POLS 904 Final Project

Monte Carlo Simulation on Causal Forest

Jiacheng He

December 18, 2017

Introduction

In social science, researchers might be interested to estimate the effect of a binary treatment (either treated or not treated). In experimental setting, individuals are randomly assigned into a control group and a treated group. Formally, denote y_i as the outcome variable, W_i as the treatment assignment variable ($W_i = 1$ if in the treated group, $W_i = 0$ if in the control group), and X_i as a set of observed covariates (e.g. age, gender, race, eduction, etc). Then the researcher can estimate the classical linear model:

$$Y_i = \tau W_i + X_i\beta + \epsilon_i \tag{1}$$

Here $\tau = E[Y_i|W_i = 1] - E[Y_i|W_i = 0]$ is intepreted as the average treatment effect (ATE) across all individuals. X_i is included into the regression to make sure of unconfoundedness and to reduce the variance of the estimator $\hat{\tau}$.

But sometimes researchers might want to go beyond ATE, and try to further estimate heterogeneous treatment effect and identify the subgroup of the population who will benefit the most (or least) from the treatment. One approach is to estimate the conditional average treatment effect (CATE) $\tau(X_i) = E[Y_i|X_i, W_i = 1] - E[Y_i|X_i, W_i = 0]$. That is, express the treatment effect τ as a function of the observed covariates.

Causal forest developed by Wager and Athey (2017) aims to algorithmically search for the covariate space, identify the subspace where heterogeneity exists, and estimate the CATE in these subspaces. It is very similar to the popular random forest method. Wager and Athey (2017) also derived asymptotic distribution of the causal forest estimator so that statistical inference and hypothesis test become feasible when adopting this forest based method.

In this project, I run Monte Carlo simulation on the causal forest to examine its finite sample perfomance, such as the mean squared error (MSE) and the confidence interval coverage rate.

Model and Algorithm

Consider a simple additive model. For each individual i, we observe the outcome variable y_i, the treatment status W_i, and a vector of covairates X_i. Define $m(X_i) = E[Y_i|X_i]$ as the conditional mean of the outcome (in regardless of individual i is treated or not), $\tau(X_i) = E[Y_i|X_i, W_i = 1] - E[Y_i|X_i, W_i = 0]$ as the CATE, and $e(X_i) = E[W_i|X_i]$ is the conditional treatment propensity. These functions are not observed and we are interested to nonparametrically estimate $\tau(X_i)$.

With some derivation, we will have such relationship:

$$Y_i = m(X_i) + \frac{W_i}{2}\tau(X_i) + \frac{1 - W_i}{2}\tau(X_i) + \epsilon_i$$

where ϵ_i is a disturbance error term.

There are several challenges in nonparametrically estimate the CATE function $\tau(X_i)$. First, in real world application, we never observe the true individual treatment effect τ_i. At each moment, an individual is

either in the treated status or in the non-treated status, so we never know what would have happened to the individual if the individual would have shifted his/her status. This is the fundamental problem in causal inference. As a result of the absence of the true τ_i, we can not perform cross validation, which is the routine in predictive machine learning.

Second, the existence of non-constant $m(\cdot)$ and $e(\cdot)$ will tend to confound our estimation, as I showed in the presentation in the final exam day.

The training algorithm of the causal forest is very closed to the random forest. We first train a large number of causal trees, then average them to obtain the forest estimates. When growing each single tree, we randomly draw a subsample of both observations and covariates. Also, similar to random forest, in causal forest algorithm it is not necessary to implement regularization or pruning when growing each single tree. However, there are two major divergence:

1. When growing each tree in the causal forest, we place the split at the point $\tilde{x_i}$, which maximizes the difference of $\hat{E}[Y_i|X_i = x_i, W_i = 1] - \hat{E}[Y_i|X_i = x_i, W_i = 0]$ ($\hat{\tau}$) across the two sides of $\tilde{x_i}$. While in the case of random forest we place the split based on $\hat{E}[Y_i|X_i = x_i]$ ($\hat{y}$).
2. When growing each tree, we use half of the training sample to do Step 1 above (placing split, identifying heterogenity covariate subspace), and use the other half of the training sample to calculate the $\hat{\tau}$ (estimation of the CATE in that subspace). Wager and Athey (2017) refer to this criterion as "honest splitting". Honest splitting is a strategy to remedy the infeasibility of cross validation.

Simulation Setup

In the Monte Carlo simulation experiment, I am interested to see how the algorithm performs as sample size and number of covariate change, under two different senarios: 1. constant treatment effect; 2. heterogeneous treatment effect. I set up two data generating processes (DGP) as:

DGP1 (constant τ)

$$\begin{aligned} \tau(X_i) &= 0 \\ e(X_i) &= (1 + f_{beta}^{2,4}(X_{1i}))/4 \\ m(X_i) &= 2X_{1i} - 1 \end{aligned}$$

where $f_{beta}^{2,4}(\cdot)$ is the density function of Beta distribution with shape parameters 2 and 4.

DGP2 (heterogeneous τ)

$$\begin{aligned} \tau(X_i) &= 1 + \frac{1}{(1 + e^{-20(X_{1i} - 1/3)})(1 + e^{-20(X_{2i} - 1/3)})} \\ e(X_i) &= 0.5 \\ m(X_i) &= 0 \end{aligned}$$

Training causal forests also requires setting up tuning parameters, the same as when we train random forests. In this project, I also try to vary different tuning paramters and evaluate the performance of these trained models. The tuning parameters I try are as follows: (1) sample fraction used in growing each tree; (2) covariates used in growing each tree; (3) Number of trees to build the forest; (4) Minimun number of observations in each terminal leaf; (5) Regularization parameter λ.

Please keep in mind that cross validation is not feasible in treatment effect estimation. So in practice there is no general guidance to select these tuning parameters in the training stage. What's more, evaluation of choices of tuning parameters is only possible when we assume a data generating process and hence know the true τ_i in Monte Carlo simulation (not in real data). And we can only implement the evaluation in the test set (not in cv set).

I draw $X_i \sim U(0,1)^d$, $W_i \sim binom(1, e(X_i))$, $\epsilon_i \sim N(0,1)$. (d is the number of covariates). Then I train the causal forest model on a training set, and evaluate the trained model on a test set with 100 data points. For each senario, I replicate it for 100 times. For each replication I generate a new training set, while the test set is invariant for all replications. Then I plot the box plot of the MSE and 95% confidence interval coverage rate.

Result

Sample Size and Number of Covariates

First, I will look at how the performance of the causal forest respond to the change of sample size n and number of covariates d:

1. Fix $d = 10$, try $n = 100, 500, 1000, 2000, 5000$;
2. Fix $n = 1000$, try $d = 2, 4, 10, 20, 40$;

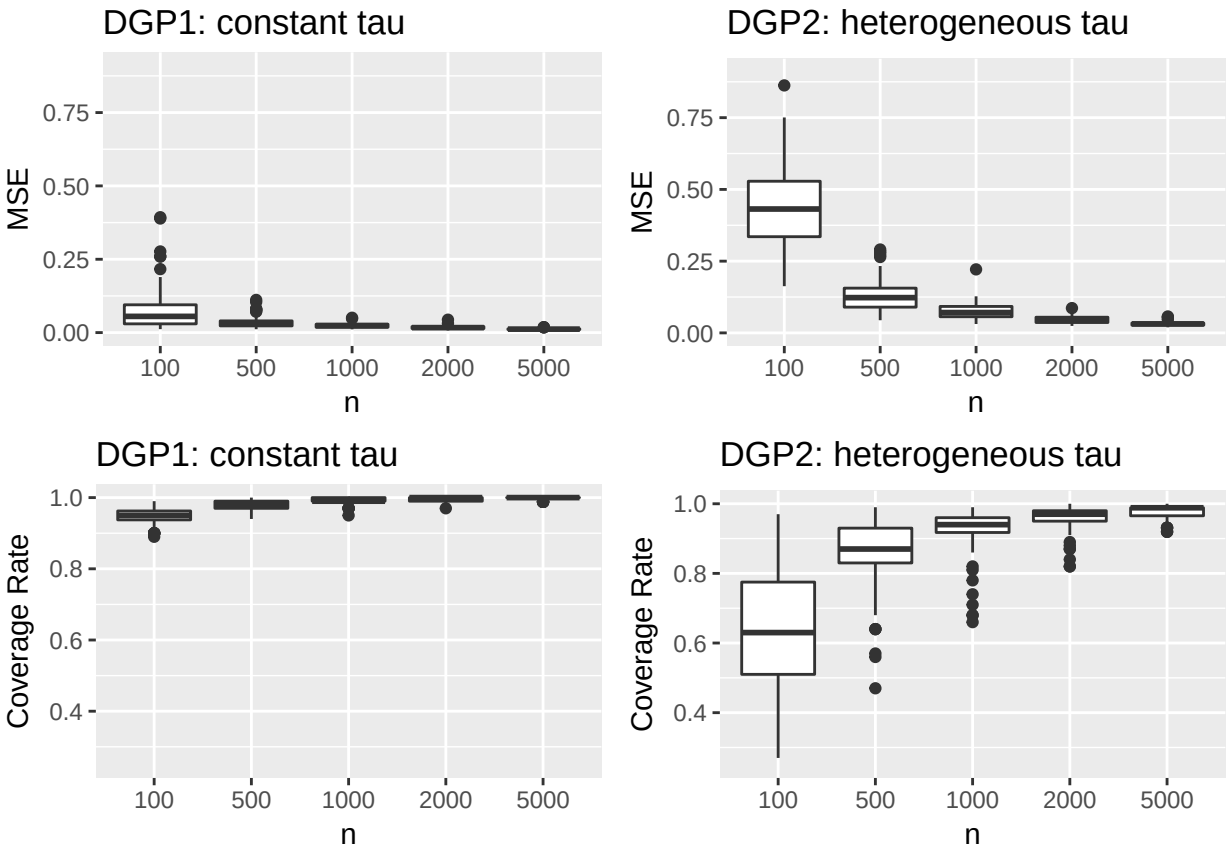

Figure 1: MSE and Coverage Rate with Different Sample Size n

Figure 1 displays the MSE and 95% confidence interval coverage rate. Under the data generating process of constant treatment effect, the MSE decays very fast as sample size n increases. Under the DGP of heterogeneous treatment effect, the MSE is slightly larger and it decays slightly slower. When n is as large as 5000, we achieve considerably small MSE.

Surprisingly, the confidence interval performs very well for the DGP of constant τ. Even in small sample $n = 100$, the simulation coverage rate achieves about 95%. What is interesting is that the confidence interval

seems "over-accurate" for DGP1. In the case of $n = 5000$, we obtain almost 100% accuracy that the 95% confidence interval will always cover the true τ.

As for the DGP2, we need larger sample for the confidence coverage to converge. When $n = 100$, in median the 95% confidence interval successfully covers the true τ for only about 60% of the time. When $n = 1000$, the median coverage rate achieves 95%. However, when $n = 5000$, we have the "over-accurate" issue again.

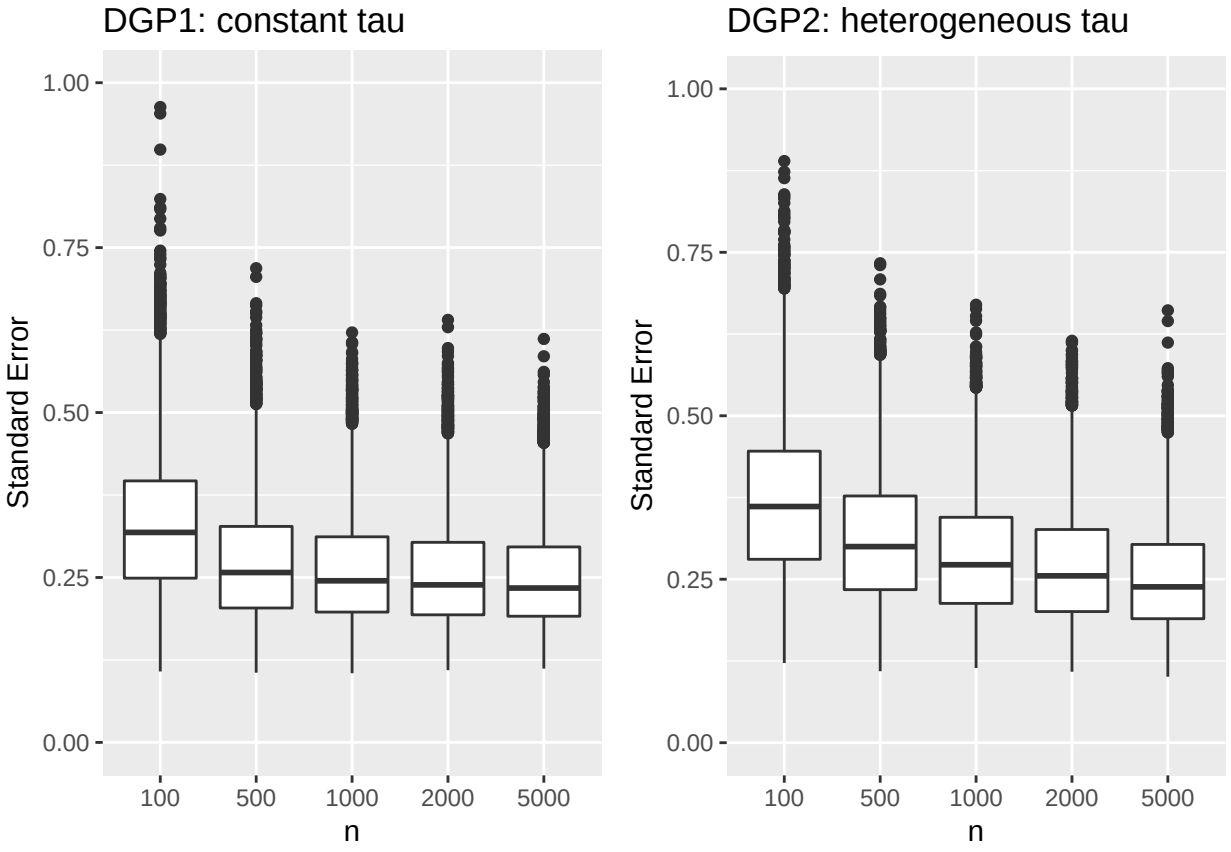

Figure 2: Standard Errors with Different Sample Size n

In Figure 2 I plot the standard errors in the simulation. We can see that the standard errors are pretty stable and have not converged even when $n = 5000$. Theoretically, as sample size increases, we should have smaller standard error for our estimator. But it is clearly not the case here. Figure 1 has told us that the MSE is shrinking and basically converges when $n = 5000$. In other words, the point estimates are very close to the true value when $n = 5000$. Therefore, I conjecture that the standard errors are "too large" in large sample, and hence the 95% confidence intervals are "too wide", which might lead us to the situation that, in hypothesis testing we are less likely to reject than we should.

In both data generating processes, only the first two covariates $X1$, $X2$ contribute to the $\tau(\cdot)$ function. Therefore, adding extra covariates is purely adding "noise" to the causal forest algorithm. We should expect the variance of the heterogenous effect estimates increases as d increases.

The simulation results of DGP2 with heterogeneous treatment effect meet our expectation. Although for resaons when the number of the covariates increases from 2 to 10, the MSE decreases a little bit, the MSE does not keep decreasring when go beyond 10. In fact, the variance of the MSE increases when we increase the number of the noise covariates from 10 to 40. The same happens to the confidence interval coverage rate. Since the sample size is defaulted to 1000, we already achieve a pretty good median coverage rate. As we have more noise covariates, the coverage rate becomes more and more unstable. When $d = 40$ (that is, most

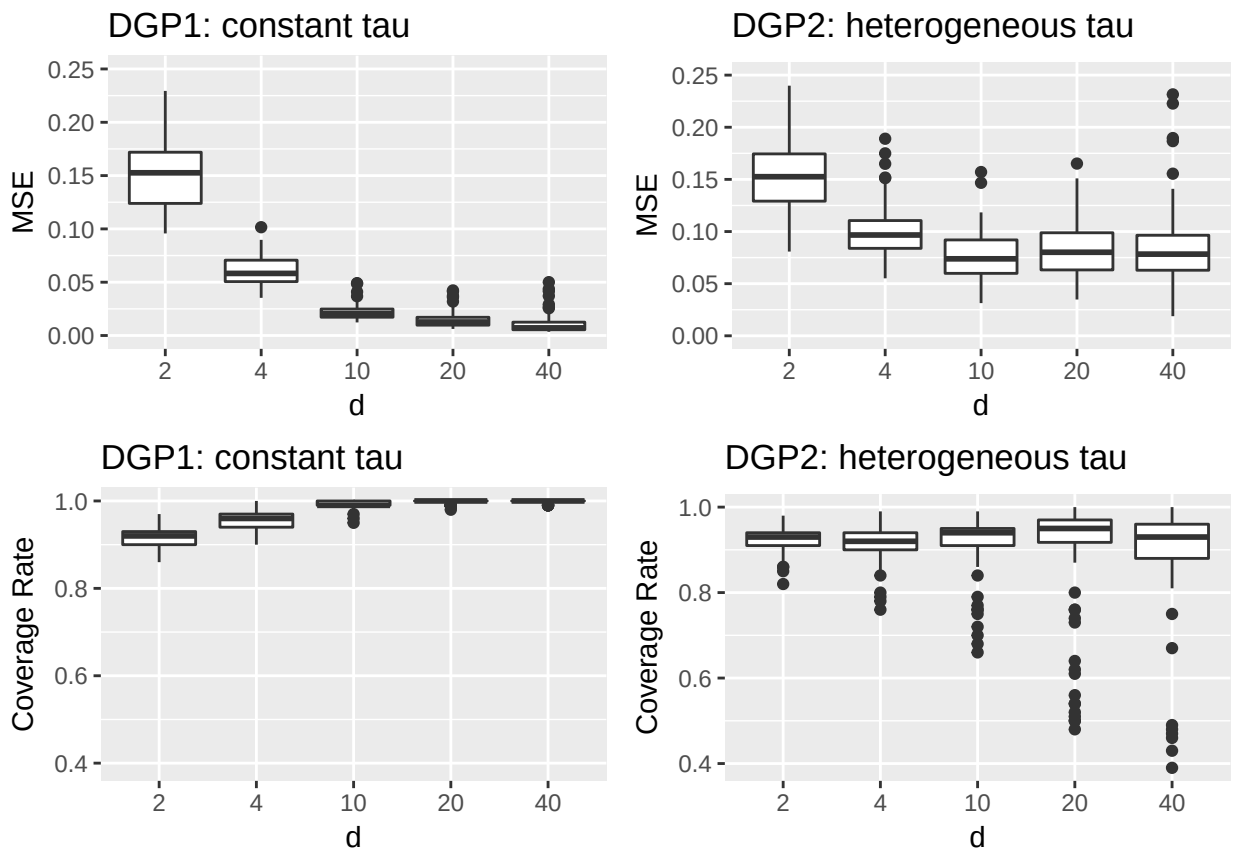

Figure 3: MSE and Coverage Rate with Different Numbers of Covariates d

of the observed covariates are not predictive to τ at all), in some test data points we will have very small (say, only 40%) coverage rate by chance.

However, this is not the case for DGP1. As we see in Figure 3, the MSE is shrinking when the number of noise covariates increases. And again we achieve almost 100% accuracy in the 95% confidence intervals when $d = 40$. This result is puzzling.

Tuning Parameters

I try varying five tuning parameters, one at a time. The simulation setting is default to: DGP2; $n_{train} = 1000$; $d = 10$; $n_{test} = 100$.

These are the tuning parameters I test:

1. Sample fraction used in each tree training; (default 0.5)
2. Covariates used in each tree training; (default $\frac{2}{3}d$)
3. Number of trees; (default 2000)
4. Minimun # observations in each terminal node; (default NULL)
5. Regularization parameter λ; (default 0)

In this section I only report the median MSE and median coverage rate for each senario. The simulation results are as follows:

The causal_forest() function in the grf package does not allow me to set the sample.fraction parameter smaller than 0.5. So I try to vary the sample.fraction from 0.1 to 0.5. As we see, perhaps we should stick to

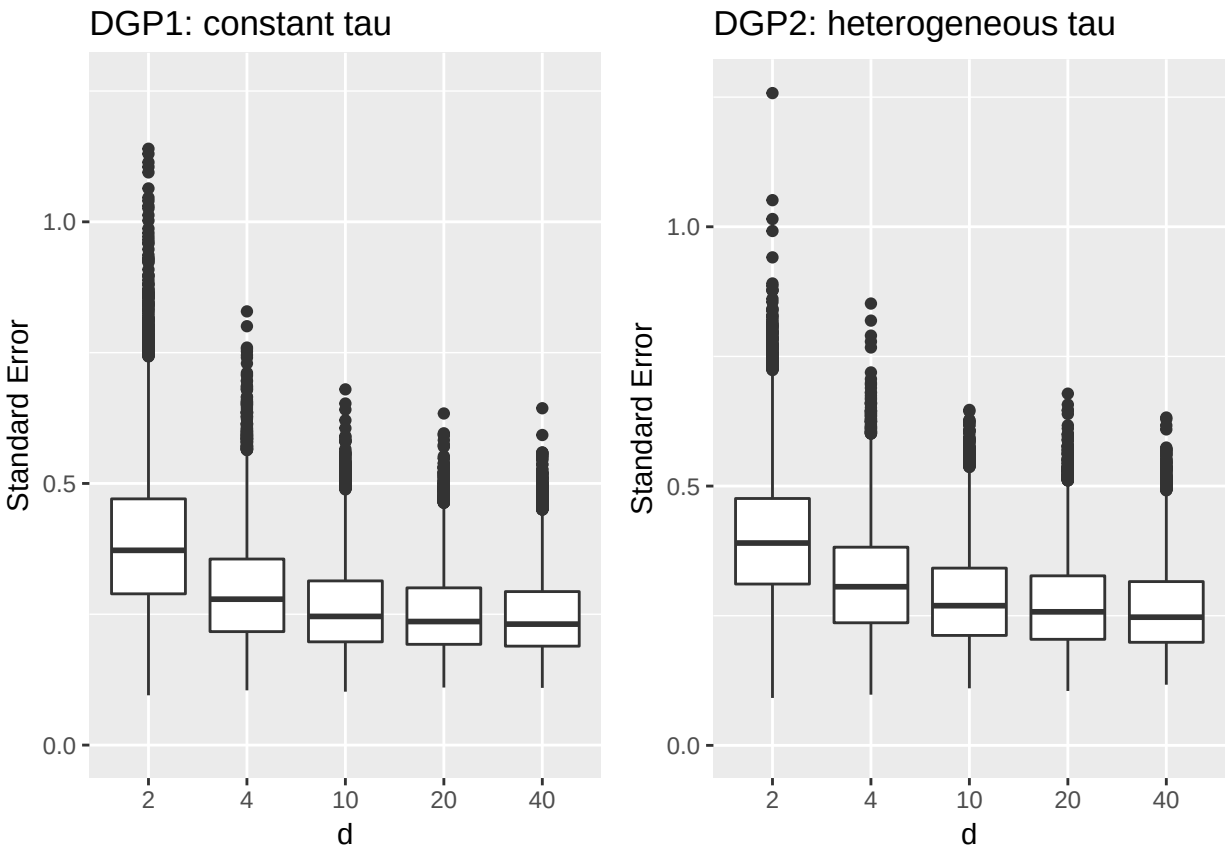

Figure 4: Standard Errors with Different Numbers of Covariates d

the default value 0.5.

I Try sample fraction $s = 0.1, 0.2, 0.3, 0.4, 0.5$

Table 1: Different Sample Fraction s

s	MSE	coverage
0.1	0.242	0.535
0.2	0.123	0.87
0.3	0.082	0.92
0.4	0.073	0.94
0.5	0.075	0.94

In causal forest we also use a subset of sample and subset of covariates to grow each tree. The default value of subsetting covariates is $\frac{2}{3}d$. In this case, it is 7. As we see, 7 covariates do give us the best performance, in terms of both MSE and coverage rate.

Table 2: Different Number of Training Covariates t

t	MSE	coverage
4	0.102	0.865
5	0.082	0.92
6	0.079	0.92
7	0.07	0.94
8	0.074	0.94

Intuitively, the more trees we train, the more variance we will average out. As number of trees increase, we do get smaller and smaller MSE as we tarin more trees. However, the gain is only marginal after we go beyond 1000 trees. But the computer training time is basically linear in the number of training trees. On the other hand, the coverage rate also decreases when we train more trees. Thus it is not clear whether it is worthwhile to train more than 2000 trees.

Table 3: Different Number of Trees b

b	MSE	coverage
500	0.088	0.97
1000	0.079	0.96
2000	0.076	0.95
4000	0.077	0.94
6000	0.075	0.915

The default setting of the algorithm does not set up the minimum node size. I try to vary the minimun node size from 0 to 80. When it defualt to 0, we achieve the best coverage rate. So we should stick to the default value 0.

Table 4: Different Minimum Node Size

size	MSE	coverage
0	0.068	0.94
10	0.066	0.89
20	0.066	0.9

size	MSE	coverage
40	0.063	0.915
80	0.076	0.905

Regularization is one of the most prominent features of machine learning algorithms. When we train decision tree, we usually add a regularization term in the optimization stage and perform backward pruning to prevent over-fitting. But when we do ensemble and average out many trees, we usually do not perform regularization. I try to vary the regularization parameter λ from 0 (default value) to 10 to see if we need to regularize in causal forest. The results below indicate that we achieve the best coverage rate, as well as second-best MSE. So, the answer is no we do not need regularization.

Table 5: Different Regularization Parameter lambda

lambda	MSE	coverage
0	0.067	0.95
0.1	0.069	0.94
1	0.066	0.94
5	0.076	0.925
10	0.079	0.92

Discussion and Conclusion

In this project I implement Monte Carlo simulation on the novel causal forest method (Wager and Athey (2017)), which introduced the philosophy of predictive machine learning and the powerful random forest algorithm into heterogeneous treatment effect estimation and statistical inference. My simulation study shows that as sample size increases, the predictive accuracy of the point estimates increases rapidly. We obatin very good performance in terms of MSE when $n = 5000$. However, the standard error estimates, which is one of the most distinguished features of their paper, does not significantly shrink as sample size increases, which might lead to an over-rejection problem in hypothesis testing.

I also examine the impact of varying the tuning parameters of the algorithm. My simulation results of the optimal tuning parameters are consistent with our common knowledge to the basic random forest algorithm: when growing each single tree, we should set the number of covariates as $\frac{2}{3}d$ and avoid to put regularization and restriction. And finally, the more trees, the better predictive accuracy.

Reference

Wager, Stefan, and Susan Athey. 2017. "Estimation and Inference of Heterogeneous Treatment Effects Using Random Forests." *Journal of the American Statistical Association* 0 (ja). Taylor & Francis: 0–0. doi:10.1080/01621459.2017.1319839.