Final Report

Zeqiu.Yu(ZYV9962), Shuo.Han(SHV7753)

June 8, 2023

For academic integrity

Most parts of the project are realized by ourselves, and for three learners, we put our reference in the end. I refer to the article "Metalearners for Estimating Heterogeneous Treatment Effects Using Machine Learning." to learn the models. We also refer to the official document of Double Machine Learning method as guidance for using DML method. In IPW section, we use the SK-learn document to tune all the models.

Code and documentation (4 points)

We have submitted our codes to github. The file "codes" includes the methods we use like Backdoor adjustment, IPW, X-Learner, S-Learner, T-Learner and Double Machine Learning. The description of the methods and return parameters are shown in the file under each function. And it also contains all the detail of all data analysis, and model tuning, each section with a title describing the usage.

Estimation implementation (2 points)

We use many estimators in this project, like backdoor do-calculus, IPW(We put all estimator functions in a notebook cell). Here we plan to explain the IPW estimator. As shown in our update file, our IPW estimator is given by: (TG: treatment, CG: control)

$$ATE = \frac{1}{n}\sum_{i\in TG}\frac{y_i}{\hat{p}(A_i|\vec{C_i})} - \frac{1}{n}\sum_{i\in CG}\frac{y_i}{\hat{p}(A_i|\vec{C_i})}$$

In the formula above, we look at each sample with treatment A_i, outcome Y_i and the confounder vector $\vec{C_i}$, and the $\hat{p}$ is the estimator of $p(A|\vec{C})$. We can estimate it with many methods: like logistic regression models and random forest.

Noted that our data consist of all categorical data, so we also make the outcome variable into 0-1 label, with 1 represent better performance. And after using the one-hot encoder, we use the sklearn.LogisticRegression model to train the model $P(A|C)$ for some covariates C. We actually confirmed that LogisticRegression model with L2 penalty can get the highest accuracy. Then the model is generated form: **LogisticRegression(C = 1, penalty="l2").fit(C = ["gender","race/ethnicity","parental level of education","test preparation course"], confounders["lunch"])**. In this model, "gender","race/ethnicity","parental level of education","test preparation course" corresponds to C, and lunch refers to A. Then we print the parameters for this fitted model:

	coef
intercept	**-0.298**
gender	**-0.0890**
race/ethnicity	**-0.0850**
test preparation course	**0.009**
parental level of education	**0.0777**

We didn't use this method in the update file, so we decide to show this one. Then we can use $E(Y^a) = \frac{1}{n}\sum_{i \in TG} \frac{y_i}{\hat{p}(A_i|\vec{C_i})}$ to calculate the counterfactual. We will use this model to estimate $P(A_i|\vec{C_i})$ for every individual then we can calculate the $E(Y^a)$.

We also use other estimation methods (meta-algorithms), we will show them in the section **Changes since the update**.

Changes since the update (3 points)

Since the Update, we continue implementing IPW, learn different meta learner models and Double Machine Learninng methods to our dataset.

IPW

IPW estimator has been discussed so many times and we have mentioned it in our update. We used gridSearchCV to find the best parameter set to get the inverse propensity score. The results will be given later as a table.

T-learner, X-learner and S-learner

All of these three methods are used to estimate the conditional average treatment effects, namely the effect of lunch type on the students' performance.

$$CATE = E[Y(1) - Y(0)|\vec{C}]$$

, Y(1) is the treatment group and Y(0) is the control group. We calculate the mean difference conditional on the founders. I use $\mu_1 = E[Y(1)|\vec{C}]$ and $mu_2 = E[Y(2)\vec{C}]$.

T-Learner

We use the observations to train the prediction models. Here I use XGBoost as the prediction model, and try to predict $\hat{\mu_1}$ with the observations in the treatment group and try to predict $\hat{\mu_0}$ with the observations in the control group. Then follow $C\hat{A}TE = \hat{\mu_1} - \hat{\mu_0}$. According to the result, we have a ATE of -0.24 together with 95% confidence interval [-0.79, 0.41].

X-Learner

The X-learner is based on T-Learner and much more complicated. Both of them train two models first. However, the X-learner involves individual treatment effect. For observations in the treatment group and control group, we can denote the individual effect with $\hat{\mu_{1i}} - Y_i$ and $Y_i - \hat{\mu_{10i}}$ respectively. Then for each group, we can calculate a CATE, we use m(c) and (1- m(c)) as the weights(propensity), then calculate the sum as the estimated CATE. According to the result, we have a ATE of -0.234 together with 95% confidence interval [-0.77, 0.42].

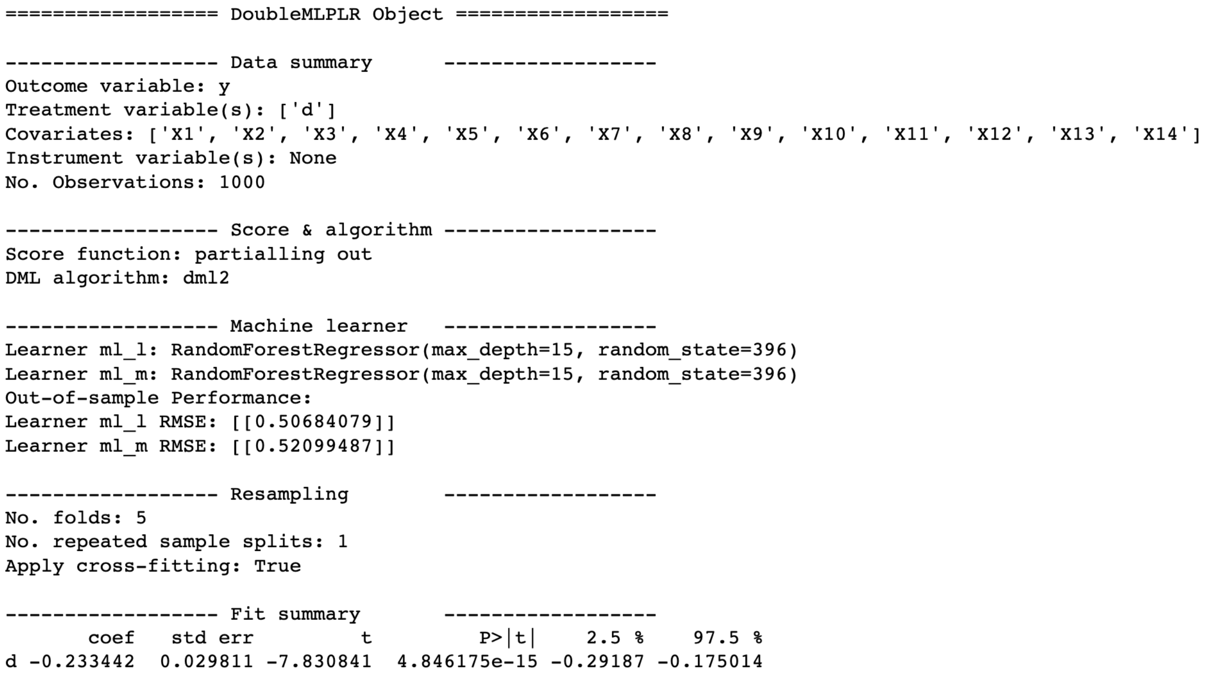
S-Learner

The difference between T-learner and S-learner is that S-learner also treats the treatment as a confounding factor, then our model is to train to predict with the confounders and the treatment. Here I also use XGBoost to predict $\mu(\hat{1})$ and $hat\mu(0)$, then $\hat{CATE} = \mu(\hat{1}) - \mu(\hat{0})$. According to the result, we have a ATE of -0.220 together with 95% confidence interval [-0.28, -0.16].

Double Machine Learning

As mentioned in our update file, the relationship between A and Y can be confounded by a nonlinear relationship. And here we use a packages **doubleml** to help us do this analysis. The code is also in the notebook with section called double machine learning. The resource is shown in reference.

Then, we create polynomial features for the high-dimensional confounders using PolynomialFeatures from sklearn.preprocessing, trying to capture more nonlinear relationship and interaction between the confounders.



```
================== DoubleMLPLR Object ==================

------------------ Data summary      ------------------
Outcome variable: y
Treatment variable(s): ['d']
Covariates: ['X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X7', 'X8', 'X9', 'X10', 'X11', 'X12', 'X13', 'X14']
Instrument variable(s): None
No. Observations: 1000

------------------ Score & algorithm ------------------
Score function: partialling out
DML algorithm: dml2

------------------ Machine learner   ------------------
Learner ml_l: RandomForestRegressor(max_depth=15, random_state=396)
Learner ml_m: RandomForestRegressor(max_depth=15, random_state=396)
Out-of-sample Performance:
Learner ml_l RMSE: [[0.50684079]]
Learner ml_m RMSE: [[0.52099487]]

------------------ Resampling        ------------------
No. folds: 5
No. repeated sample splits: 1
Apply cross-fitting: True

------------------ Fit summary       ------------------
       coef   std err         t         P>|t|    2.5 %    97.5 %
d -0.233442  0.029811 -7.830841  4.846175e-15 -0.29187 -0.175014
```

The next step is to fit a Double Machine Learning model based on the polynomial features. We define the Base Learner with a random forest regressor model (maximum depth: 15, n-estimators:

100). Then we clone the Base Learner and create a DoubleMLPLR object. Finally we fit the DML model using the polynomial data and print the result, Which is shown above. And according to the result, we have a ATE of -0.2334 together with 95% confidence interval [-0.292, -0.175].

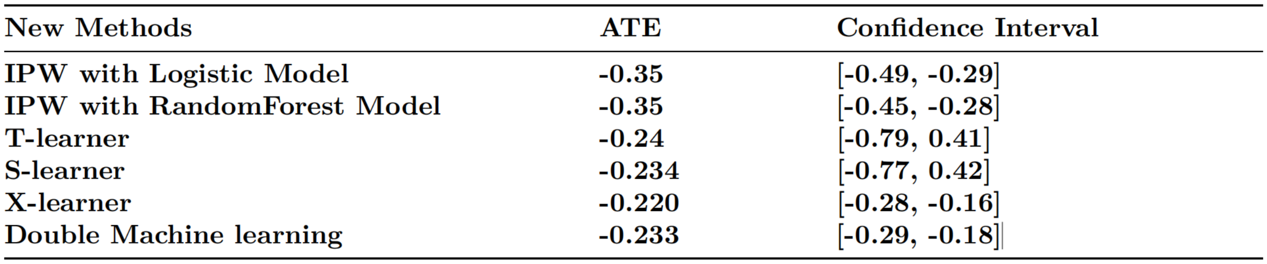
Interpreting your results (3 points)

Before

We are interested in whether having the free/reduced lunch is going to have a negative effect on students' test performance. In naive ATE method, we get a ATE of -0.22 together with confidence interval [-0.29,0.16], and in do calculus backdoor, we get a ATE of -0.25 together with confidence interval [-0.31, -0.20]. And this value is significantly smaller than 0, and we can tell that students who have free/reduced lunch are more likely to perform badly during test. Most importantly, all of our results are significant at a 95% confidence level so I think this is something that all high school and relative policy department should pay attention to.

After

After some improvement and inplement of new methods, we have the following result :



New Methods	ATE	Confidence Interval
IPW with Logistic Model	**-0.35**	**[-0.49, -0.29]**
IPW with RandomForest Model	**-0.35**	**[-0.45, -0.28]**
T-learner	**-0.24**	**[-0.79, 0.41]**
S-learner	**-0.234**	**[-0.77, 0.42]**
X-learner	**-0.220**	**[-0.28, -0.16]**
Double Machine learning	**-0.233**	**[-0.29, -0.18]**

As we can see from the results, we try many new methods. And most methods have a estimation of ATE around -0.235. And the results from IPW methods tend to be smaller. And results form all learners seem to have bigger confidence interval range. Also recall that we incorporate high dimensional confounders in the double machine learning method. And it help us to generate non-linear features. The results from DML also tells us that the effect do exist.

Before vs After

After comparing results from before and after, we can conclude 2 major things.

First, we can tell that students who have free/reduced lunch are more likely to perform badly during test. And we think the mechanism behind this that this variable is a direct correlation to the income of parents and whether the students can take enough food after school in the US and this variable also shows the students' eating style outside of campus (Since students who can afford lunch are more likely to eat better at home dinner and during weekends while the students who have free lunch might not eat enough after school), which may affect the physical conditions of the students, like brain development.

Second, we were worried that the relationship between A and Y is confounded by a nonlinear relationship involving some high-dimensional confounders. But as we can see from the result from DML, such worries might not exist since we are getting similar results compared with backdoor do-calculus. But it do have a smaller variance and we have the reason to believe that it has more stable estimation.

Reflections (4 points)

What was interesting?

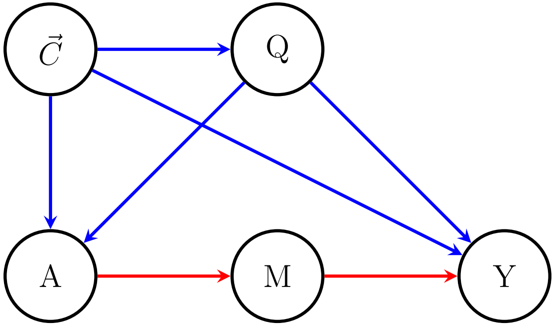
We know that the DAG in our project is a relatively simple one. But we still learn a lot from this project. We did check the slice 'Assumption', 'Adjustment' and 'propensity' to help us with the project. Also, we read many resources online to help us implement do-calculus, X,T and S learner and this help us understand more about other estimators which are not mentioned during class. And we also study the documentation for DoubleML to help us analyse this casual question, and we used dowhy for the extra credit part, we know the power of using others' packages, which is more efficient and they all contains more detailed results for us to interpret our results.

What was difficult?

The most difficult part is all about the datasets and interpreting of the results. It is always helpful to choose a dataset and think more about it whether it can work or not. How complex it is going to be. At first we had a image X-ray dataset, we didn't check this dataset enough and started the proposal directly. If a friend of mine is going to take this course, I would encourage him to do the analysis to the dataset at the most beginning, find out what the outcome and treatment are, make a basic outline of what analysis he should do. Also, after finding the result, It is difficult for as to interpret the findings and the mechanism behind this surprising casual question.

Unaddressed challenges

First reason is the missing variables and confounders, this dataset contains limited variables, and there are many other features that will also infect the results.



Let's now consider a new DAG, where Q represent family economic status of this students. And the family economic status will be influenced by their parental level education in the confounder C. And family economic status will also influence A, whether the student can afford the lunch at school. M can represent the student's IQ, brain development, and this variable can be influenced by variable A and it will also influence on the performance Y. Given this new graph, the backdoor criteria won't work by simply condition on all confounders.

The second reason is followed by the first reason, we still need to use DML method to consider the non-linear relationship between the outcomes and the confounders and treatments, since we are having new variables and DML always assumes that there is no unmeasured confundings. In this project we find out this problem is fine in this small dataset, but when having more confounders and more data points, we need to reanalyse this problem again, otherwise it might influence the result. In sum, the first reason is more important, and it is always batter to check with the second reason.

What’s left to do?

We think we have done enough on this dataset. But when given one more month together with 1,000,000 dollar, we would use this money to help us build a better data set, with more variables and more data points based on some questionnaires in the entire states. In this case, we will have a much bigger data with more information. However, I can imagine that more problems will also arise. We will need to deal with the missing value problem maybe with MICE algorithms, since in this way, the collected data must contain missing data. Then we plan to go through the analysis framework again in this project with a new DAG. And if again we get to find out such an effect do exist, we might consider giving some concrete evidence to relevant policy makers to make some changes like give such group of students more benefice after class such as free dinner or offer them nutritional supplement as an extra benefit. And in the end, we can use our funding to help a lager group of students in control group and see whether their test performance will improve after some time. Then we get to collect new feedback data to double check our conclusion.

References

- Amit Sharma, Emre Kiciman. DoWhy: An End-to-End Library for Causal Inference. 2020. https://arxiv.org/abs/2011.04216
- Bach, P., Chernozhukov, V., Kurz, M. S., and Spindler, M. (2022), DoubleML - An Object-Oriented Implementation of Double Machine Learning in Python, Journal of Machine Learning Research, 23(53): 1-6, https://www.jmlr.org/papers/v23/21-0862.html.
- Künzel, Sören R., Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. ”Metalearners for Estimating Heterogeneous Treatment Effects Using Machine Learning.” February 15, 2019. https://doi.org/10.1073/pnas.1804597116.
- ”Students Performance in Exams.” Kaggle dataset. Accessed May 25, 2023. https://www.kaggle.com/da performance-in-exams
- ”DoWhy 0.9.1 Documentation.” PyWhy. Accessed May 25, 2023. https://www.pywhy.org/dowhy/v0.9.1
- Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.