sample splitting and cross-fitting provides a quick route for reminder term to vanish.

B Output Appendix

B.1 Section I: Comparison of Estimators

We calculate the average $\hat{\beta}$ over 100 iterations with a minimum of 1000 observations and 50 covariates.

Estimation Method	$\hat{\beta}$ ($\beta_0 = 1$)
OLS (2.1)	1.069105
Lasso (2.2)	1.021917
Lasso as screening method + OLS (2.3)	0.9450858
Post-double selection lasso (2.4)	1.070166
Frisch-Waugh + ML method of choice (2.5)	1.035527
Double machine learning (2.6)	0.997512

Table 1: Estimated β for Different Regression Methods

B.2 Section II: K-fold Double Machine Learning

Table 2: K-fold Double Machine Learning with Different Ks (With Random Forest, 5 iterations only, just to illustrate the merit of K-fold)

Method	$\hat{\beta}$	S.E.
$K = 2$	1.051	0.0273
$K = 3$	1.038	0.0272
$K = 5$	1.076	0.0278

As seen above, increasing K from 2 to 3 makes our estimator closer to the true parameter with smaller variance, but from 3 to 5 deviates more from the truth with large variance. There are merits and costs of k-fold cross-fitting undergoing here.

Table 3: K-fold Double Machine Learning with Different Ks With Test Code

Method	$\hat{\beta}$	S.E.
$K = 5$	1.0861	0.0271

B.3 Section III: Social Pressure and Voter Turnout with Double Machine Learning

Table 4: Replication with K-fold Double Machine Learning

Method	$\hat{\beta}$	S.E.
Naive OLS	0.142	0.013
Elastic Net ($K = 2$)	0.133	0.0133
Random Forest ($K = 2$)	0.097	0.068
Random Forest ($K = 4$)	0.085	0.069

Note that even though Y_i is binary we cannot get binary predictions here as in the lass classification assignment. We are interested in casual estimation with meaning inferences, and to guarantee that we have meaningful interpretation of $E[Y_i|D_i]$, we should treat Y_i as a continuous variable for a probabilistic prediction outcomes and gain more variability from it.